Berkeley DB .NET API Documentation
BerkeleyDB Namespace
<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AckPolicy</td>
<td>The AckPolicy class specifies how master and client sites will handle acknowledgment of replication messages which are necessary for &quot;permanent&quot; records. The current implementation requires all sites in a replication group configure the same acknowledgement policy.</td>
</tr>
<tr>
<td>ActiveTransaction</td>
<td>The ActiveTransaction class describes a currently active transaction.</td>
</tr>
<tr>
<td>BadSecondaryException</td>
<td>A secondary index has been corrupted. This is likely the result of an application operating on related databases without first associating them.</td>
</tr>
<tr>
<td>BaseCursor</td>
<td>The abstract base class from which all cursor classes inherit.</td>
</tr>
<tr>
<td>Cursors</td>
<td>Cursors may span threads, but only serially, that is, the application must serialize access to the cursor handle.</td>
</tr>
<tr>
<td>BaseDatabase</td>
<td>The base class from which all database classes inherit.</td>
</tr>
<tr>
<td>BTreeCursor</td>
<td>A class for traversing the records of a BTreeDatabase</td>
</tr>
<tr>
<td>BTreeDatabase</td>
<td>A class representing a BTreeDatabase. The Btree format is a representation of a sorted, balanced tree structure.</td>
</tr>
<tr>
<td></td>
<td>A class representing configuration</td>
</tr>
<tr>
<td>Class Name</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------</td>
<td>-------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>BTreeDatabaseConfig</td>
<td>parameters for BTreeDatabase</td>
</tr>
<tr>
<td>BTreeStats</td>
<td>Statistical information about a BTreeDatabase</td>
</tr>
<tr>
<td>ByteOrder</td>
<td>A class to represent the database byte order.</td>
</tr>
<tr>
<td>CacheInfo</td>
<td>A class to represent information about the Berkeley DB cache</td>
</tr>
<tr>
<td>CachePriority</td>
<td>A class to represent cache priority for database pages</td>
</tr>
<tr>
<td>CompactConfig</td>
<td>A class to represent configuration settings for Compact(CompactConfig) and Compact(CompactConfig).</td>
</tr>
<tr>
<td>CompactData</td>
<td>A class for representing compact operation statistics</td>
</tr>
<tr>
<td>Cursor</td>
<td>A class representing database cursors, which allow for traversal of database records.</td>
</tr>
<tr>
<td>CursorConfig</td>
<td>A class representing configuration parameters for Cursor</td>
</tr>
<tr>
<td>Database</td>
<td>A class representing a Berkeley DB database, a base class for access method specific classes.</td>
</tr>
<tr>
<td>DatabaseConfig</td>
<td>A class representing configuration parameters for Database</td>
</tr>
<tr>
<td>DatabaseEntry</td>
<td>A class representing a key or data item in a Berkeley DB database</td>
</tr>
<tr>
<td>DatabaseEnvironment</td>
<td>A class representing a Berkeley DB database environment - a collection including support for some or all of caching, locking, logging and transaction subsystems, as well as databases and log files.</td>
</tr>
<tr>
<td>DatabaseEnvironmentConfig</td>
<td>A class representing configuration parameters for DatabaseEnvironment</td>
</tr>
<tr>
<td>Class/Exception</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>-------------</td>
</tr>
<tr>
<td>DatabaseException</td>
<td>Represents errors that occur during Berkley DB operations.</td>
</tr>
<tr>
<td>DatabaseType</td>
<td>A class representing the supported Berkeley DB access methods.</td>
</tr>
<tr>
<td>DbThreadID</td>
<td>A class representing a unique identifier for a thread of control in a Berkeley DB application. When multiple threads of control are modifying the database, there is normally the potential for deadlock. In Berkeley DB, deadlock is signified by a DeadlockException thrown from the Berkeley DB function. Whenever a Berkeley DB function throws a DeadlockException, the enclosing transaction should be aborted.</td>
</tr>
<tr>
<td>DeadlockException</td>
<td>A class to represent what lock request(s) should be rejected during deadlock resolution.</td>
</tr>
<tr>
<td>DeadlockPolicy</td>
<td>Constants representing error codes returned by the Berkeley DB library.</td>
</tr>
<tr>
<td>ErrorCodes</td>
<td>In-memory logs are configured and no more log buffer space is available.</td>
</tr>
<tr>
<td>ForeignConflictException</td>
<td>A class for traversing the records of a HashDatabase</td>
</tr>
<tr>
<td>FullLogBufferException</td>
<td>A class representing a HashDatabase. The Hash format is an extensible, dynamic hashing scheme.</td>
</tr>
<tr>
<td>HashCursor</td>
<td>A class representing configuration parameters for HashDatabase</td>
</tr>
<tr>
<td>HashDatabase</td>
<td>Statistical information about a HashDatabase</td>
</tr>
<tr>
<td>HashDatabaseConfig</td>
<td>A class representing a join cursor, for use in performing equality or natural joins on secondary indices. For information on how to organize your</td>
</tr>
</tbody>
</table>
JoinCursor

data to use this functionality, see Equality join in the Programmer's Reference Guide.

KeyEmptyException

The requested key/data pair logically exists but was never explicitly created by the application, or that the requested key/data pair was deleted and never re-created. In addition, the Queue access method will throw a KeyEmptyException for records that were created as part of a transaction that was later aborted and never re-created.

A key/data pair was inserted into the database using
PutNoOverwrite(DatabaseEntry, DatabaseEntry) and the key already exists in the database, or using
PutNoDuplicate(DatabaseEntry, DatabaseEntry) or
PutNoDuplicate(DatabaseEntry, DatabaseEntry) and the key/data pair already exists in the database.

KeyExistException

A class representing an estimate of the proportion of keys that are less than, equal to, and greater than a given key.

KeyRange

Values are in the range of 0 to 1; for example, if the field less is 0.05, 5% of the keys in the database are less than the key parameter. The value for equal will be zero if there is no matching key, and will be non-zero otherwise.
LeaseExpiredException

The site's replication master lease has expired.

LockingConfig

A class representing configuration parameters for a DatabaseEnvironment's locking subsystem.

LockingInfo

A class representing the locking options for Berkeley DB operations. If TimeNotGranted is true, database calls timing out based on lock or transaction timeout values will throw a LockNotGrantedException, instead of a DeadlockException.

LockNotGrantedException

If TimeNotGranted is true, database calls timing out based on lock or transaction timeout values will throw a LockNotGrantedException, instead of a DeadlockException.

LockStats

Statistical information about the locking subsystem

LogConfig

A class representing configuration parameters for a DatabaseEnvironment's logging subsystem.

LogStats

Statistical information about the logging subsystem

LSN

A log sequence number, which specifies a unique location in a log file.

MPoolConfig

A class representing configuration parameters for a DatabaseEnvironment's memory pool subsystem.

MPoolFileStats

Statistical information about a file in the memory pool

MPoolStats

Statistical information about the memory pool subsystem

MultipleDatabaseEntry

A class providing access to multiple DatabaseEntry objects.

MultipleKeyDatabaseEntry

A class providing access to multiple key/data pairs.
<table>
<thead>
<tr>
<th>Class/Exception</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MutexConfig</td>
<td>A class representing configuration parameters for a DatabaseEnvironment's mutex subsystem.</td>
</tr>
<tr>
<td>MutexStats</td>
<td>Statistical information about the mutex subsystem. The requested key/data pair did not exist in the database or that start-of-or end-of-file has been reached by a cursor.</td>
</tr>
<tr>
<td>NotFoundException</td>
<td>This version of Berkeley DB is unable to upgrade a given database.</td>
</tr>
<tr>
<td>OldVersionException</td>
<td>A class representing a transaction that must be resolved by the application following Recover(UInt32, Boolean).</td>
</tr>
<tr>
<td>QueueDatabase</td>
<td>A class representing a QueueDatabase. The Queue format supports fast access to fixed-length records accessed sequentially or by logical record number.</td>
</tr>
<tr>
<td>QueueDatabaseConfig</td>
<td>A class representing configuration parameters for QueueDatabase.</td>
</tr>
<tr>
<td>QueueStats</td>
<td>Statistical information about a QueueDatabase.</td>
</tr>
<tr>
<td>RecnoCursor</td>
<td>A class for traversing the records of a RecnoDatabase.</td>
</tr>
<tr>
<td>RecnoDatabase</td>
<td>A class representing a RecnoDatabase. The Recno format supports fixed- or variable-length records, accessed sequentially or by logical record number, and optionally backed by a flat text file.</td>
</tr>
<tr>
<td>RecnoDatabaseConfig</td>
<td>A class representing configuration parameters for RecnoDatabase.</td>
</tr>
<tr>
<td>RecnoStats</td>
<td>Statistical information about a</td>
</tr>
<tr>
<td>Class</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>RecnoDatabase</td>
<td>A class representing configuration parameters for a DatabaseEnvironment’s replication subsystem.</td>
</tr>
<tr>
<td>ReplicationConfig</td>
<td>A class representing configuration parameters for a DatabaseEnvironment’s replication subsystem.</td>
</tr>
<tr>
<td>ReplicationHostAddress</td>
<td>A class representing the address of a replication site used by Berkeley DB HA.</td>
</tr>
<tr>
<td>ReplicationStats</td>
<td>Statistical information about the replication subsystem.</td>
</tr>
<tr>
<td>RepMgrSite</td>
<td>A class representing a replication site used by Replication Manager.</td>
</tr>
<tr>
<td>RepMgrStats</td>
<td>Statistical information about the Replication Manager.</td>
</tr>
<tr>
<td>RepProcMsgResult</td>
<td>A class representing the return value of RepProcessMessage(DatabaseEntry, DatabaseEntry, Int32).</td>
</tr>
<tr>
<td>RunRecoveryException</td>
<td>Berkeley DB has encountered an error it considers fatal to an entire environment. Once a RunRecoveryException has been thrown by any interface, it will be returned from all subsequent Berkeley DB calls made by any threads of control participating in the environment.</td>
</tr>
<tr>
<td>SecondaryBTreeDatabase</td>
<td>A class representing a SecondaryBTreeDatabase. The Btree format is a representation of a sorted, balanced tree structure.</td>
</tr>
<tr>
<td>SecondaryBTreeDatabaseConfig</td>
<td>A class representing configuration parameters for SecondaryBTreeDatabase.</td>
</tr>
<tr>
<td>SecondaryCursor</td>
<td>A class representing database cursors over secondary indexes, which allow for traversal of database records.</td>
</tr>
<tr>
<td>Class</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>SecondaryDatabase</td>
<td>A class representing a secondary Berkeley DB database, a base class for access method specific classes.</td>
</tr>
<tr>
<td>SecondaryDatabaseConfig</td>
<td>A class representing configuration parameters for SecondaryDatabase.</td>
</tr>
<tr>
<td>SecondaryHashDatabase</td>
<td>A class representing a SecondaryHashDatabase. The Hash format is an extensible, dynamic hashing scheme.</td>
</tr>
<tr>
<td>SecondaryHashDatabaseConfig</td>
<td>A class representing configuration parameters for SecondaryHashDatabase.</td>
</tr>
<tr>
<td>SecondaryQueueDatabase</td>
<td>A class representing a SecondaryQueueDatabase. The Queue format supports fast access to fixed-length records accessed sequentially or by logical record number.</td>
</tr>
<tr>
<td>SecondaryQueueDatabaseConfig</td>
<td>A class representing configuration parameters for SecondaryQueueDatabase.</td>
</tr>
<tr>
<td>SecondaryRecnoDatabase</td>
<td>A class representing a SecondaryRecnoDatabase. The Recno format supports fixed- or variable-length records, accessed sequentially or by logical record number, and optionally backed by a flat text file.</td>
</tr>
<tr>
<td>SecondaryRecnoDatabaseConfig</td>
<td>A class representing configuration parameters for RecnoDatabase.</td>
</tr>
<tr>
<td>Sequence</td>
<td>A class that provides an arbitrary number of persistent objects that return an increasing or decreasing sequence of integers.</td>
</tr>
<tr>
<td>SequenceConfig</td>
<td>Configuration properties for a Sequence</td>
</tr>
<tr>
<td>SequenceStats</td>
<td>Statistical information about a Sequence</td>
</tr>
<tr>
<td><strong>Transaction</strong></td>
<td>A class representing Berkeley DB transactions</td>
</tr>
<tr>
<td>-----------------</td>
<td>-----------------------------------------------</td>
</tr>
<tr>
<td><strong>TransactionConfig</strong></td>
<td>A class representing configuration parameters for a Transaction.</td>
</tr>
<tr>
<td><strong>TransactionStats</strong></td>
<td>Statistical information about the transaction subsystem</td>
</tr>
<tr>
<td><strong>VerboseMessages</strong></td>
<td>Enable specific additional informational and debugging messages.</td>
</tr>
<tr>
<td><strong>VerificationException</strong></td>
<td>Thrown by Verify(String, DatabaseConfig) if a database is corrupted, and by Salvage(String, DatabaseConfig) if all key/data pairs in the file may not have been successfully output.</td>
</tr>
<tr>
<td><strong>VersionMismatchException</strong></td>
<td>The version of the Berkeley DB library doesn't match the version that created the database environment.</td>
</tr>
</tbody>
</table>
# Delegates

<table>
<thead>
<tr>
<th>Delegate</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AppendRecordDelegate</td>
<td>A function to call after the record number has been selected but before the data has been stored into the database.</td>
</tr>
<tr>
<td>BTreeCompressDelegate</td>
<td>A function to store a compressed key/data pair into a supplied buffer.</td>
</tr>
<tr>
<td>BTreeDecompressDelegate</td>
<td>A function to decompress a key/data pair from a supplied buffer.</td>
</tr>
<tr>
<td>DatabaseFeedbackDelegate</td>
<td>The application-specified feedback function called to report Berkeley DB operation progress.</td>
</tr>
<tr>
<td>EntryComparisonDelegate</td>
<td>An application-specified comparison function.</td>
</tr>
<tr>
<td>EnvironmentFeedbackDelegate</td>
<td>The application-specified feedback function called to report Berkeley DB operation progress.</td>
</tr>
<tr>
<td>ErrorFeedbackDelegate</td>
<td>The application-specified error reporting function.</td>
</tr>
<tr>
<td>EventNotifyDelegate</td>
<td>The application's event notification function.</td>
</tr>
<tr>
<td>ForeignKeyNullifyDelegate</td>
<td>The application-specified hash function.</td>
</tr>
<tr>
<td>HashFunctionDelegate</td>
<td>The function used to transmit data using the replication application's communication infrastructure.</td>
</tr>
<tr>
<td>ReplicationTransportDelegate</td>
<td>The function that creates the set of secondary keys corresponding to a given primary key and data pair.</td>
</tr>
<tr>
<td>SecondaryKeyGenDelegate</td>
<td>A function which returns a unique identifier pair for a thread of control in</td>
</tr>
</tbody>
</table>
a Berkeley DB application.

SetThreadNameDelegate

A function which returns an identifier pair for a thread of control formatted for display.

ThreadIsAliveDelegate

A function which returns whether the thread of control, identified by info, is still running.
## Enumerations

<table>
<thead>
<tr>
<th>Enumeration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ActiveTransaction::&lt;br&gt;TransactionStatus</td>
<td>The status of an active transaction.</td>
</tr>
<tr>
<td>CreatePolicy</td>
<td>The policy for how to handle database creation.</td>
</tr>
<tr>
<td>Cursor::&lt;br&gt;InsertLocation</td>
<td>Specifies where to place duplicate data elements of the key to which the cursor refers.</td>
</tr>
<tr>
<td>Database::&lt;br&gt;VerifyOperation</td>
<td>Specifies the type of verification to perform</td>
</tr>
<tr>
<td>DatabaseFeedbackEvent</td>
<td>Specifies the database operation whose progress is being reported</td>
</tr>
<tr>
<td>DuplicatesPolicy</td>
<td>Policy for duplicate data items in the database; that is, whether insertion when the key of the key/data pair being inserted already exists in the database will be successful.</td>
</tr>
<tr>
<td>EncryptionAlgorithm</td>
<td>Specifies an algorithm used for encryption and decryption</td>
</tr>
<tr>
<td>EnvironmentFeedbackEvent</td>
<td>Specifies the environment operation whose progress is being reported</td>
</tr>
<tr>
<td>ForeignKeyDeleteAction</td>
<td>Specifies the action to take when deleting a foreign key</td>
</tr>
<tr>
<td>Isolation</td>
<td>Specify the degree of isolation for transactional operations</td>
</tr>
<tr>
<td>NotificationEvent</td>
<td>Specify a Berkeley DB event</td>
</tr>
<tr>
<td>RepProcMsgResult::&lt;br&gt;ProcMsgResult</td>
<td>The result of processing an incoming replication message.</td>
</tr>
<tr>
<td></td>
<td>Specifies the log flushing</td>
</tr>
</tbody>
</table>
TransactionConfig:::LogFlush behavior on transaction commit

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Acknowledgment of replication messages which are necessary for "permanent" records. The current implementation requires all sites in a replication group to configure the same acknowledgement policy.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public class AckPolicy

Visual Basic (Declaration)

Public Class AckPolicy

Visual C++

public ref class AckPolicy
Inheritance Hierarchy

System..::.Object
BerkeleyDB..::.AckPolicy
See Also

AckPolicy Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `AckPolicy` type exposes the following members.
### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a <a href="#">String</a> that represents the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
</tbody>
</table>
## Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALL</td>
<td>The master should wait until all replication clients have acknowledged each permanent replication message.</td>
</tr>
<tr>
<td>ALL PEERS</td>
<td>The master should wait until all electable peers have acknowledged each permanent replication message (where &quot;electable peer&quot; means a client capable of being subsequently elected master of the replication group).</td>
</tr>
<tr>
<td>NONE</td>
<td>The master should not wait for any client replication message acknowledgments.</td>
</tr>
<tr>
<td>ONE</td>
<td>The master should wait until at least one client site has acknowledged each permanent replication message.</td>
</tr>
<tr>
<td>ONE PEER</td>
<td>The master should wait until at least one electable peer has acknowledged each permanent replication message (where &quot;electable peer&quot; means a client capable of being subsequently elected master of the replication group).</td>
</tr>
<tr>
<td>QUORUM</td>
<td>The master should wait until it has received acknowledgements from the minimum number of electable peers sufficient to ensure that the effect of the permanent record remains durable if an election is held (where &quot;electable peer&quot; means a client capable of being subsequently elected master of the replication group).</td>
</tr>
</tbody>
</table>

This is the default acknowledgement policy.
See Also

AckPolicy Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `AckPolicy` type exposes the following members.
<table>
<thead>
<tr>
<th><strong>Name</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>ALL</td>
<td>The master should wait until all replication clients have acknowledged each permanent replication message.</td>
</tr>
<tr>
<td>ALL_PEERS</td>
<td>The master should wait until all electable peers have acknowledged each permanent replication message (where &quot;electable peer&quot; means a client capable of being subsequently elected master of the replication group).</td>
</tr>
<tr>
<td>NONE</td>
<td>The master should not wait for any client replication message acknowledgments.</td>
</tr>
<tr>
<td>ONE</td>
<td>The master should wait until at least one client site has acknowledged each permanent replication message.</td>
</tr>
<tr>
<td>ONE_PEER</td>
<td>The master should wait until at least one electable peer has acknowledged each permanent replication message (where &quot;electable peer&quot; means a client capable of being subsequently elected master of the replication group).</td>
</tr>
<tr>
<td>QUORUM</td>
<td>The master should wait until it has received acknowledgements from the minimum number of electable peers sufficient to ensure that the effect of the permanent record remains durable if an election is held (where &quot;electable peer&quot; means a client capable of being subsequently elected master of the replication group). This is the default acknowledgement policy.</td>
</tr>
</tbody>
</table>
See Also

 AckPolicy Class
 BerkeleyDB Namespace

 Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The master should wait until all replication clients have acknowledged each permanent replication message.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

**C#**

```csharp
public static AckPolicy ALL
```

**Visual Basic (Declaration)**

```vbnet
Public Shared ALL As AckPolicy
```

**Visual C++**

```cpp
public:
static AckPolicy^ ALL
```
See Also

AckPolicy Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The master should wait until all electable peers have acknowledged each permanent replication message (where "electable peer" means a client capable of being subsequently elected master of the replication group).

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public static AckPolicy ALL_PEERS

Visual Basic (Declaration)

Public Shared ALL_PEERS As AckPolicy

Visual C++

public:
static AckPolicy^ ALL_PEERS
See Also

AckPolicy Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
AckPolicy.....NONE Field

AckPolicy Class  See Also

The master should not wait for any client replication message acknowledgments.

Namespace:  BerkeleyDB
Syntax

C#

public static AckPolicy NONE

Visual Basic (Declaration)

Public Shared NONE As AckPolicy

Visual C++

public:
static AckPolicy^ NONE
See Also

AckPolicy Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Acknowledging Messages

The master should wait until at least one client site has acknowledged each permanent replication message.

**Namespace:** `BerkeleyDB`

**Assembly:** `libdb_dotnet48` (in `libdb_dotnet48.dll`) Version: 4.8.24.0
Syntax

C#
public static AckPolicy ONE

Visual Basic (Declaration)
Public Shared ONE As AckPolicy

Visual C++
public:
static AckPolicy^ ONE
See Also

AckPolicy Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The master should wait until at least one electable peer has acknowledged each permanent replication message (where "electable peer" means a client capable of being subsequently elected master of the replication group).

Namespace: BerkeleyDB
Syntax

C#

public static AckPolicy ONE_PEER

Visual Basic (Declaration)

Public Shared ONE_PEER As AckPolicy

Visual C++

public:
static AckPolicy^ ONE_PEER
See Also

AckPolicy Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The master should wait until it has received acknowledgements from the minimum number of electable peers sufficient to ensure that the effect of the permanent record remains durable if an election is held (where "electable peer" means a client capable of being subsequently elected master of the replication group). This is the default acknowledgement policy.

Namespace: BerkeleyDB
Syntax

C#

public static AckPolicy QUORUM

Visual Basic (Declaration)

Public Shared QUORUM As AckPolicy

Visual C++

public:
static AckPolicy^ QUORUM
See Also

AckPolicy Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The AckPolicy type exposes the following members.
### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <em>Object</em> is equal to the current <em>Object</em>. (Inherited from <em>Object</em>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <em>Object</em>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <em>Type</em> of the current instance. (Inherited from <em>Object</em>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a <em>String</em> that represents the current <em>Object</em>. (Inherited from <em>Object</em>.)</td>
</tr>
</tbody>
</table>
See Also

AckPolicy Class  
BerkeleyDB Namespace  

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The ActiveTransaction class describes a currently active transaction.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

**C#**

public class ActiveTransaction

**Visual Basic (Declaration)**

Public Class ActiveTransaction

**Visual C++**

public ref class ActiveTransaction
Inheritance Hierarchy

System..:::Object
BerkeleyDB..:::ActiveTransaction
See Also

ActiveTransaction Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `ActiveTransaction` type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a <a href="#">String</a> that represents the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Begun</td>
<td>The current log sequence number when the transaction was begun.</td>
</tr>
<tr>
<td>BufferCopiesInCache</td>
<td>The number of MVCC buffer copies created by this transaction that remain in cache.</td>
</tr>
<tr>
<td>GlobalID</td>
<td>If the transaction is a prepare transaction, the transaction's Global ID. Otherwise, the GlobalID contents are undefined.</td>
</tr>
<tr>
<td>ID</td>
<td>The transaction ID of the transaction.</td>
</tr>
<tr>
<td>Name</td>
<td>If a name was specified for the transaction, up to the first 50 bytes of that name.</td>
</tr>
<tr>
<td>ParentID</td>
<td>The transaction ID of the parent transaction (or 0, if no parent).</td>
</tr>
<tr>
<td>ProcessID</td>
<td>The process ID of the originator of the transaction.</td>
</tr>
<tr>
<td>SnapshotReads</td>
<td>The log sequence number of reads for snapshot transactions.</td>
</tr>
<tr>
<td>Status</td>
<td>Status of the transaction.</td>
</tr>
<tr>
<td>ThreadID</td>
<td>The thread of control ID of the originator of the transaction.</td>
</tr>
</tbody>
</table>
See Also

ActiveTransaction Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `ActiveTransaction` type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a <a href="#">String</a> that represents the current <a href="#">Object</a>.               (Inherited from <a href="#">Object</a>.)</td>
</tr>
</tbody>
</table>
See Also

[ActiveTransaction Class]
[BerkeleyDB Namespace]

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
ActiveTransaction Properties

The **ActiveTransaction** type exposes the following members.
### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Begun</td>
<td>The current log sequence number when the transaction was begun.</td>
</tr>
<tr>
<td>BufferCopiesInCache</td>
<td>The number of MVCC buffer copies created by this transaction that remain in cache.</td>
</tr>
<tr>
<td>GlobalID</td>
<td>If the transaction is a prepare transaction, the transaction's Global ID. Otherwise, the GlobalID contents are undefined.</td>
</tr>
<tr>
<td>ID</td>
<td>The transaction ID of the transaction.</td>
</tr>
<tr>
<td>Name</td>
<td>If a name was specified for the transaction, up to the first 50 bytes of that name.</td>
</tr>
<tr>
<td>ParentID</td>
<td>The transaction ID of the parent transaction (or 0, if no parent).</td>
</tr>
<tr>
<td>ProcessID</td>
<td>The process ID of the originator of the transaction.</td>
</tr>
<tr>
<td>SnapshotReads</td>
<td>The log sequence number of reads for snapshot transactions.</td>
</tr>
<tr>
<td>Status</td>
<td>Status of the transaction.</td>
</tr>
<tr>
<td>ThreadID</td>
<td>The thread of control ID of the originator of the transaction.</td>
</tr>
</tbody>
</table>
See Also

ActiveTransaction Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
ActiveTransaction.Begun Property

The current log sequence number when the transaction was begun.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public LSN Begun { get; }

Visual Basic (Declaration)

Public ReadOnly Property Begun As LSN

Visual C++

public:
property LSN^ Begun {
    LSN^ get ();
}


See Also

ActiveTransaction Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The number of MVCC buffer copies created by this transaction that remain in cache.

Namespace: BerkeleyDB
Syntax

C#

public uint BufferCopiesInCache { get; }

Visual Basic (Declaration)

Public ReadOnly Property BufferCopiesInCache AsUInteger

Visual C++

public:
property unsigned int BufferCopiesInCache {
    unsigned int get ();
}
See Also

ActiveTransaction Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If the transaction is a prepare transaction, the transaction's Global ID. Otherwise, the GlobalID contents are undefined.

Namespace: BerkeleyDB
Syntax

C#

public byte[] GlobalID { get; }

Visual Basic (Declaration)

Public ReadOnly Property GlobalID As Byte()

Visual C++

public:
property array<unsigned char>^ GlobalID {
    array<unsigned char>^ get ();
}

See Also

ActiveTransaction Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The transaction ID of the transaction.

**Namespace:**  [BerkeleyDB](#)

**Assembly:**  `libdb_dotnet48` (in `libdb_dotnet48.dll`)  Version: 4.8.24.0
Syntax

C#

public uint ID { get; }

Visual Basic (Declaration)

Public ReadOnly Property ID AsUInteger

Visual C++

public:
property unsigned int ID {
    unsigned int get ();
}
See Also

ActiveTransaction Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If a name was specified for the transaction, up to the first 50 bytes of that name.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll)  
**Version:** 4.8.24.0
Syntax

**C#**

```csharp
public string Name { get; }
```

**Visual Basic (Declaration)**

```vbnet
Public ReadOnly Property Name As String
```

**Visual C++**

```cpp
public:
property String^ Name {
    String^ get ();
}
```
See Also

ActiveTransaction Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The transaction ID of the parent transaction (or 0, if no parent).

Namespace:  BerkeleyDB
Syntax

C#

public uint ParentID { get; }

Visual Basic (Declaration)

Public ReadOnly Property ParentID AsUInteger

Visual C++

public:
property unsigned int ParentID {
    unsigned int get ();
}

See Also

ActiveTransaction Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The process ID of the originator of the transaction.

**Namespace:**  BerkeleyDB  
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public int ProcessID { get; }

Visual Basic (Declaration)

Public ReadOnly Property ProcessID As Integer

Visual C++

public:
property int ProcessID {
    int get ();
}

See Also

ActiveTransaction Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The log sequence number of reads for snapshot transactions.

**Namespace:** BerkeleyDB
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public LSN SnapshotReads { get; }

Visual Basic (Declaration)

Public ReadOnly Property SnapshotReads As LSN

Visual C++

public:
property LSN^ SnapshotReads {
    LSN^ get ();
}

See Also

ActiveTransaction Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Status of the transaction.

Namespace:  BerkeleyDB
Syntax

**C#**

```csharp
public ActiveTransaction...:::TransactionStatus Status { get; }
```

**Visual Basic (Declaration)**

```vbnet
Public ReadOnly Property Status As ActiveTransaction...:::TransactionStatus
```

**Visual C++**

```cpp
public:
property ActiveTransaction...:::TransactionStatus Status { 
    ActiveTransaction...:::TransactionStatus get ();
}
```
See Also

ActiveTransaction Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The thread of control ID of the originator of the transaction.

Namespace:  BerkeleyDB  
Syntax

C#

public uint ThreadID { get; }

Visual Basic (Declaration)

Public ReadOnly Property ThreadID AsUInteger

Visual C++

public:
property unsigned int ThreadID {
    unsigned int get ();
}

See Also

ActiveTransaction Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
ActiveTransaction...:::TransactionStatus Enumeration

ActiveTransaction Class  See Also

The status of an active transaction.

Namespace:  BerkeleyDB
Syntax

C#

public enum TransactionStatus

Visual Basic (Declaration)

Public Enumeration TransactionStatus

Visual C++

public enum class TransactionStatus
## Members

<table>
<thead>
<tr>
<th>Member name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABORTED</td>
<td>The transaction has been aborted</td>
</tr>
<tr>
<td>COMMITTED</td>
<td>The transaction has been committed</td>
</tr>
<tr>
<td>PREPARED</td>
<td>The transaction has been prepared</td>
</tr>
<tr>
<td>RUNNING</td>
<td>The transaction is running</td>
</tr>
</tbody>
</table>
See Also

BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
AppendRecordDelegate Delegate

See Also

A function to call after the record number has been selected but before the data has been stored into the database.

Namespace: BerkeleyDB
### Syntax

**C#**

```csharp
public delegate void AppendRecordDelegate(
    DatabaseEntry data,
    uint recno
)
```

**Visual Basic (Declaration)**

```vbnet
Public Delegate Sub AppendRecordDelegate ( _
    data As DatabaseEntry, _
    recno As UInteger _
)
```

**Visual C++**

```c++
public delegate void AppendRecordDelegate(
    DatabaseEntry^ data,
    unsigned int recno
)
```

### Parameters

**data**

Type: `BerkeleyDB::DatabaseEntry`

The data to be stored.

**recno**

Type: `System::UInt32`

The generated record number.
See Also

BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
A secondary index has been corrupted. This is likely the result of an application operating on related databases without first associating them.

Namespace: BerkeleyDB
Syntax

C#

public class BadSecondaryException : DatabaseException

Visual Basic (Declaration)

Public Class BadSecondaryException
    Inherits DatabaseException

Visual C++

public ref class BadSecondaryException : public DatabaseException
Inheritance Hierarchy

System:::Object
  System:::Exception
    BerkeleyDB:::DatabaseException
      BerkeleyDB:::BadSecondaryException
See Also

BadSecondaryException Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The *BadSecondaryException* type exposes the following members.
### Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BadSecondaryException</td>
<td>Initialize a new instance of the BadSecondaryException</td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified Object is equal to the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>GetBaseException</strong></td>
<td>When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions. (Inherited from Exception.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>GetObjectData</strong></td>
<td>When overridden in a derived class, sets the SerializationInfo with information about the exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the runtime type of the current instance. (Inherited from Exception.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Creates and returns a string representation of the current exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| ErrorCode | The underlying error code from the Berkeley DB C library.  
                  (Inherited from DatabaseException.) |
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data</strong></td>
<td>Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from <a href="#">Exception</a>.)</td>
</tr>
<tr>
<td><strong>HelpLink</strong></td>
<td>Gets or sets a link to the help file associated with this exception. (Inherited from <a href="#">Exception</a>.)</td>
</tr>
<tr>
<td><strong>InnerException</strong></td>
<td>Gets the <a href="#">Exception</a> instance that caused the current exception. (Inherited from <a href="#">Exception</a>.)</td>
</tr>
<tr>
<td><strong>Message</strong></td>
<td>Gets a message that describes the current exception. (Inherited from <a href="#">Exception</a>.)</td>
</tr>
<tr>
<td><strong>Source</strong></td>
<td>Gets or sets the name of the application or the object that causes the error. (Inherited from <a href="#">Exception</a>.)</td>
</tr>
<tr>
<td><strong>StackTrace</strong></td>
<td>Gets a string representation of the frames on the call stack at the time the current exception was thrown. (Inherited from <a href="#">Exception</a>.)</td>
</tr>
<tr>
<td><strong>TargetSite</strong></td>
<td>Gets the method that throws the current exception. (Inherited from <a href="#">Exception</a>.)</td>
</tr>
</tbody>
</table>
See Also

BadSecondaryException Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Initialize a new instance of the BadSecondaryException

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public BadSecondaryException()

Visual Basic (Declaration)

Public Sub New

Visual C++

public:
BadSecondaryException()
See Also

BadSecondaryException Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The **BadSecondaryException** type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ErrorCode</td>
<td>The underlying error code from the Berkeley DB C library. (Inherited from <a href="https://example.com">DatabaseException</a>).</td>
</tr>
</tbody>
</table>
See Also

BadSecondaryException Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `BadSecondaryException` type exposes the following members.
### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="/en/library/system#object">Object</a> is equal to the current <a href="/en/library/system#object">Object</a>. When overridden in a derived class, returns the <a href="/en/library/system#exception">Exception</a> that is the root cause of one or more subsequent exceptions. (Inherited from <a href="/en/library/system#exception">Exception</a>.)</td>
</tr>
<tr>
<td><strong>GetBaseException</strong></td>
<td>Serves as a hash function for a particular type. When overridden in a derived class, sets the <a href="/en/library/system#serializationinfo">SerializationInfo</a> with information about the exception. (Inherited from <a href="/en/library/system#exception">Exception</a>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Gets the runtime type of the current instance. (Inherited from <a href="/en/library/system#exception">Exception</a>.)</td>
</tr>
<tr>
<td><strong>GetObjectData</strong></td>
<td>Creates and returns a string representation of the current exception. (Inherited from <a href="/en/library/system#exception">Exception</a>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td></td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td></td>
</tr>
</tbody>
</table>
See Also

BadSecondaryException Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The **BadSecondaryException** type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| Data        | Gets a collection of key/value pairs that provide additional user-defined information about the exception.  
(Inherited from Exception.) |
| HelpLink    | Gets or sets a link to the help file associated with this exception.  
(Inherited from Exception.) |
| InnerException | Gets the Exception instance that caused the current exception.  
(Inherited from Exception.) |
| Message     | Gets a message that describes the current exception.  
(Inherited from Exception.) |
| Source      | Gets or sets the name of the application or the object that causes the error.  
(Inherited from Exception.) |
| StackTrace  | Gets a string representation of the frames on the call stack at the time the current exception was thrown.  
(Inherited from Exception.) |
| TargetSite  | Gets the method that throws the current exception.  
(Inherited from Exception.) |
See Also

BadSecondaryException Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The abstract base class from which all cursor classes inherit.

Cursors may span threads, but only serially, that is, the application must serialize access to the cursor handle.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public abstract class BaseCursor : IDisposable,
      IEnumerable<KeyValuePair<DatabaseEntry, DatabaseEntry>>, IE...

Visual Basic (Declaration)

Public MustInherit Class BaseCursor _
      Implements IDisposable, IEnumerable(Of KeyValuePair(Of Date...

Visual C++

public ref class BaseCursor abstract : IDisposable,
      IEnumerable<KeyValuePair<DatabaseEntry^, DatabaseEntry^>>, I...
Inheritance Hierarchy

**System..::.Object**
BerkeleyDB..::.BaseCursor
  BerkeleyDB..::.Cursor
  BerkeleyDB..::.SecondaryCursor
See Also

BaseCursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The **BaseCursor** type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Close</strong></td>
<td>Discard the cursor. It is possible for the Close() method to throw a <a href="#">DeadlockException</a>, signaling that any enclosing transaction should be aborted. If the application is already intending to abort the transaction, this error should be ignored, and the application should proceed. After Close has been called, regardless of its result, the object may not be used again.</td>
</tr>
<tr>
<td><strong>Compare</strong></td>
<td>Compare this cursor's position to another's.</td>
</tr>
<tr>
<td><strong>Count</strong></td>
<td>Returns a count of the number of data items for the key to which the cursor refers.</td>
</tr>
<tr>
<td><strong>Delete</strong></td>
<td>Delete the key/data pair to which the cursor refers. When called on a SecondaryCursor, delete the key/data pair from the primary database and all secondary indices. The cursor position is unchanged after a delete, and subsequent calls to cursor functions expecting the cursor to refer to an existing key will fail.</td>
</tr>
<tr>
<td><strong>Dispose</strong></td>
<td>Release the resources held by this object, and close the cursor if it's still open.</td>
</tr>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="#">Object</a> is equal to the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>GetEnumerator</strong></td>
<td>Returns an enumerator that iterates through the cursor.</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type.</td>
</tr>
</tbody>
</table>
- **GetType** (Inherited from **Object**.)
  Gets the **Type** of the current instance.
  (Inherited from **Object**.)

- **ToString** (Inherited from **Object**.)
  Returns a **String** that represents the current **Object**.
  (Inherited from **Object**.)
See Also

BaseCursor Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `BaseCursor` type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Close</strong></td>
<td>Discard the cursor.  It is possible for the Close() method to throw a <a href="#">DeadlockException</a>, signaling that any enclosing transaction should be aborted. If the application is already intending to abort the transaction, this error should be ignored, and the application should proceed. After Close has been called, regardless of its result, the object may not be used again.</td>
</tr>
<tr>
<td><strong>Compare</strong></td>
<td>Compare this cursor's position to another's.</td>
</tr>
<tr>
<td><strong>Count</strong></td>
<td>Returns a count of the number of data items for the key to which the cursor refers.</td>
</tr>
<tr>
<td><strong>Delete</strong></td>
<td>Delete the key/data pair to which the cursor refers. When called on a SecondaryCursor, delete the key/data pair from the primary database and all secondary indices. The cursor position is unchanged after a delete, and subsequent calls to cursor functions expecting the cursor to refer to an existing key will fail.</td>
</tr>
<tr>
<td><strong>Dispose</strong></td>
<td>Release the resources held by this object, and close the cursor if it's still open.</td>
</tr>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetEnumerator</strong></td>
<td>Returns an enumerator that iterates through the cursor.</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type.</td>
</tr>
</tbody>
</table>
- **GetType**
  Gets the **Type** of the current instance.
  (Inherited from **Object**.)

- **ToString**
  Returns a **String** that represents the current **Object**.
  (Inherited from **Object**.)
See Also

BaseCursor Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Discard the cursor.

It is possible for the Close() method to throw a `DeadlockException`, signaling that any enclosing transaction should be aborted. If the application is already intending to abort the transaction, this error should be ignored, and the application should proceed.

After Close has been called, regardless of its result, the object may not be used again.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

**C#**

```csharp
public void Close()
```

**Visual Basic (Declaration)**

```vbnet
Public Sub Close
```

**Visual C++**

```cpp
public:
    void Close()
```
### Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>BerkeleyDB::DeadlockException</td>
<td></td>
</tr>
</tbody>
</table>
See Also

BaseCursor Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Compare this cursor's position to another's.

**Namespace:**  BerkeleyDB  
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
C#

public bool Compare(
    Cursor compareTo
)

Visual Basic (Declaration)

Public Function Compare ( _
    compareTo As Cursor _
) As Boolean

Visual C++

public:
    bool Compare(
        Cursor^ compareTo
    )

Parameters

compareTo

Type: BerkeleyDB::Cursor
    The cursor with which to compare.

Return Value

True if both cursors point to the same item, false otherwise.
See Also

BaseCursor Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Returns a count of the number of data items for the key to which the cursor refers.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public uint Count()

Visual Basic (Declaration)

Public Function Count As UInteger

Visual C++

public:
unsigned int Count()

Return Value

A count of the number of data items for the key to which the cursor refers.
See Also

BaseCursor Class  
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Delete the key/data pair to which the cursor refers.

When called on a SecondaryCursor, delete the key/data pair from the primary database and all secondary indices.

The cursor position is unchanged after a delete, and subsequent calls to cursor functions expecting the cursor to refer to an existing key will fail.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

**C#**

```csharp
public void Delete()
```

**Visual Basic (Declaration)**

```vbnet
Public Sub Delete
```

**Visual C++**

```cpp
public:
void Delete()
```
<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>BerkeleyDB::KeyEmptyException</code></td>
<td>Thrown if the element has already been deleted.</td>
</tr>
</tbody>
</table>
See Also

BaseCursor Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
BaseCursor Class  See Also

Release the resources held by this object, and close the cursor if it's still open.

Namespace:  BerkeleyDB
Syntax

C#

public void Dispose()

Visual Basic (Declaration)

Public Sub Dispose

Visual C++

public:
virtual void Dispose() sealed

Implements

IDisposable::..Dispose()()
See Also

BaseCursor Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Returns an enumerator that iterates through the cursor.

**Namespace:**  BerkeleyDB
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public virtual IEnumerator<
KeyValuePair<
DatabaseEntry,
DatabaseEntry>

Visual Basic (Declaration)

Public Overridable Function GetEnumerator As IEnumera
tor(Of KeyValue

Visual C++

public:
virtual IEnumerator<
KeyValuePair<
DatabaseEntry^,
DatabaseEntry^>

ReturnValue

An enumerator for the cursor.

Implements

IEnumerable<(Of (T)>),...GetEnumerator())()
See Also

BaseCursor Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The base class from which all database classes inherit

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

**C#**

public class BaseDatabase : IDisposable

**Visual Basic (Declaration)**

Public Class BaseDatabase _
    Implements IDisposable

**Visual C++**

public ref class BaseDatabase : IDisposable
Inheritance Hierarchy

System..::.Object
BerkeleyDB..::.BaseDatabase
   BerkeleyDB..::.Database
   BerkeleyDB..::.SecondaryDatabase
See Also

BaseDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `BaseDatabase` type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overloaded.</td>
<td>Although closing a database will close any open cursors, it is recommended that applications explicitly close all their Cursor objects before closing the database. The reason why is that when the cursor is explicitly closed, the memory allocated for it is reclaimed; however, this will not happen if you close a database while cursors are still opened.</td>
</tr>
<tr>
<td><strong>Close</strong></td>
<td>The same rule, for the same reasons, hold true for Transaction objects. Simply make sure you resolve all your transaction objects before closing your database handle.</td>
</tr>
<tr>
<td>Overloaded.</td>
<td>Because key/data pairs are cached in memory, applications should make a point to always either close database handles or sync their data to disk (using Sync() before exiting, to ensure that any data cached in main memory are reflected in the underlying file system.</td>
</tr>
<tr>
<td>Overloaded.</td>
<td>When called on a database that is the primary database for a secondary index, the primary database should be closed only after all secondary indices referencing it have been closed.</td>
</tr>
<tr>
<td>Overloaded.</td>
<td>When multiple threads are using the object concurrently, only a single thread may call the Close method.</td>
</tr>
<tr>
<td>Overloaded.</td>
<td>The object may not be accessed again after Close is called, regardless of its outcome.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Delete</strong></td>
<td>Overloaded. Release the resources held by this object, and close the database if it's still open.</td>
</tr>
<tr>
<td><strong>Dispose</strong></td>
<td>Determines whether the specified Object is equal to the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>Equals</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>Exists</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>Get</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>GetBoth</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the Type of the current instance. (Inherited from Object.) Overloaded.</td>
</tr>
<tr>
<td><strong>PrintFastStats</strong></td>
<td>The statistical information is described by the BTreeStats, HashStats, QueueStats, and RecnoStats classes. Overloaded.</td>
</tr>
<tr>
<td><strong>PrintStats</strong></td>
<td>The statistical information is described by the BTreeStats, HashStats, QueueStats, and RecnoStats classes. Overloaded.</td>
</tr>
</tbody>
</table>

Applications should never remove databases with open DB handles, or in the case of removing a file, when any database in the file has an open handle. For example, some architectures do not permit the removal of files with open system handles. On these architectures, attempts to remove databases currently in use by any thread of control in the system may fail.

Remove should not be called if the remove is intended to be transactionally safe; RemoveDB(String, Boolean) should be used instead.

Overloaded.
Applications should not rename databases that are currently in use. If an underlying file is being renamed and logging is currently enabled in the database environment, no database in the file may be open when Rename is called. In particular, some architectures do not permit renaming files with open handles. On these architectures, attempts to rename databases that are currently in use by any thread of control in the system may fail.

Rename should not be called if the rename is intended to be transactionally safe; RenameDB(String, String, Boolean) should be used instead.

- **Sync**
  Flush any cached information to disk.

- **ToString**
  Returns a String that represents the current Object. (Inherited from Object.)

- **Truncate**
  Overloaded.
  When called on a database configured with secondary indices, Truncate will truncate the primary database and all secondary indices. A count of the records discarded from the primary database is returned.
### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AutoCommit</td>
<td>If true, all database modification operations based on this object will be transactionally protected.</td>
</tr>
<tr>
<td>CacheSize</td>
<td>The size of the shared memory buffer pool -- that is, the cache.</td>
</tr>
<tr>
<td>Creation</td>
<td>The CreatePolicy with which this database was opened.</td>
</tr>
<tr>
<td>DatabaseName</td>
<td>The name of this database, if it has one.</td>
</tr>
<tr>
<td>DoChecksum</td>
<td>If true, do checksum verification of pages read into the cache from the backing filestore.</td>
</tr>
<tr>
<td>EncryptAlgorithm</td>
<td>The algorithm used by the Berkeley DB library to perform encryption and decryption.</td>
</tr>
<tr>
<td>Encrypted</td>
<td>If true, encrypt all data stored in the database.</td>
</tr>
<tr>
<td>Endianness</td>
<td>The database byte order.</td>
</tr>
<tr>
<td>ErrorFeedback</td>
<td>The mechanism for reporting detailed error messages to the application.</td>
</tr>
<tr>
<td>ErrorPrefix</td>
<td>The prefix string that appears before error messages issued by Berkeley DB.</td>
</tr>
<tr>
<td>Feedback</td>
<td>Monitor progress within long running operations.</td>
</tr>
<tr>
<td>FileName</td>
<td>The filename of this database, if it has one.</td>
</tr>
<tr>
<td>FreeThreaded</td>
<td>If true, the object is free-threaded; that is, concurrently usable by multiple threads in the address space.</td>
</tr>
<tr>
<td>HasMultiple</td>
<td>If true, the object references a physical file supporting multiple databases.</td>
</tr>
<tr>
<td>InHostOrder</td>
<td>If true, the underlying database files were created on an architecture of the same byte order as the current one. This information may be used to determine whether application data needs to be adjusted for this architecture or not.</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>-------------</td>
</tr>
<tr>
<td><strong>NoMMap</strong></td>
<td>If true, this database is not mapped into process memory. See <a href="#">MMapSize</a> for further information.</td>
</tr>
<tr>
<td><strong>NonDurableTxns</strong></td>
<td>If true, Berkeley DB will not write log records for this database.</td>
</tr>
<tr>
<td><strong>Pagesize</strong></td>
<td>The database's current page size.</td>
</tr>
<tr>
<td><strong>Priority</strong></td>
<td>The cache priority for pages referenced by this object. If true, this database has been opened for reading only. Any attempt to modify items in the database will fail, regardless of the actual permissions of any underlying files.</td>
</tr>
<tr>
<td><strong>ReadOnly</strong></td>
<td>If true, this database supports transactional read operations with degree 1 isolation. Read operations on the database may request the return of modified but not yet committed data.</td>
</tr>
<tr>
<td><strong>ReadUncommitted</strong></td>
<td>If true, this database has been opened in a transactional mode.</td>
</tr>
<tr>
<td><strong>Transactional</strong></td>
<td>If true, the underlying file was physically truncated upon open, discarding all previous databases it might have held.</td>
</tr>
<tr>
<td><strong>Truncated</strong></td>
<td>The type of the underlying access method (and file format). This value may be used to determine the type of the database after an <a href="#">Open(String, DatabaseConfig)</a>.</td>
</tr>
<tr>
<td><strong>Type</strong></td>
<td>If true, the database was opened with support for multiversion concurrency control.</td>
</tr>
</tbody>
</table>
See Also

**BaseDatabase Class**

**BerkeleyDB Namespace**

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The **BaseDatabase** type exposes the following members.
### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Close</td>
<td>Although closing a database will close any open cursors, it is recommended that applications explicitly close all their Cursor objects before closing the database. The reason why is that when the cursor is explicitly closed, the memory allocated for it is reclaimed; however, this will not happen if you close a database while cursors are still opened. The same rule, for the same reasons, hold true for Transaction objects. Simply make sure you resolve all your transaction objects before closing your database handle.</td>
</tr>
<tr>
<td>Cursor</td>
<td>Because key/data pairs are cached in memory, applications should make a point to always either close database handles or sync their data to disk (using Sync() before exiting, to ensure that any data cached in main memory are reflected in the underlying file system.</td>
</tr>
<tr>
<td></td>
<td>When called on a database that is the primary database for a secondary index, the primary database should be closed only after all secondary indices referencing it have been closed.</td>
</tr>
<tr>
<td></td>
<td>When multiple threads are using the object concurrently, only a single thread may call the Close method.</td>
</tr>
<tr>
<td></td>
<td>The object may not be accessed again after Close is called, regardless of its outcome.</td>
</tr>
</tbody>
</table>
Delete Overloaded.

Dispose Release the resources held by this object, and close the database if it's still open.
Determined whether the specified `Object` is equal to the current `Object`.
(Inherited from `Object`.)

Equals Overloaded.

Exists Overloaded.

Get Overloaded.

GetBoth Overloaded.

GetHashCode Serves as a hash function for a particular type.
(Inherited from `Object`.)

GetType Gets the `Type` of the current instance.
(Inherited from `Object`.)
Overloaded.

PrintFastStats The statistical information is described by the `BTreeStats`, `HashStats`, `QueueStats`, and `RecnoStats` classes.
Overloaded.

PrintStats The statistical information is described by the `BTreeStats`, `HashStats`, `QueueStats`, and `RecnoStats` classes.
Overloaded.

Applications should never remove databases with open DB handles, or in the case of removing a file, when any database in the file has an open handle. For example, some architectures do not permit the removal of files with open system handles. On these architectures, attempts to remove databases currently in use by any thread of control in the system may fail.

Remove should not be called if the remove is intended to be transactionally safe; `RemoveDB(String, Boolean)` should be used instead.

Overloaded.
Applications should not rename databases that are currently in use. If an underlying file is being renamed and logging is currently enabled in the database environment, no database in the file may be open when Rename is called. In particular, some architectures do not permit renaming files with open handles. On these architectures, attempts to rename databases that are currently in use by any thread of control in the system may fail.

Rename should not be called if the rename is intended to be transactionally safe; RenameDB(String, String, Boolean) should be used instead.

Sync
Flush any cached information to disk.

ToString
Returns a String that represents the current Object. (Inherited from Object.)

Truncate
Overloaded.
When called on a database configured with secondary indices, Truncate will truncate the primary database and all secondary indices. A count of the records discarded from the primary database is returned.
See Also

BaseDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Although closing a database will close any open cursors, it is recommended that applications explicitly close all their Cursor objects before closing the database. The reason why is that when the cursor is explicitly closed, the memory allocated for it is reclaimed; however, this will not happen if you close a database while cursors are still opened.

The same rule, for the same reasons, hold true for Transaction objects. Simply make sure you resolve all your transaction objects before closing your database handle.

Because key/data pairs are cached in memory, applications should make a point to always either close database handles or sync their data to disk (using `Sync()`) before exiting, to ensure that any data cached in main memory are reflected in the underlying file system.

When called on a database that is the primary database for a secondary index, the primary database should be closed only after all secondary indices referencing it have been closed.

When multiple threads are using the object concurrently, only a single thread may call the Close method.

The object may not be accessed again after Close is called, regardless of its outcome.
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Close()()</td>
<td>Flush any cached database information to disk, close any open Cursor() objects, free any allocated resources, and close any underlying files. Optionally flush any cached database information to disk, close any open Cursor() objects, free any allocated resources, and close any underlying files.</td>
</tr>
<tr>
<td>Close(Boolean)</td>
<td></td>
</tr>
</tbody>
</table>

Flush any cached database information to disk, close any open Cursor() objects, free any allocated resources, and close any underlying files. Optionally flush any cached database information to disk, close any open Cursor() objects, free any allocated resources, and close any underlying files.
See Also

BaseDatabase Class
BaseDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Flush any cached database information to disk, close any open Cursor objects, free any allocated resources, and close any underlying files.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public void Close()

Visual Basic (Declaration)

Public Sub Close

Visual C++

public:
void Close()
See Also

BaseDatabase Class
Close Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Optionally flush any cached database information to disk, close any open
`Cursor()` objects, free any allocated resources, and close any underlying files.

**Namespace:** BerkeleyDB

**Assembly:** `libdb_dotnet48` (in `libdb_dotnet48.dll`) Version: 4.8.24.0
Syntax

C#

public void Close(
    bool sync
)

Visual Basic (Declaration)

Public Sub Close (_
    sync As Boolean _
)

Visual C++

public:
void Close(
    bool sync
)

Parameters

sync
Type: System::Boolean
If false, do not flush cached information to disk.
Remarks

The sync parameter is a dangerous option. It should be set to false only if the application is doing logging (with transactions) so that the database is recoverable after a system or application crash, or if the database is always generated from scratch after any system or application crash.

It is important to understand that flushing cached information to disk only minimizes the window of opportunity for corrupted data. Although unlikely, it is possible for database corruption to happen if a system or application crash occurs while writing data to the database. To ensure that database corruption never occurs, applications must either use transactions and logging with automatic recovery or edit a copy of the database, and once all applications using the database have successfully called Close, atomically replace the original database with the updated copy.

Note that this parameter only works when the database has been opened using an environment.
See Also

BaseDatabase Class
Close Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
BaseDatabase Class  See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Cursor()</code></td>
<td>Create a database cursor.</td>
</tr>
<tr>
<td><code>Cursor(CursorConfig)</code></td>
<td>Create a database cursor with the given configuration.</td>
</tr>
<tr>
<td><code>Cursor(Transaction)</code></td>
<td>Create a transactionally protected database cursor.</td>
</tr>
<tr>
<td><code>Cursor(CursorConfig, Transaction)</code></td>
<td>Create a transactionally protected database cursor with the given configuration.</td>
</tr>
</tbody>
</table>
See Also

BaseDatabase Class
BaseDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Create a database cursor.

**Namespace:** BerkeleyDB
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public Cursor Cursor()

Visual Basic (Declaration)

Public Function Cursor As Cursor

Visual C++

public:
Cursor^ Cursor()

Return Value

A newly created cursor
See Also

BaseDatabase Class
Cursor Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Create a database cursor with the given configuration.

**Namespace:**  BerkeleyDB  
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public Cursor Cursor(
    CursorConfig cfg
)

Visual Basic (Declaration)

Public Function Cursor ( _
    cfg As CursorConfig _
) As Cursor

Visual C++

public:
Cursor^ Cursor(
    CursorConfig^ cfg
)

Parameters

cfg
    Type: BerkeleyDB::<CursorConfig
    The configuration properties for the cursor.

Return Value

A newly created cursor
See Also

BaseDatabase Class
Cursor Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Create a transactionally protected database cursor.

**Namespace:**  BerkeleyDB
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
**Syntax**

**C#**

```csharp
public Cursor Cursor(
    Transaction txn
)
```

**Visual Basic (Declaration)**

```vbnet
Public Function Cursor ( _
   txn As Transaction _
) As Cursor
```

**Visual C++**

```cpp
public:
    Cursor^ Cursor(
        Transaction^ txn
    )
```

**Parameters**

`txn`

Type: **BerkeleyDB::Transaction**

The transaction context in which the cursor may be used.

**Return Value**

A newly created cursor
See Also

BaseDatabase Class
Cursor Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Create a transactionally protected database cursor with the given configuration.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

```csharp
public Cursor Cursor(
    CursorConfig cfg,
    Transaction txn
)
```

Visual Basic (Declaration)

```vbnet
Public Function Cursor ( _
    cfg As CursorConfig, _
    txn As Transaction _
) As Cursor
```

Visual C++

```cpp
public: 
Cursor^ Cursor(
    CursorConfig^ cfg,
    Transaction^ txn
)
```

Parameters

cfg
Type: BerkeleyDB::::CursorConfig
The configuration properties for the cursor.

txn
Type: BerkeleyDB::::Transaction
The transaction context in which the cursor may be used.

Return Value

A newly created cursor
See Also

BaseDatabase Class
Cursor Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation

BaseDatabase...::Delete Method

BaseDatabase Class   See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delete(DatabaseEntry)</td>
<td>Remove key/data pairs from the database. The key/data pair associated with key is discarded from the database. In the presence of duplicate key values, all records associated with the designated key will be discarded.</td>
</tr>
<tr>
<td>Delete(DatabaseEntry, Transaction)</td>
<td>Remove key/data pairs from the database. The key/data pair associated with key is discarded from the database. In the presence of duplicate key values, all records associated with the designated key will be discarded.</td>
</tr>
</tbody>
</table>
See Also

BaseDatabase Class
BaseDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Remove key/data pairs from the database. The key/data pair associated with key is discarded from the database. In the presence of duplicate key values, all records associated with the designated key will be discarded.

**Namespace:** BerkeleyDB
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

```csharp
public void Delete(
    DatabaseEntry key
)
```

Visual Basic (Declaration)

```vbnet
Public Sub Delete (_
    key As DatabaseEntry _
)
```

Visual C++

```cpp
public:
void Delete(
    DatabaseEntry^ key
)
```

Parameters

key

Type: BerkeleyDB::DatabaseEntry
Discard the key/data pair associated with key.
Remarks

When called on a secondary database, remove the key/data pair from the primary database and all secondary indices.

If the operation occurs in a transactional database, the operation will be implicitly transaction protected.
## Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>BerkeleyDB::: NotFoundException</td>
<td>A NotFoundException is thrown if key is not in the database.</td>
</tr>
<tr>
<td></td>
<td>A KeyEmptyException is thrown if the database is a QueueDatabase or</td>
</tr>
<tr>
<td></td>
<td>RecnoDatabase database and key exists, but was never explicitly created by</td>
</tr>
<tr>
<td></td>
<td>the application or was later deleted.</td>
</tr>
</tbody>
</table>
See Also

BaseDatabase Class
Delete Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Remove key/data pairs from the database. The key/data pair associated with key is discarded from the database. In the presence of duplicate key values, all records associated with the designated key will be discarded.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public void Delete(
    DatabaseEntry key,
    Transaction txn
)

Visual Basic (Declaration)

Public Sub Delete ( _
    key As DatabaseEntry, _
    txn As Transaction _
)

Visual C++

public:
void Delete(
    DatabaseEntry^ key,
    Transaction^ txn
)

Parameters

key
Type: BerkeleyDB::DatabaseEntry
Discard the key/data pair associated with key.

txn
Type: BerkeleyDB::Transaction
If the operation is part of an application-specified transaction, txn is a Transaction object returned from BeginTransaction(); if the operation is part of a Berkeley DB Concurrent Data Store group, txn is a handle returned from BeginCDSGroup(); otherwise null.
Remarks

When called on a secondary database, remove the key/data pair from the primary database and all secondary indices.

If txn is null and the operation occurs in a transactional database, the operation will be implicitly transaction protected.
## Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>BerkeleyDB::::NotFoundException</td>
<td>A NotFoundException is thrown if key is not in the database.</td>
</tr>
<tr>
<td></td>
<td>A KeyEmptyException is thrown if the database is a QueueDatabase or</td>
</tr>
<tr>
<td></td>
<td>RecnoDatabase database and key exists, but was never explicitly created</td>
</tr>
<tr>
<td></td>
<td>by the application or was later deleted.</td>
</tr>
</tbody>
</table>
See Also

BaseDatabase Class
Delete Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Release the resources held by this object, and close the database if it's still open.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public void Dispose()

Visual Basic (Declaration)

Public Sub Dispose

Visual C++

public:
virtual void Dispose() sealed

Implements

IDisposable:::Dispose()()
See Also

BaseDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
BaseDatabase Class

See Also
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exists(DatabaseEntry)</td>
<td>Check whether key appears in the database.</td>
</tr>
<tr>
<td>Exists(DatabaseEntry, Transaction)</td>
<td>Check whether key appears in the database.</td>
</tr>
<tr>
<td>Exists(DatabaseEntry, Transaction, LockingInfo)</td>
<td>Check whether key appears in the database.</td>
</tr>
</tbody>
</table>
See Also

BaseDatabase Class
BaseDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Check whether key appears in the database.

**Namespace:**  [BerkeleyDB](https://www.berkeleydb.com)

**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll)  Version: 4.8.24.0
### Syntax

#### C#

```csharp
public bool Exists(
    DatabaseEntry key
)
```

#### Visual Basic (Declaration)

```vbnet
Public Function Exists (_
    key As DatabaseEntry_
) As Boolean
```

#### Visual C++

```cpp
public:
    bool Exists(
        DatabaseEntry^ key
    )
```

### Parameters

**key**

Type: `BerkeleyDB::::DatabaseEntry`

The key to search for.

### Return Value

True if key appears in the database, false otherwise.
Remarks

If the operation occurs in a transactional database, the operation will be implicitly transaction protected.
## Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>BerkeleyDB::::NotFoundException</code></td>
<td>A <code>NotFoundException</code> is thrown if key is not in the database.</td>
</tr>
<tr>
<td><code>BerkeleyDB::::KeyEmptyException</code></td>
<td>A <code>KeyEmptyException</code> is thrown if the database is a <code>QueueDatabase</code> or <code>RecnoDatabase</code> database and key exists, but was never explicitly created by the application or was later deleted.</td>
</tr>
</tbody>
</table>
See Also

BaseDatabase Class
Exists Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Check whether key appears in the database.

**Namespace:**  [BerkeleyDB](#)  
**Assembly:**  [libdb_dotnet48](#) (in libdb_dotnet48.dll)  
**Version:**  4.8.24.0
**Syntax**

**C#**

```csharp
public bool Exists(
    DatabaseEntry key,
    Transaction txn
)
```

**Visual Basic (Declaration)**

```vbnet
Public Function Exists ( _
    key As DatabaseEntry, _
    txn As Transaction _
) As Boolean
```

**Visual C++**

```cpp
public:
bool Exists(  
    DatabaseEntry^ key,
    Transaction^ txn
)
```

**Parameters**

**key**

Type: `BerkeleyDB::::DatabaseEntry`
The key to search for.

**txn**

Type: `BerkeleyDB::::Transaction`
If the operation is part of an application-specified transaction, txn is a Transaction object returned from `BeginTransaction()()`; if the operation is part of a Berkeley DB Concurrent Data Store group, txn is a handle returned from `BeginCDSGroup()()`; otherwise null.

**Return Value**
True if key appears in the database, false otherwise.
Remarks

If txn is null and the operation occurs in a transactional database, the operation will be implicitly transaction protected.
## Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>BerkeleyDB::NotFoundException</code></td>
<td>A <code>NotFoundException</code> is thrown if key is not in the database.</td>
</tr>
<tr>
<td></td>
<td>A <code>KeyEmptyException</code> is thrown if the database is a <code>QueueDatabase</code> or <code>RecnoDatabase</code> database and key exists, but was never explicitly created by the application or was later deleted.</td>
</tr>
</tbody>
</table>
See Also

BaseDatabase Class
Exists Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Check whether key appears in the database.

**Namespace:** [BerkeleyDB](https://www.berkeleydb.com)

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool Exists(
    DatabaseEntry key,
    Transaction txn,
    LockingInfo info
)

Visual Basic (Declaration)

Public Function Exists ( _
    key As DatabaseEntry, _
    txn As Transaction, _
    info As LockingInfo _
) As Boolean

Visual C++

public:
bool Exists(
    DatabaseEntry^ key,
    Transaction^ txn,
    LockingInfo^ info
)

Parameters

key
Type: BerkeleyDB::::DatabaseEntry
The key to search for.

txn
Type: BerkeleyDB::::Transaction
If the operation is part of an application-specified transaction, txn is a Transaction object returned from BeginTransaction(); if the operation is part of a Berkeley DB Concurrent Data Store group, txn is a handle returned from BeginCDSGroup(); otherwise null.
info
Type: BerkeleyDB::LockingInfo
The locking behavior to use.

Return Value

True if key appears in the database, false otherwise.
Remarks

If txn is null and the operation occurs in a transactional database, the operation will be implicitly transaction protected.
## Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>BerkeleyDB::::NotFoundException</td>
<td>A Not Found Exception is thrown if key is not in the database.</td>
</tr>
<tr>
<td></td>
<td>A Key Empty Exception is thrown if the database is a QueueDatabase or RecnoDatabase database and key exists, but was never explicitly created by the application or was later deleted.</td>
</tr>
</tbody>
</table>
See Also

BaseDatabase Class
Exists Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
BaseDatabase Class  See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Get(DatabaseEntry)</td>
<td>Retrieve a key/data pair from the database. In the presence of duplicate key values, Get will return the first data item for key.</td>
</tr>
<tr>
<td>Get(DatabaseEntry, Transaction)</td>
<td>Retrieve a key/data pair from the database. In the presence of duplicate key values, Get will return the first data item for key.</td>
</tr>
<tr>
<td>Get(DatabaseEntry, Transaction, LockingInfo)</td>
<td>Retrieve a key/data pair from the database. In the presence of duplicate key values, Get will return the first data item for key.</td>
</tr>
</tbody>
</table>
See Also

BaseDatabase Class
BaseDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
BaseDatabase.....::Get Method (DatabaseEntry)

BaseDatabase Class  See Also

Retrieve a key/data pair from the database. In the presence of duplicate key values, Get will return the first data item for key.

Namespace:  BerkeleyDB
Syntax

C#

public KeyValuePair<DatabaseEntry, DatabaseEntry> Get(
    DatabaseEntry key
)

Visual Basic (Declaration)

Public Function Get ( _
    key As DatabaseEntry _
) As KeyValuePair(Of DatabaseEntry, DatabaseEntry)

Visual C++

public:
    KeyValuePair<DatabaseEntry^, DatabaseEntry^> Get(
    DatabaseEntry^ key
)

Parameters

key
    Type: BerkeleyDB::DatabaseEntry
    The key to search for

Return Value

A KeyValuePair(Of (TKey, TValue)>) whose Key parameter is key and whose Value parameter is the retrieved data.
Remarks

If the operation occurs in a transactional database, the operation will be implicitly transaction protected.
### Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>BerkeleyDB::::NotFoundException</code></td>
<td>A <code>NotFoundException</code> is thrown if key is not in the database.</td>
</tr>
<tr>
<td><code>BerkeleyDB::::KeyEmptyException</code></td>
<td>A <code>KeyEmptyException</code> is thrown if the database is a <code>QueueDatabase</code> or <code>RecnoDatabase</code> database and key exists, but was never explicitly created by the application or was later deleted.</td>
</tr>
</tbody>
</table>
See Also

BaseDatabase Class
Get Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Retrieves a key/data pair from the database. In the presence of duplicate key values, Get will return the first data item for key.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public KeyValuePair<DatabaseEntry, DatabaseEntry> Get(
    DatabaseEntry key,
    Transaction txn
)

Visual Basic (Declaration)

Public Function Get (_
    key As DatabaseEntry, _
    txn As Transaction _
) As KeyValuePair(Of DatabaseEntry, DatabaseEntry)

Visual C++

public:
    KeyValuePair<DatabaseEntry^, DatabaseEntry^> Get(
        DatabaseEntry^ key,
        Transaction^ txn
    )

Parameters

key

Type: BerkeleyDB::DatabaseEntry
The key to search for

txn

Type: BerkeleyDB::Transaction
txn is a Transaction object returned from BeginTransaction(); if the operation is part of a Berkeley DB Concurrent Data Store group, txn is a handle returned from BeginCDSGroup(); otherwise null.

Return Value
A `KeyValuePair<(Of <(TKey, TValue)>)>` whose Key parameter is key and whose Value parameter is the retrieved data.
Remarks

If txn is null and the operation occurs in a transactional database, the operation will be implicitly transaction protected.
## Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BerkeleyDB::::NotFoundException</strong></td>
<td>A <code>NotFoundException</code> is thrown if key is not in the database.</td>
</tr>
<tr>
<td><strong>BerkeleyDB::::KeyEmptyException</strong></td>
<td>A <code>KeyEmptyException</code> is thrown if the database is a <a href="#">QueueDatabase</a> or <a href="#">RecnoDatabase</a> database and key exists, but was never explicitly created by the application or was later deleted.</td>
</tr>
</tbody>
</table>
See Also

BaseDatabase Class
Get Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Retrieve a key/data pair from the database. In the presence of duplicate key values, Get will return the first data item for key.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public KeyValuePair<DatabaseEntry, DatabaseEntry> Get(
    DatabaseEntry key,
    Transaction txn,
    LockingInfo info
)

Visual Basic (Declaration)

Public Function Get ( _
    key As DatabaseEntry, _
    txn As Transaction, _
    info As LockingInfo _
) As KeyValuePair(Of DatabaseEntry, DatabaseEntry)

Visual C++

public:
    KeyValuePair<DatabaseEntry^, DatabaseEntry^> Get(
        DatabaseEntry^ key,
        Transaction^ txn,
        LockingInfo^ info
    )

Parameters

key
Type: BerkeleyDB::DatabaseEntry
The key to search for

txn
Type: BerkeleyDB::Transaction
txn is a Transaction object returned from BeginTransaction(); if the operation is part of a Berkeley DB Concurrent Data Store group, txn is a handle returned from BeginCDSGroup(); otherwise null.
info
Type: BerkeleyDB:::LockingInfo
The locking behavior to use.

Return Value

A KeyValuePair<(Of <(TKey, TValue)>)> whose Key parameter is key and whose Value parameter is the retrieved data.
Remarks

If txn is null and the operation occurs in a transactional database, the operation will be implicitly transaction protected.
## Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>BerkeleyDB::::NotFoundException</td>
<td>A NotFoundException is thrown if key is not in the database.</td>
</tr>
<tr>
<td>BerkeleyDB::::KeyEmptyException</td>
<td>A KeyEmptyException is thrown if the database is a QueueDatabase or RecnoDatabase database and key exists, but was never explicitly created by the application or was later deleted.</td>
</tr>
</tbody>
</table>
See Also

BaseDatabase Class
Get Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation

BaseDatabase.GetBoth Method

BaseDatabase Class  See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>GetBoth(DatabaseEntry, DatabaseEntry)</strong></td>
<td>Retrieve a key/data pair from the database which matches key and data.</td>
</tr>
<tr>
<td><strong>GetBoth(DatabaseEntry, DatabaseEntry, Transaction)</strong></td>
<td>Retrieve a key/data pair from the database which matches key and data.</td>
</tr>
<tr>
<td><strong>GetBoth(DatabaseEntry, DatabaseEntry, Transaction, LockingInfo)</strong></td>
<td>Retrieve a key/data pair from the database which matches key and data.</td>
</tr>
</tbody>
</table>
See Also

BaseDatabase Class
BaseDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Retrieve a key/data pair from the database which matches key and data.

**Namespace:** [BerkeleyDB](https://www.berkeleydb.com/)

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
**Syntax**

**C#**

```csharp
public KeyValuePair<DatabaseEntry, DatabaseEntry> GetBoth(
    DatabaseEntry key,
    DatabaseEntry data
)
```

**Visual Basic (Declaration)**

```vbnet
Public Function GetBoth ( _
    key As DatabaseEntry, _
    data As DatabaseEntry _
) As KeyValuePair(Of DatabaseEntry, DatabaseEntry)
```

**Visual C++**

```cpp
public:
    KeyValuePair<DatabaseEntry^, DatabaseEntry^> GetBoth(
        DatabaseEntry^ key,
        DatabaseEntry^ data
    )
```

**Parameters**

key
   Type: BerkeleyDB::DatabaseEntry
   The key to search for

data
   Type: BerkeleyDB::DatabaseEntry
   The data to search for

**Return Value**

A KeyValuePair(Of TKey, TValue) whose Key parameter is key and whose Value parameter is data.
Remarks

If the operation occurs in a transactional database, the operation will be implicitly transaction protected.
## Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="#">BerkeleyDB::NotNotFoundException</a></td>
<td>A <code>NotFoundException</code> is thrown if key and data are not in the database.</td>
</tr>
<tr>
<td><a href="#">BerkeleyDB::KeyEmptyException</a></td>
<td>A <code>KeyEmptyException</code> is thrown if the database is a <a href="#">QueueDatabase</a> or <a href="#">RecnoDatabase</a> database and key exists, but was never explicitly created by the application or was later deleted.</td>
</tr>
</tbody>
</table>
See Also

BaseDatabase Class
GetBoth Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Retrieves a key/data pair from the database which matches key and data.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
**Syntax**

**C#**

```csharp
public KeyValuePair<DatabaseEntry, DatabaseEntry> GetBoth(
    DatabaseEntry key,
    DatabaseEntry data,
    Transaction txn
)
```

**Visual Basic (Declaration)**

```vbnet
Public Function GetBoth ( _
    key As DatabaseEntry, _
    data As DatabaseEntry, _
    txn As Transaction _
) As KeyValuePair(Of DatabaseEntry, DatabaseEntry)
```

**Visual C++**

```cpp
public:
    KeyValuePair<DatabaseEntry^, DatabaseEntry^> GetBoth(
        DatabaseEntry^ key,
        DatabaseEntry^ data,
        Transaction^ txn
    )
```

**Parameters**

**key**

Type: `BerkeleyDB::::DatabaseEntry`  
The key to search for

**data**

Type: `BerkeleyDB::::DatabaseEntry`  
The data to search for

**txn**

Type: `BerkeleyDB::::Transaction`
txn is a Transaction object returned from `BeginTransaction()`; if the operation is part of a Berkeley DB Concurrent Data Store group, txn is a handle returned from `BeginCDSGroup()`; otherwise null.

**Return Value**

A `KeyValuePair<Of <(TKey, TValue)>)>` whose Key parameter is key and whose Value parameter is data.
Remarks

If txn is null and the operation occurs in a transactional database, the operation will be implicitly transaction protected.
## Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>BerkeleyDB:::NotFoundException</td>
<td>A <code>NotFoundException</code> is thrown if key and data are not in the database.</td>
</tr>
<tr>
<td></td>
<td>A KeyEmptyException is thrown if the database is a <code>QueueDatabase</code> or</td>
</tr>
<tr>
<td></td>
<td><code>RecnoDatabase</code> database and key exists, but was never explicitly created</td>
</tr>
<tr>
<td></td>
<td>by the application or was later deleted.</td>
</tr>
</tbody>
</table>
See Also

BaseDatabase Class
GetBoth Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Retrieve a key/data pair from the database which matches key and data.

Namespace: BerkeleyDB
Syntax

C#

public KeyValuePair<DatabaseEntry, DatabaseEntry> GetBoth(
    DatabaseEntry key,
    DatabaseEntry data,
    Transaction txn,
    LockingInfo info
)

Visual Basic (Declaration)

Public Function GetBoth ( _
    key As DatabaseEntry, _
    data As DatabaseEntry, _
    txn As Transaction, _
    info As LockingInfo _
) As KeyValuePair(Of DatabaseEntry, DatabaseEntry)

Visual C++

public:
    KeyValuePair<DatabaseEntry^, DatabaseEntry^> GetBoth(
        DatabaseEntry^ key,
        DatabaseEntry^ data,
        Transaction^ txn,
        LockingInfo^ info
    )

Parameters

key
Type: BerkeleyDB...:DatabaseEntry
The key to search for

data
Type: BerkeleyDB...:DatabaseEntry
The data to search for
txn
Type: `BerkeleyDB::::Transaction`
txn is a Transaction object returned from `BeginTransaction()`; if the operation is part of a Berkeley DB Concurrent Data Store group, txn is a handle returned from `BeginCDSGroup()`; otherwise null.

info
Type: `BerkeleyDB::::LockingInfo`
The locking behavior to use.

**Return Value**

A `KeyValuePair<(Of <(TKey, TValue)>)>` whose Key parameter is key and whose Value parameter is data.
Remarks

If txn is null and the operation occurs in a transactional database, the operation will be implicitly transaction protected.
# Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>BerkeleyDB:::NotFoundException</td>
<td>A <code>NotFoundException</code> is thrown if key and data are not in the database. A <code>KeyEmptyException</code> is thrown if the database is a <code>QueueDatabase</code> or <code>RecnoDatabase</code> database and key exists, but was never explicitly created by the application or was later deleted.</td>
</tr>
</tbody>
</table>
See Also

BaseDatabase Class
GetBoth Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The statistical information is described by the BTreeStats, HashStats, QueueStats, and RecnoStats classes.
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>PrintFastStats()</code></td>
<td>Display the database statistical information which does not require traversal of the database.</td>
</tr>
<tr>
<td><code>PrintFastStats(Boolean)</code></td>
<td>Display the database statistical information which does not require traversal of the database.</td>
</tr>
</tbody>
</table>
See Also

BaseDatabase Class
BaseDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Display the database statistical information which does not require traversal of the database.

Namespace: BerkeleyDB
Syntax

C#

public void PrintFastStats()

Visual Basic (Declaration)

Public Sub PrintFastStats

Visual C++

public:
void PrintFastStats()
Remarks

Among other things, this method makes it possible for applications to request key and record counts without incurring the performance penalty of traversing the entire database.
See Also

BaseDatabase Class
PrintFastStats Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Display the database statistical information which does not require traversal of the database.

Namespace: BerkeleyDB
Syntax

C#

public void PrintFastStats(
    bool PrintAll
)

Visual Basic (Declaration)

Public Sub PrintFastStats ( _
    PrintAll As Boolean _
)

Visual C++

public:
void PrintFastStats(
    bool PrintAll
)

Parameters

PrintAll
Type: System::Boolean
If true, display all available information.
Remarks

Among other things, this method makes it possible for applications to request key and record counts without incurring the performance penalty of traversing the entire database.
See Also

BaseDatabase Class
PrintFastStats Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The statistical information is described by the **BTreeStats**, **HashStats**, **QueueStats**, and **RecnoStats** classes.
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PrintStats()</td>
<td>Display the database statistical information.</td>
</tr>
<tr>
<td>PrintStats(Boolean)</td>
<td>Display the database statistical information.</td>
</tr>
</tbody>
</table>
See Also

BaseDatabase Class
BaseDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Display the database statistical information.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public void PrintStats()

Visual Basic (Declaration)

Public Sub PrintStats

Visual C++

public:
void PrintStats()
See Also

BaseDatabase Class
PrintStats Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Display the database statistical information.

**Namespace:**  [BerkeleyDB](https://www.berkeleydb.com)

**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public void PrintStats(
    bool PrintAll
)

Visual Basic (Declaration)

Public Sub PrintStats ( _
    PrintAll As Boolean _
)

Visual C++

public:
void PrintStats(
    bool PrintAll
)

Parameters

PrintAll
    Type: System::::Boolean
    If true, display all available information.
See Also

BaseDatabase Class
PrintStats Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Applications should never remove databases with open DB handles, or in the case of removing a file, when any database in the file has an open handle. For example, some architectures do not permit the removal of files with open system handles. On these architectures, attempts to remove databases currently in use by any thread of control in the system may fail.

Remove should not be called if the remove is intended to be transactionally safe; RemoveDB(String, Boolean) should be used instead.
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>s Remove(String)</code></td>
<td>Remove the underlying file represented by Filename, incidentally removing all of the databases it contained.</td>
</tr>
<tr>
<td><code>s Remove(String, DatabaseEnvironment)</code></td>
<td>Remove the underlying file represented by Filename, incidentally removing all of the databases it contained.</td>
</tr>
<tr>
<td><code>s Remove(String, String)</code></td>
<td>Remove the database specified by Filename and DatabaseName.</td>
</tr>
<tr>
<td><code>s Remove(String, String, DatabaseEnvironment)</code></td>
<td>Remove the database specified by Filename and DatabaseName.</td>
</tr>
</tbody>
</table>
See Also

BaseDatabase Class
BaseDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Remove the underlying file represented by Filename, incidentally removing all of the databases it contained.

**Namespace:** [BerkeleyDB](http://www.berkeleydb.com)

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public static void Remove(
    string Filename
)

Visual Basic (Declaration)

Public Shared Sub Remove ( _
    Filename As String _
)

Visual C++

public:
static void Remove(
    String^ Filename
)

Parameters

Filename
    Type: System::String
    The file to remove
See Also

BaseDatabase Class
Remove Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Remove Method (String, DatabaseEnvironment)

Remove the underlying file represented by Filename, incidentally removing all of the databases it contained.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public static void Remove(
    string Filename,
    DatabaseEnvironment DbEnv
)

Visual Basic (Declaration)

Public Shared Sub Remove ( _
    Filename As String, _
    DbEnv As DatabaseEnvironment _
)

Visual C++

public:
    static void Remove(
        String^ Filename,
        DatabaseEnvironment^ DbEnv
    )

Parameters

Filename
Type: System::String
The file to remove

DbEnv
Type: BerkeleyDB::DatabaseEnvironment
The DatabaseEnvironment the database belongs to
See Also

BaseDatabase Class
Remove Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Remove Method (String, String)

Remove the database specified by Filename and DatabaseName.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
### Syntax

**C#**

```csharp
public static void Remove(
    string Filename,
    string DatabaseName
)
```

**Visual Basic (Declaration)**

```vbnet
Public Shared Sub Remove (_
    Filename As String,
    DatabaseName As String _
)
```

**Visual C++**

```cpp
public:
static void Remove(
    String^ Filename,
    String^ DatabaseName
)
```

### Parameters

**Filename**
- Type: `System::String`
- The file to remove

**DatabaseName**
- Type: `System::String`
- The database to remove
See Also

BaseDatabase Class
Remove Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation

BaseDatabase..::.Remove Method (String, String, DatabaseEnvironment)

**BaseDatabase Class**  [See Also](#)

Remove the database specified by Filename and DatabaseName.

**Namespace:**  [BerkeleyDB](#)

**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public static void Remove(
    string Filename,
    string DatabaseName,
    DatabaseEnvironment DbEnv
)

Visual Basic (Declaration)

Public Shared Sub Remove (_,
    Filename As String, _,
    DatabaseName As String, _,
    DbEnv As DatabaseEnvironment _
)

Visual C++

public:
    static void Remove(
        String^ Filename,
        String^ DatabaseName,
        DatabaseEnvironment^ DbEnv
    )

Parameters

Filename
    Type: System::String
    The file to remove

DatabaseName
    Type: System::String
    The database to remove

DbEnv
    Type: BerkeleyDB::DatabaseEnvironment
The DatabaseEnvironment the database belongs to
See Also

BaseDatabase Class
Remove Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Applications should not rename databases that are currently in use. If an underlying file is being renamed and logging is currently enabled in the database environment, no database in the file may be open when Rename is called. In particular, some architectures do not permit renaming files with open handles. On these architectures, attempts to rename databases that are currently in use by any thread of control in the system may fail.

Rename should not be called if the rename is intended to be transactionally safe; `RenameDB(String, String, Boolean)` should be used instead.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rename(String, String)</td>
<td>Rename the underlying file represented by Filename, incidentally renaming all of the databases it contained.</td>
</tr>
<tr>
<td>Rename(String, String, DatabaseEnvironment)</td>
<td>Rename the underlying file represented by Filename, incidentally renaming all of the databases it contained.</td>
</tr>
<tr>
<td>Rename(String, String, String)</td>
<td>Rename the database specified by Filename and DatabaseName.</td>
</tr>
<tr>
<td>Rename(String, String, DatabaseEnvironment)</td>
<td>Rename the database specified by Filename and DatabaseName.</td>
</tr>
</tbody>
</table>
See Also

BaseDatabase Class
BaseDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
rename method (String, String)

namespace: BerkeleyDB
assembly: libdb_dotnet48 (in libdb_dotnet48.dll) version: 4.8.24.0
Syntax

C#

public static void Rename(
    string Filename,
    string NewName
)

Visual Basic (Declaration)

Public Shared Sub Rename (_
    Filename As String, _
    NewName As String _
)

Visual C++

public:
static void Rename(
    String^ Filename,
    String^ NewName
)

Parameters

Filename
    Type: System::String
    The file to rename

NewName
    Type: System::String
    The new filename
See Also

BaseDatabase Class
Rename Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
BaseDatabase Class See Also

Rename the underlying file represented by Filename, incidentally renaming all of the databases it contained.

Namespace: BerkeleyDB
Syntax

C#

```csharp
public static void Rename(
    string Filename,
    string NewName,
    DatabaseEnvironment DbEnv
)
```

Visual Basic (Declaration)

```vbnet
Public Shared Sub Rename (_
    Filename As String, _
    NewName As String, _
    DbEnv As DatabaseEnvironment _
)
```

Visual C++

```cpp
public:
static void Rename(
    String^ Filename,
    String^ NewName,
    DatabaseEnvironment^ DbEnv
)
```

Parameters

Filename
Type: `System::String`
The file to rename

NewName
Type: `System::String`
The new filename

DbEnv
Type: `BerkeleyDB::DatabaseEnvironment`
The DatabaseEnvironment the database belongs to
See Also

BaseDatabase Class
Rename Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
BaseDatabase....:::Rename Method (String, String, String)

BaseDatabase Class  See Also

Rename the database specified by Filename and DatabaseName.

Namespace:  BerkeleyDB
Syntax

C#

public static void Rename(
    string Filename,
    string DatabaseName,
    string NewName
)

Visual Basic (Declaration)

Public Shared Sub Rename (
    FileName As String, 
    DatabaseName As String, 
    NewName As String
)

Visual C++

public:
static void Rename(
    String^ Filename,
    String^ DatabaseName,
    String^ NewName
)

Parameters

Filename
   Type: System::String
   The file to rename

DatabaseName
   Type: System::String
   The database to rename

NewName
   Type: System::String
The new database name
See Also

BaseDatabase Class
Rename Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
BaseDatabase.Rename Method (String, String, String, DatabaseEnvironment)

See Also

Rename the database specified by Filename and DatabaseName.

Namespace: BerkeleyDB
### Syntax

**C#**

```csharp
public static void Rename(
    string Filename,
    string DatabaseName,
    string NewName,
    DatabaseEnvironment DbEnv
)
```

**Visual Basic (Declaration)**

```vbnet
Public Shared Sub Rename (_
    Filename As String, _
    DatabaseName As String, _
    NewName As String, _
    DbEnv As DatabaseEnvironment _
)
```

**Visual C++**

```cpp
public:
static void Rename(
    String^ Filename,
    String^ DatabaseName,
    String^ NewName,
    DatabaseEnvironment^ DbEnv
)
```

### Parameters

**Filename**
- **Type:** `System::::String`
- The file to rename

**DatabaseName**
- **Type:** `System::::String`
- The database to rename
NewName
Type: System::String
The new database name

DbEnv
Type: BerkeleyDB::DatabaseEnvironment
The DatabaseEnvironment the database belongs to
See Also

BaseDatabase Class
Rename Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Flush any cached information to disk.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#
public void Sync()

Visual Basic (Declaration)
Public Sub Sync

Visual C++
public:
void Sync()
Remarks

If the database is in memory only, Sync has no effect and will always succeed.

It is important to understand that flushing cached information to disk only minimizes the window of opportunity for corrupted data. Although unlikely, it is possible for database corruption to happen if a system or application crash occurs while writing data to the database. To ensure that database corruption never occurs, applications must either: use transactions and logging with automatic recovery or edit a copy of the database, and once all applications using the database have successfully called `Close()` atomically replace the original database with the updated copy.
See Also

BaseDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
When called on a database configured with secondary indices, Truncate will truncate the primary database and all secondary indices. A count of the records discarded from the primary database is returned.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Truncate()()()</td>
<td>Empty the database, discarding all records it contains.</td>
</tr>
<tr>
<td>Truncate(Transaction)</td>
<td>Empty the database, discarding all records it contains.</td>
</tr>
</tbody>
</table>
See Also

BaseDatabase Class
BaseDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Empty the database, discarding all records it contains.

**Namespace:** BerkeleyDB
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public uint Truncate()

Visual Basic (Declaration)

Public Function Truncate As UInteger

Visual C++

public:
unsigned int Truncate()

Return Value

The number of records discarded from the database.
Remarks

If the operation occurs in a transactional database, the operation will be implicitly transaction protected.
See Also

BaseDatabase Class
Truncate Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Empty the database, discarding all records it contains.

Namespace: BerkeleyDB
Syntax

**C#**

```csharp
public uint Truncate(
    Transaction txn
)
```

**Visual Basic (Declaration)**

```vbnet
Public Function Truncate(_
    As Transaction _
) As UInteger
```

**Visual C++**

```cpp
public:
    unsigned int Truncate(_
        Transaction^ txn
    )
```

Parameters

txn

Type: BerkeleyDB::::Transaction

txn is a Transaction object returned from `BeginTransaction()()`; if the operation is part of a Berkeley DB Concurrent Data Store group, txn is a handle returned from `BeginCDSGroup()()`; otherwise null.

Return Value

The number of records discarded from the database.
Remarks

If txn is null and the operation occurs in a transactional database, the operation will be implicitly transaction protected.
See Also

BaseDatabase Class
Truncate Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
BaseDatabase Properties

The **BaseDatabase** type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AutoCommit</strong></td>
<td>If true, all database modification operations based on this object will be transactionally protected.</td>
</tr>
<tr>
<td><strong>CacheSize</strong></td>
<td>The size of the shared memory buffer pool -- that is, the cache.</td>
</tr>
<tr>
<td><strong>Creation</strong></td>
<td>The CreatePolicy with which this database was opened.</td>
</tr>
<tr>
<td><strong>DatabaseName</strong></td>
<td>The name of this database, if it has one.</td>
</tr>
<tr>
<td><strong>DoChecksum</strong></td>
<td>If true, do checksum verification of pages read into the cache from the backing filestore.</td>
</tr>
<tr>
<td><strong>EncryptAlgorithm</strong></td>
<td>The algorithm used by the Berkeley DB library to perform encryption and decryption.</td>
</tr>
<tr>
<td><strong>Encrypted</strong></td>
<td>If true, encrypt all data stored in the database.</td>
</tr>
<tr>
<td><strong>Endianness</strong></td>
<td>The database byte order.</td>
</tr>
<tr>
<td><strong>ErrorFeedback</strong></td>
<td>The mechanism for reporting detailed error messages to the application.</td>
</tr>
<tr>
<td><strong>ErrorPrefix</strong></td>
<td>The prefix string that appears before error messages issued by Berkeley DB.</td>
</tr>
<tr>
<td><strong>Feedback</strong></td>
<td>Monitor progress within long running operations.</td>
</tr>
<tr>
<td><strong>FileName</strong></td>
<td>The filename of this database, if it has one.</td>
</tr>
<tr>
<td><strong>FreeThreaded</strong></td>
<td>If true, the object is free-threaded; that is, concurrently usable by multiple threads in the address space.</td>
</tr>
<tr>
<td><strong>HasMultiple</strong></td>
<td>If true, the object references a physical file supporting multiple databases.</td>
</tr>
<tr>
<td><strong>InHostOrder</strong></td>
<td>If true, the underlying database files were created on an architecture of the same byte order as the current one. This information may be used to determine whether application data needs to be adjusted for this architecture or not.</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-------------</td>
</tr>
<tr>
<td><strong>NoMMap</strong></td>
<td>If true, this database is not mapped into process memory. See <strong>MMapSize</strong> for further information.</td>
</tr>
<tr>
<td><strong>NonDurableTxns</strong></td>
<td>If true, Berkeley DB will not write log records for this database.</td>
</tr>
<tr>
<td><strong>Pagesize</strong></td>
<td>The database's current page size.</td>
</tr>
<tr>
<td><strong>Priority</strong></td>
<td>The cache priority for pages referenced by this object. If true, this database has been opened for reading only. Any attempt to modify items in the database will fail, regardless of the actual permissions of any underlying files.</td>
</tr>
<tr>
<td><strong>ReadOnly</strong></td>
<td>If true, this database supports transactional read operations with degree 1 isolation. Read operations on the database may request the return of modified but not yet committed data.</td>
</tr>
<tr>
<td><strong>ReadUncommitted</strong></td>
<td>If true, this database has been opened in a transactional mode.</td>
</tr>
<tr>
<td><strong>Transactional</strong></td>
<td>If true, the underlying file was physically truncated upon open, discarding all previous databases it might have held.</td>
</tr>
<tr>
<td><strong>Truncated</strong></td>
<td>The type of the underlying access method (and file format). This value may be used to determine the type of the database after an <strong>Open(String, DatabaseConfig)</strong>.</td>
</tr>
<tr>
<td><strong>Type</strong></td>
<td>If true, the database was opened with support for multiversion concurrency control.</td>
</tr>
</tbody>
</table>
See Also

BaseDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, all database modification operations based on this object will be transactionally protected.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool AutoCommit { get; }

Visual Basic (Declaration)

Public ReadOnly Property AutoCommit As Boolean

Visual C++

public:
property bool AutoCommit {
    bool get ();
}

See Also

BaseDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The size of the shared memory buffer pool -- that is, the cache.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

**C#**

public CacheInfo CacheSize { get; }

**Visual Basic (Declaration)**

Public ReadOnly Property CacheSize As CacheInfo

**Visual C++**

public:
property CacheInfo^ CacheSize {
    CacheInfo^ get ();
}
See Also

BaseDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The CreatePolicy with which this database was opened.

Namespace: BerkeleyDB
Syntax

C#

public CreatePolicy Creation { get; }

Visual Basic (Declaration)

Public ReadOnly Property Creation As CreatePolicy

Visual C++

public:
property CreatePolicy Creation {
>CreatePolicy get();
}
See Also

BaseDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The name of this database, if it has one.

**Namespace:**  BerkeleyDB  
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public string DatabaseName { get; }

Visual Basic (Declaration)

Public ReadOnly Property DatabaseName As String

Visual C++

public:
property String^ DatabaseName {
    String^ get ();
}

}
See Also

BaseDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, do checksum verification of pages read into the cache from the backing filestore.

**Namespace:** [BerkeleyDB](#)  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool DoChecksum { get; }

Visual Basic (Declaration)

Public ReadOnly Property DoChecksum As Boolean

Visual C++

public:
property bool DoChecksum {
    bool get ();
}


Remarks

Berkeley DB uses the SHA1 Secure Hash Algorithm if encryption is configured and a general hash algorithm if it is not.
See Also

BaseDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The algorithm used by the Berkeley DB library to perform encryption and decryption.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

**C#**

public `EncryptionAlgorithm` EncryptAlgorithm { get; }

**Visual Basic (Declaration)**

Public ReadOnly Property EncryptAlgorithm As `EncryptionAlgorithm`

**Visual C++**

public:

property `EncryptionAlgorithm` EncryptAlgorithm {
    `EncryptionAlgorithm` get ();
}

See Also

BaseDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, encrypt all data stored in the database.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
### Syntax

**C#**

```csharp
class YourClass
{
    public bool Encrypted { get; }
}
```

**Visual Basic (Declaration)**

```vbnet
Public ReadOnly Property Encrypted As Boolean
```

**Visual C++**

```cpp
public:
    bool Encrypted { get; }
}
```
See Also

BaseDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The database byte order.

**Namespace:** [BerkeleyDB](https://docs.oracle.com/en/database/oracle/database/11.1.0/append/compliance-bdb.html#GUID-511C9C26-7565-406C-A452-3B33A0D4EF92)

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

```csharp
public ByteOrder Endianness { get; }
```

Visual Basic (Declaration)

```vbnet
Public Readonly Property Endianness As ByteOrder
```

Visual C++

```cpp
public:
property ByteOrder^ Endianness {
    ByteOrder^ get ();
}
```
See Also

BaseDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The mechanism for reporting detailed error messages to the application.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public ErrorFeedbackDelegate ErrorFeedback { get; set; }

Visual Basic (Declaration)

Public Property ErrorFeedback As ErrorFeedbackDelegate

Visual C++

public:
    property ErrorFeedbackDelegate^ ErrorFeedback {
        ErrorFeedbackDelegate^ get ();
        void set (ErrorFeedbackDelegate^ value);
    }
Remarks

When an error occurs in the Berkeley DB library, a DatabaseException, or subclass of DatabaseException, is thrown. In some cases, however, the exception may be insufficient to completely describe the cause of the error, especially during initial application debugging.

In some cases, when an error occurs, Berkeley DB will call the given delegate with additional error information. It is up to the delegate to display the error message in an appropriate manner.

Setting ErrorFeedback to NULL unconfigures the callback interface.

This error-logging enhancement does not slow performance or significantly increase application size, and may be run during normal operation as well as during application debugging.

For databases opened inside of a DatabaseEnvironment, setting ErrorFeedback affects the entire environment and is equivalent to setting DatabaseEnvironment.ErrorFeedback.

For databases not opened in an environment, setting ErrorFeedback configures operations performed using the specified object, not all operations performed on the underlying database.
See Also

BaseDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The prefix string that appears before error messages issued by Berkeley DB.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public string ErrorPrefix { get; set; }

Visual Basic (Declaration)

Public Property ErrorPrefix As String

Visual C++

public:
property String^ ErrorPrefix {
    String^ get ();
    void set (String^ value);
}
Remarks

For databases opened inside of a DatabaseEnvironment, setting ErrorPrefix affects the entire environment and is equivalent to setting ErrorPrefix.

Setting ErrorPrefix configures operations performed using the specified object, not all operations performed on the underlying database.
See Also

- BaseDatabase Class
- BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Monitor progress within long running operations.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public DatabaseFeedbackDelegate Feedback { get; set; }

Visual Basic (Declaration)

Public Property Feedback As DatabaseFeedbackDelegate

Visual C++

public:
property DatabaseFeedbackDelegate^ Feedback {
    DatabaseFeedbackDelegate^ get ();
    void set (DatabaseFeedbackDelegate^ value);
}
Remarks

Some operations performed by the Berkeley DB library can take non-trivial amounts of time. The Feedback delegate can be used by applications to monitor progress within these operations. When an operation is likely to take a long time, Berkeley DB will call the specified delegate with progress information.

It is up to the delegate to display this information in an appropriate manner.
See Also

BaseDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The filename of this database, if it has one.

**Namespace:** BerkeleyDB
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public string FileName { get; }

Visual Basic (Declaration)

PublicReadOnly Property FileName As String

Visual C++

public:
property String^ FileName {
    String^ get();
    }


See Also

BaseDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, the object is free-threaded; that is, concurrently usable by multiple threads in the address space.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

```csharp
public bool FreeThreaded { get; }
```

Visual Basic (Declaration)

```vbnet
Public ReadOnly Property FreeThreaded As Boolean
```

Visual C++

```cpp
public:
property bool FreeThreaded {
    bool get ();
}
```
See Also

BaseDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, the object references a physical file supporting multiple databases.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool HasMultiple { get; }

Visual Basic (Declaration)

Public ReadOnly Property HasMultiple As Boolean

Visual C++

public:
property bool HasMultiple {
    bool get ();
}

Remarks

If true, the object is a handle on a database whose key values are the names of the databases stored in the physical file and whose data values are opaque objects. No keys or data values may be modified or stored using the database handle.
See Also

BaseDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, the underlying database files were created on an architecture of the same byte order as the current one. This information may be used to determine whether application data needs to be adjusted for this architecture or not.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool InHostOrder { get; }

Visual Basic (Declaration)

Public ReadOnly Property InHostOrder As Boolean

Visual C++

public:
property bool InHostOrder {
    bool get ();
}

See Also

BaseDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, this database is not mapped into process memory.

See [MMapSize](#) for further information.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool NoMMap { get; }

Visual Basic (Declaration)

Public ReadOnly Property NoMMap As Boolean

Visual C++

public:
property bool NoMMap {
        bool get ();
}

See Also

BaseDatabase Class  
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
NonDurableTxns Property

If true, Berkeley DB will not write log records for this database.

**Namespace:** BerkeleyDB
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool NonDurableTxns { get; }

Visual Basic (Declaration)

Public ReadOnly Property NonDurableTxns As Boolean

Visual C++

public:
property bool NonDurableTxns {
        bool get ();
}
See Also

BaseDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The database's current page size.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

```csharp
public uint Pagesize { get; }
```

Visual Basic (Declaration)

```vbnet
Public ReadOnly Property Pagesize As UInteger
```

Visual C++

```cpp
public:
property unsigned int Pagesize {
    unsigned int get ();
}
```
Remarks

If PageSize was not set by your application, then the default pagesize is selected based on the underlying filesystem I/O block size.
See Also

BaseDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The cache priority for pages referenced by this object.

**Namespace:**  BerkeleyDB
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public CachePriority Priority { get; }

Visual Basic (Declaration)

Public ReadOnly Property Priority As CachePriority

Visual C++

public:
property CachePriority^ Priority {
    CachePriority^ get ();
}
See Also

BaseDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
BaseDatabase.ReadOnly Property

If true, this database has been opened for reading only. Any attempt to modify items in the database will fail, regardless of the actual permissions of any underlying files.

Namespace: BerkeleyDB
Syntax

C#

public bool ReadOnly { get; }

Visual Basic (Declaration)

Public ReadOnly Property ReadOnly As Boolean

Visual C++

public:
property bool ReadOnly {
    bool get ();
}

See Also

BaseDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, this database supports transactional read operations with degree 1 isolation. Read operations on the database may request the return of modified but not yet committed data.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool ReadUncommitted { get; }

Visual Basic (Declaration)

Public ReadOnly Property ReadUncommitted As Boolean

Visual C++

public:
property bool ReadUncommitted {
    bool get();
}
See Also

BaseDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, this database has been opened in a transactional mode.

**Namespace:**  BerkeleyDB  
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool Transactional { get; }

Visual Basic (Declaration)

Public ReadOnly Property Transactional As Boolean

Visual C++

public:
property bool Transactional {
    bool get ();
}

See Also

BaseDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, the underlying file was physically truncated upon open, discarding all previous databases it might have held.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool Truncated { get; }

Visual Basic (Declaration)

Public ReadOnly Property Truncated As Boolean

Visual C++

public:
property bool Truncated {
    bool get ();
}

See Also

BaseDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The type of the underlying access method (and file format). This value may be used to determine the type of the database after an `Open(String, DatabaseConfig)`.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public DatabaseType Type { get; }

Visual Basic (Declaration)

Public ReadOnly Property Type As DatabaseType

Visual C++

public:
property DatabaseType^ Type {
    DatabaseType^ get ();
}
See Also

BaseDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, the database was opened with support for multiversion concurrency control.

Namespace: BerkeleyDB
Syntax

C#

public bool UseMVCC { get; }

Visual Basic (Declaration)

Public ReadOnly Property UseMVCC As Boolean

Visual C++

public:
property bool UseMVCC {
    bool get ();
}

}
See Also

BaseDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
A function to store a compressed key/data pair into a supplied buffer.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
### Syntax

**C#**

```csharp
public delegate bool BTreeCompressDelegate(
    DatabaseEntry prevKey,
    DatabaseEntry prevData,
    DatabaseEntry key,
    DatabaseEntry data,
    ref byte[] dest,
    out int size
)
```

**Visual Basic (Declaration)**

```vbnet
Public Delegate Function BTreeCompressDelegate ( _
    prevKey As DatabaseEntry, _
    prevData As DatabaseEntry, _
    key As DatabaseEntry, _
    data As DatabaseEntry, _
    ByRef dest As Byte(), _
    <OutAttribute> ByRef size As Integer _
) As Boolean
```

**Visual C++**

```cpp
public delegate bool BTreeCompressDelegate(
    DatabaseEntry^ prevKey,
    DatabaseEntry^ prevData,
    DatabaseEntry^ key,
    DatabaseEntry^ data,
    array<unsigned char>^% dest,
    [OutAttribute] int% size
)
```

**Parameters**

prevKey

- **Type:** BerkeleyDB::DatabaseEntry
  - The key immediately preceding the application supplied key.
prevData
Type: BerkeleyDB::DatabaseEntry
The data associated with prevKey.

key
Type: BerkeleyDB::DatabaseEntry
The application supplied key.

data
Type: BerkeleyDB::DatabaseEntry
The application supplied data.

dest
Type: array<System::Byte>[]
The compressed data to be stored in the database.

size
Type: System::Int32
The number of compressed bytes written to dest, or the required size of dest, if too small.

Return Value

True on success, false if dest is too small to contain the compressed data. All other errors should throw an exception.
See Also

BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation
BTreeCursor Class

Members  See Also

A class for traversing the records of a BTreeDatabase

Namespace:  BerkeleyDB
# Syntax

**C#**

```csharp
public class BTreeCursor : Cursor
```

**Visual Basic (Declaration)**

```vbnet
Public Class BTreeCursor
    Inherits Cursor
```

**Visual C++**

```cpp
public ref class BTreeCursor : public Cursor
```
Inheritance Hierarchy

System...Object
BerkeleyDB...BaseCursor
BerkeleyDB...Cursor
BerkeleyDB...BTreeCursor
See Also

BTreeCursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
BTreeCursor Members

The **BTreeCursor** type exposes the following members.
### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add</td>
<td>Overloaded. Insert the specified key/data pair into the database, unless a key/data pair comparing equally to it already exists in the database.</td>
</tr>
<tr>
<td>AddUnique</td>
<td>Discard the cursor. It is possible for the Close() method to throw a DeadlockException, signaling that any enclosing transaction should be aborted. If the application is already intending to abort the transaction, this error should be ignored, and the application should proceed.</td>
</tr>
<tr>
<td>Close</td>
<td>After Close has been called, regardless of its result, the object may not be used again. (Inherited from BaseCursor.)</td>
</tr>
<tr>
<td>Compare</td>
<td>Compare this cursor's position to another's. (Inherited from BaseCursor.)</td>
</tr>
<tr>
<td>Count</td>
<td>Returns a count of the number of data items for the key to which the cursor refers. (Inherited from BaseCursor.)</td>
</tr>
<tr>
<td>Delete</td>
<td>Delete the key/data pair to which the cursor refers. (Inherited from Cursor.)</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Dispose</td>
<td>Release the resources held by this object, and close the cursor if it's still open. (Inherited from <code>BaseCursor</code>.)</td>
</tr>
<tr>
<td>Duplicate</td>
<td>Create a new cursor that uses the same transaction and locker ID as the original cursor. Determines whether the specified <code>Object</code> is equal to the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>Equals</td>
<td>Returns an enumerator that iterates through the <code>Cursor</code>. (Inherited from <code>Cursor</code>.) Serves as a hash function for a particular type. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetEnumerator</td>
<td>Returns an enumerator that iterates through the <code>Cursor</code>. (Inherited from <code>Cursor</code>.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as a hash function for a particular type. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the <code>Type</code> of the current instance. (Inherited from <code>Object</code>.) Insert the data element as a duplicate element of the key to which the cursor refers.</td>
</tr>
<tr>
<td>Insert</td>
<td>Overloaded. If positioning the cursor fails, <code>CurrentMultiple</code> will contain an empty <code>KeyValuePair @(Of @(TKey, TValue))</code>.</td>
</tr>
<tr>
<td>Move</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>MoveFirst</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>MoveMultiple</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>MoveMultipleKey</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>MoveNext</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>MoveNextDuplicate</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>MoveNextDuplicateMultiple</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>MoveNextDuplicateMultipleKey</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>MoveNextDuplicateMultipleKey</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td><code>MoveNextMultiple</code></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><code>MoveNextMultipleKey</code></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><code>MoveNextUnique</code></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><code>MoveNextUniqueMultiple</code></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><code>MoveNextUniqueMultipleKey</code></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><code>MovePrev</code></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><code>MovePrevDuplicate</code></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><code>MovePrevUnique</code></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><code>Overwrite</code></td>
<td>Overwrite the data of the key/data pair to which the cursor refers with the specified data item. (Inherited from <code>Cursor</code>.)</td>
</tr>
<tr>
<td><code>Recno</code></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><code>Refresh</code></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><code>RefreshMultiple</code></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><code>RefreshMultipleKey</code></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><code>ToString</code></td>
<td>Returns a <code>String</code> that represents the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
</tbody>
</table>
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current</td>
<td>The key/data pair at which the cursor currently points. (Inherited from Cursor.)</td>
</tr>
<tr>
<td>CurrentMultiple</td>
<td>The key and multiple data items at which the cursor currently points. (Inherited from Cursor.)</td>
</tr>
<tr>
<td>CurrentMultipleKey</td>
<td>The multiple key and data items at which the cursor currently points. (Inherited from Cursor.)</td>
</tr>
<tr>
<td>Priority</td>
<td>The cache priority for pages referenced by the cursor. (Inherited from Cursor.)</td>
</tr>
</tbody>
</table>
See Also

BTreeCursor Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `BTreeCursor` type exposes the following members.
# Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add</td>
<td>Overloaded. Insert the specified key/data pair into the database, unless a key/data pair comparing equally to it already exists in the database.</td>
</tr>
<tr>
<td>AddUnique</td>
<td>Discard the cursor. It is possible for the Close() method to throw a <a href="#">DeadlockException</a>, signaling that any enclosing transaction should be aborted. If the application is already intending to abort the transaction, this error should be ignored, and the application should proceed. After Close has been called, regardless of its result, the object may not be used again.</td>
</tr>
<tr>
<td>Close</td>
<td>(Inherited from <a href="#">BaseCursor</a>.) Compare this cursor's position to another's.</td>
</tr>
<tr>
<td>Compare</td>
<td>(Inherited from <a href="#">BaseCursor</a>.) Returns a count of the number of data items for the key to which the cursor refers.</td>
</tr>
<tr>
<td>Count</td>
<td>(Inherited from <a href="#">BaseCursor</a>.) Delete the key/data pair to which the cursor refers.</td>
</tr>
<tr>
<td>Delete</td>
<td>(Inherited from <a href="#">Cursor</a>.)</td>
</tr>
</tbody>
</table>
- **Dispose**
  Release the resources held by this object, and close the cursor if it's still open.
  (Inherited from `BaseCursor`.)

- **Duplicate**
  Create a new cursor that uses the same transaction and locker ID as the original cursor.
  Determines whether the specified `Object` is equal to the current `Object`.
  (Inherited from `Object`.)

- **Equals**
  Returns an enumerator that iterates through the `Cursor`.
  (Inherited from `Cursor`.)

- **GetEnumerator**
  Serves as a hash function for a particular type.
  (Inherited from `Object`.)

- **GetHashCode**
  Gets the `Type` of the current instance.
  (Inherited from `Object`.)

- **GetType**
  Insert the data element as a duplicate element of the key to which the cursor refers.
  Overloaded.

- **Move**
  Overloaded.

- **MoveFirst**
  Overloaded.

- **MoveMultiple**
  If positioning the cursor fails, `CurrentMultiple` will contain an empty `KeyValuePair<(Of <(TKey, TValue)>)>`.

- **MoveFirstMultiple**
  Overloaded.

- **MoveFirstMultipleKey**
  Overloaded.

- **MoveLast**
  Overloaded.

- **MoveMultiple**
  Overloaded.

- **MoveMultipleKey**
  Overloaded.

- **MoveNext**
  Overloaded.

- **MoveNextDuplicate**
  Overloaded.

- **MoveNextDuplicateMultiple**
  Overloaded.

- **MoveNextDuplicateMultipleKey**
  Overloaded.
<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MoveNextMultiple</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>MoveNextMultipleKey</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>MoveNextUnique</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>MoveNextUniqueMultiple</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>MoveNextUniqueMultipleKey</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>MovePrev</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>MovePrevDuplicate</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>MovePrevUnique</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>Overwrite</td>
<td>Overwrite the data of the key/data pair to which the cursor refers with the specified data item. (Inherited from Cursor.)</td>
</tr>
<tr>
<td>Recno</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>Refresh</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>RefreshMultiple</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>RefreshMultipleKey</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a String that represents the current Object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>
See Also

BTreeCursor Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
BTreeCursor Class  See Also
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add(KeyValuePair&lt;br&gt;(Of DatabaseEntry, DatabaseEntry)&gt;))</td>
<td>Stores the key/data pair in the database. (Inherited from Cursor.)</td>
</tr>
<tr>
<td>Add(KeyValuePair&lt;br&gt;(Of DatabaseEntry, DatabaseEntry)&gt;), Cursor...::InsertLocation)</td>
<td>Insert the specified key/data pair into the database.</td>
</tr>
</tbody>
</table>
See Also

BTreeCursor Class
BTreeCursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
BTreeCursor..::..Add Method (KeyValuePair(Of (DatabaseEntry, DatabaseEntry)>), Cursor..::..InsertLocation)

BTreeCursor Class  See Also

Insert the specified key/data pair into the database.

Namespace:  BerkeleyDB
Syntax

C#

public void Add(
    KeyValuePair<DatabaseEntry, DatabaseEntry> pair,
    Cursor::<::InsertLocation loc
)

Visual Basic (Declaration)

Public Sub Add (_
    pair As KeyValuePair(Of DatabaseEntry, DatabaseEntry), _
    loc As Cursor::<::InsertLocation _
)

Visual C++

public:
void Add(
    KeyValuePair<DatabaseEntry^, DatabaseEntry^> pair,
    Cursor::<::InsertLocation loc
)

Parameters

pair
Type: System.Collections.Generic::<::KeyValuePair<((DatabaseEntry,
    DatabaseEntry)>))
The key/data pair to be inserted

loc
Type: BerkeleyDB::<::Cursor::<::InsertLocation
If the key already exists in the database and no duplicate sort function has
been specified, specify whether the inserted data item is added as the first or
the last of the data items for that key.
See Also

BTreeCursor Class
Add Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
BTreeCursor Class  See Also

Insert the specified key/data pair into the database, unless a key/data pair comparing equally to it already exists in the database.

Namespace: BerkeleyDB
Syntax

C#

public void AddUnique(
    KeyValuePair<DatabaseEntry, DatabaseEntry> pair
)

Visual Basic (Declaration)

Public Sub AddUnique ( _
    pair As KeyValuePair(Of DatabaseEntry, DatabaseEntry) _
)

Visual C++

public:
void AddUnique(
    KeyValuePair<DatabaseEntry^, DatabaseEntry^> pair
)

Parameters

pair
    Type: System.Collections.Generic..::.KeyValuePair<(Of <(DatabaseEntry,
        DatabaseEntry)>))
    The key/data pair to be inserted
## Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>BerkeleyDB::KeyExistsException</td>
<td>Thrown if a matching key/data pair already exists in the database.</td>
</tr>
</tbody>
</table>
See Also

BTreeCursor Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Create a new cursor that uses the same transaction and locker ID as the original cursor.

Namespace: BerkeleyDB
Syntax

C#

```csharp
public BTreeCursor Duplicate(
    bool keepPosition
)
```

Visual Basic (Declaration)

```vbnet
Public Function Duplicate (_
    keepPosition As Boolean _
) As BTreeCursor
```

Visual C++

```cpp
public:
BTreeCursor^ Duplicate(
    bool keepPosition
)
```

Parameters

keepPosition
Type: System::::Boolean
If true, the newly created cursor is initialized to refer to the same position in the database as the original cursor (if any) and hold the same locks (if any). If false, or the original cursor does not hold a database position and locks, the created cursor is uninitialized and will behave like a cursor newly created by Cursor().

Return Value

A newly created cursor
Remarks

This is useful when an application is using locking and requires two or more cursors in the same thread of control.
See Also

BTreeCursor Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
BTreeCursor::Insert Method

Insert the data element as a duplicate element of the key to which the cursor refers.

Namespace: BerkeleyDB
Syntax

C#

public void Insert(
    DatabaseEntry data,
    Cursor..::..InsertLocation loc
)

Visual Basic (Declaration)

Public Sub Insert ( _
    data As DatabaseEntry, _
    loc As Cursor..::..InsertLocation _
)

Visual C++

public:
    void Insert(
        DatabaseEntry^ data,
        Cursor..::..InsertLocation loc
    )

Parameters

data
    Type: BerkeleyDB..::..DatabaseEntry
    The data element to insert

loc
    Type: BerkeleyDB..::..Cursor..::..InsertLocation
    Specify whether to insert the data item immediately before or immediately after the cursor's current position.
See Also

BTreeCursor Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
BTreeCursor Class  See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Move(UInt32)</strong></td>
<td>Position the cursor at a specific key/data pair in the database, and store the key/data pair in <a href="#">Current</a>.</td>
</tr>
<tr>
<td><strong>Move(UInt32, LockingInfo)</strong></td>
<td>Position the cursor at a specific key/data pair in the database, and store the key/data pair in <a href="#">Current</a>.</td>
</tr>
<tr>
<td><strong>Move(DatabaseEntry, Boolean)</strong></td>
<td>Set the cursor to refer to key, and store the datum associated with the given key in <a href="#">Current</a>. In the presence of duplicate key values, the first data item in the set of duplicates is stored in <a href="#">Current</a>. (Inherited from <a href="#">Cursor</a>.)</td>
</tr>
<tr>
<td><strong>Move(KeyValuePair(Of (DatabaseEntry, DatabaseEntry)&gt;, Boolean))</strong></td>
<td>Move the cursor to the specified key/data pair of the database. The cursor is positioned to a key/data pair if both the key and data match the values provided on the key and data parameters. (Inherited from <a href="#">Cursor</a>.)</td>
</tr>
<tr>
<td><strong>Move(DatabaseEntry, Boolean, LockingInfo)</strong></td>
<td>Set the cursor to refer to key, and store the datum associated with the given key in <a href="#">Current</a>. In the presence of duplicate key values, the first data item in the set of duplicates is stored in <a href="#">Current</a>. (Inherited from <a href="#">Cursor</a>.)</td>
</tr>
<tr>
<td><strong>Move(KeyValuePair(Of (DatabaseEntry, DatabaseEntry)&gt;, Boolean, LockingInfo))</strong></td>
<td>Move the cursor to the specified key/data pair of the database. The cursor is positioned to a key/data pair if both the key and data match the values provided on the key and data parameters. (Inherited from <a href="#">Cursor</a>.)</td>
</tr>
</tbody>
</table>
See Also

BTreeCursor Class
BTreeCursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
BTreeCursor Class  See Also

Position the cursor at a specific key/data pair in the database, and store the key/data pair in Current.

Namespace:  BerkeleyDB
Syntax

C#

```csharp
public bool Move(
    uint recno
)
```

Visual Basic (Declaration)

```vbnet
Public Function Move ( _
    recno As UInteger _
) As Boolean
```

Visual C++

```cpp
public:
    bool Move(
        unsigned int recno
    )
```

Parameters

recno

Type: `System::::UInt32`

The specific numbered record of the database at which to position the cursor.

Return Value

True if the cursor was positioned successfully, false otherwise.
See Also

BTreeCursor Class
Move Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
BTreeCursor...::Move Method (UInt32, LockingInfo)

Position the cursor at a specific key/data pair in the database, and store the key/data pair in Current.

**Namespace:** BerkeleyDB
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool Move(
    uint recno,
    LockingInfo info
)

Visual Basic (Declaration)

Public Function Move ( _
    recno AsUInteger, _
    info As LockingInfo _
) As Boolean

Visual C++

public:
  bool Move(
      unsigned int recno,
      LockingInfo^ info
  )

Parameters

recno
  Type: System::UInt32
  The specific numbered record of the database at which to position the cursor.

info
  Type: BerkeleyDB::::LockingInfo
  The locking behavior to use

Return Value

True if the cursor was positioned successfully, false otherwise.
See Also

BTreeCursor Class
Move Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
BTreeCursor Class  See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MoveFirst()()()</td>
<td>Set the cursor to refer to the first key/data pair of the database, and store that pair in Current. If the first key has duplicate values, the first data item in the set of duplicates is stored in Current. (Inherited from Cursor.)</td>
</tr>
<tr>
<td>MoveFirst(LockingInfo)</td>
<td>Set the cursor to refer to the first key/data pair of the database, and store that pair in Current. If the first key has duplicate values, the first data item in the set of duplicates is stored in Current. (Inherited from Cursor.)</td>
</tr>
</tbody>
</table>
See Also

BTreeCursor Class
BTreeCursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If positioning the cursor fails, **CurrentMultiple** will contain an empty
**KeyValuePair<(Of <(TKey, TValue)>)>**.
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MoveFirstMultiple()()</td>
<td>Set the cursor to refer to the first key/data pair of the database, and store that key and as many duplicate data items that can fit in a buffer the size of one database page in CurrentMultiple. (Inherited from Cursor.)</td>
</tr>
<tr>
<td>MoveFirstMultiple(LockingInfo)</td>
<td>Set the cursor to refer to the first key/data pair of the database, and store that key and as many duplicate data items that can fit in a buffer the size of one database page in CurrentMultiple. (Inherited from Cursor.)</td>
</tr>
<tr>
<td>MoveFirstMultiple(Int32)</td>
<td>Set the cursor to refer to the first key/data pair of the database, and store that key and as many duplicate data items that can fit in a buffer the size of BufferSize in CurrentMultiple. (Inherited from Cursor.)</td>
</tr>
<tr>
<td>MoveFirstMultiple(Int32, LockingInfo)</td>
<td>Set the cursor to refer to the first key/data pair of the database, and store that key and as many duplicate data items that can fit in a buffer the size of BufferSize in CurrentMultiple. (Inherited from Cursor.)</td>
</tr>
</tbody>
</table>
See Also

BTreeCursor Class
BTreeCursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
BTreeCursor Class   See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MoveFirstMultipleKey()()</td>
<td>Set the cursor to refer to the first key/data pair of the database, and store that pair and as many ensuing key/data pairs that can fit in a buffer the size of one database page in CurrentMultipleKey. (Inherited from Cursor.)</td>
</tr>
<tr>
<td>MoveFirstMultipleKey(LockingInfo)</td>
<td>Set the cursor to refer to the first key/data pair of the database, and store that pair and as many ensuing key/data pairs that can fit in a buffer the size of one database page in CurrentMultipleKey. (Inherited from Cursor.)</td>
</tr>
<tr>
<td>MoveFirstMultipleKey(Int32)</td>
<td>Set the cursor to refer to the first key/data pair of the database, and store that pair and as many ensuing key/data pairs that can fit in a buffer the size of BufferSize in CurrentMultipleKey. (Inherited from Cursor.)</td>
</tr>
<tr>
<td>MoveFirstMultipleKey(Int32, LockingInfo)</td>
<td>Set the cursor to refer to the first key/data pair of the database, and store that pair and as many ensuing key/data pairs that can fit in a buffer the size of BufferSize in CurrentMultipleKey. (Inherited from Cursor.)</td>
</tr>
</tbody>
</table>
See Also

BTreeCursor Class
BTreeCursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
BTreeCursor BTreeCursor.....MoveLast Method
BTreeCursor Class  See Also
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MoveLast</td>
<td>Set the cursor to refer to the last key/data pair of the database, and store that pair in <code>Current</code>. If the last key has duplicate values, the last data item in the set of duplicates is stored in <code>Current</code>. (Inherited from <code>Cursor</code>.)</td>
</tr>
<tr>
<td>MoveLast(LockingInfo)</td>
<td>Set the cursor to refer to the last key/data pair of the database, and store that pair in <code>Current</code>. If the last key has duplicate values, the last data item in the set of duplicates is stored in <code>Current</code>. (Inherited from <code>Cursor</code>.)</td>
</tr>
</tbody>
</table>
See Also

BTreeCursor Class
BTreeCursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
BTreeCursor.BTreeCursor.MoveMultiple Method

BTreeCursor Class  See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MoveMultiple(UInt32)</strong></td>
<td>Position the cursor at a specific key/data pair in the database, and store the key/data pair and as many duplicate data items that can fit in a buffer the size of one database page in CurrentMultiple.</td>
</tr>
<tr>
<td><strong>MoveMultiple(UInt32, LockingInfo)</strong></td>
<td>Position the cursor at a specific key/data pair in the database, and store the key/data pair and as many duplicate data items that can fit in a buffer the size of one database page in CurrentMultiple.</td>
</tr>
<tr>
<td><strong>MoveMultiple(UInt32, Int32)</strong></td>
<td>Position the cursor at a specific key/data pair in the database, and store the key/data pair and as many duplicate data items that can fit in a buffer the size of one database page in CurrentMultiple.</td>
</tr>
<tr>
<td><strong>MoveMultiple(DatabaseEntry, Boolean)</strong></td>
<td>Set the cursor to refer to key, and store that key and as many duplicate data items associated with the given key that can fit in a buffer the size of one database page in CurrentMultiple. (Inherited from Cursor.)</td>
</tr>
<tr>
<td><strong>MoveMultiple(KeyValuePair(Of DatabaseEntry, )</strong></td>
<td>Move the cursor to the specified key/data pair of the database, and store that key/data pair and as many duplicate data items associated with the given key that can fit in a buffer the size of one database page in CurrentMultiple. The cursor is</td>
</tr>
</tbody>
</table>
MoveMultiple(DatabaseEntry, Boolean, LockingInfo)

MoveMultiple<DatabaseEntry>, Boolean)

Positioned to a key/data pair if both the key and data match the values provided on the key and data parameters. (Inherited from Cursor.)

MoveMultiple(UInt32, Int32, LockingInfo)

MoveMultiple<UInt32, Int32, LockingInfo>

Position the cursor at a specific key/data pair in the database, and store the key/data pair and as many duplicate data items that can fit in a buffer the size of one database page in CurrentMultiple.

MoveMultiple(DatabaseEntry, Boolean, LockingInfo)

MoveMultiple<DatabaseEntry, Boolean, LockingInfo>

Set the cursor to refer to key, and store that key and as many duplicate data items associated with the given key that can fit in a buffer the size of one database page in CurrentMultiple. (Inherited from Cursor.)

MoveMultiple(DatabaseEntry, Boolean, Int32)

MoveMultiple<DatabaseEntry, Boolean, Int32>

Set the cursor to refer to key, and store that key and as many duplicate data items associated with the given key that can fit in a buffer the size of BufferSize in CurrentMultiple. (Inherited from Cursor.)

MoveMultiple< KeyValuePair< Of <(DatabaseEntry, DatabaseEntry)>, Boolean, LockingInfo>

MoveMultiple< KeyValuePair< Of <(DatabaseEntry, DatabaseEntry)>, Boolean, LockingInfo>

Move the cursor to the specified key/data pair of the database, and store that key/data pair and as many duplicate data items associated with the given key that can fit in a buffer the size of one database page in CurrentMultiple. The cursor is positioned to a key/data pair if both the key and data match the values provided on the key and data parameters. (Inherited from Cursor.)

Move the cursor to the specified key/data pair in the database.
MoveMultiple(KeyValuePair<
(DatabaseEntry,
DatabaseEntry)>), Boolean,
Int32)

key/data pair of the database, and
store that key/data pair and as many
duplicate data items associated with
the given key that can fit in a buffer
the size of BufferSize in
CurrentMultiple. The cursor is
positioned to a key/data pair if both
the key and data match the values
provided on the key and data
parameters.
(Inherited from Cursor.)

MoveMultiple(DatabaseEntry,
Boolean, Int32, LockingInfo)

Set the cursor to refer to key, and
store that key and as many duplicate
data items associated with the given
key that can fit in a buffer the size of
BufferSize in CurrentMultiple.
(Inherited from Cursor.)

MoveMultiple(KeyValuePair<
(DatabaseEntry,
DatabaseEntry)>), Boolean,
Int32, LockingInfo)

Move the cursor to the specified
key/data pair of the database, and
store that key/data pair and as many
duplicate data items associated with
the given key that can fit in a buffer
the size of BufferSize in
CurrentMultiple. The cursor is
positioned to a key/data pair if both
the key and data match the values
provided on the key and data
parameters.
(Inherited from Cursor.)
See Also

BTreeCursor Class
BTreeCursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Position the cursor at a specific key/data pair in the database, and store the key/data pair and as many duplicate data items that can fit in a buffer the size of one database page in CurrentMultiple.

**Namespace:**  BerkeleyDB  
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll)  
**Version:** 4.8.24.0
Syntax

C#

public bool MoveMultiple(
    uint recno
)

Visual Basic (Declaration)

Public Function MoveMultiple (_
    recno As UInteger _
) As Boolean

Visual C++

public:
    bool MoveMultiple(
        unsigned int recno
    )

Parameters

recno

  Type: System::::UInt32
  The specific numbered record of the database at which to position the cursor.

Return Value

True if the cursor was positioned successfully, false otherwise.
See Also

BTreeCursor Class
MoveMultiple Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
BTreeCursor::MoveMultiple Method (UInt32, LockingInfo)

Position the cursor at a specific key/data pair in the database, and store the key/data pair and as many duplicate data items that can fit in a buffer the size of one database page in CurrentMultiple.

Namespace: BerkeleyDB
Syntax

C#

```csharp
public bool MoveMultiple(
    uint recno,
    LockingInfo info
)
```

Visual Basic (Declaration)

```vbnet
Public Function MoveMultiple ( _
    recno As UInteger, _
    info As LockingInfo _
) As Boolean
```

Visual C++

```cpp
public:
bool MoveMultiple(
    unsigned int recno,
    LockingInfo^ info
)
```

Parameters

recno
Type: System::UInt32
The specific numbered record of the database at which to position the cursor.

info
Type: BerkeleyDB::LockingInfo
The locking behavior to use

Return Value

True if the cursor was positioned successfully, false otherwise.
See Also

BTreeCursor Class
MoveMultiple Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Position the cursor at a specific key/data pair in the database, and store the key/data pair and as many duplicate data items that can fit in a buffer the size of one database page in `CurrentMultiple`.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

```csharp
public bool MoveMultiple(
    uint recno,
    int BufferSize
)
```

Visual Basic (Declaration)

```vbnet
Public Function MoveMultiple ( _
    recno As UInteger, _
    BufferSize As Integer _
) As Boolean
```

Visual C++

```cpp
public:
    bool MoveMultiple(
        unsigned int recno,
        int BufferSize
    )
```

Parameters

recno
Type: System::::UInt32
The specific numbered record of the database at which to position the cursor.

BufferSize
Type: System::::Int32
The size of a buffer to fill with duplicate data items. Must be at least the page size of the underlying database and be a multiple of 1024.

Return Value
True if the cursor was positioned successfully, false otherwise.
**See Also**

_BTreeCursor Class_
_MoveMultiple Overload_
_BerkeleyDB Namespace_

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Position the cursor at a specific key/data pair in the database, and store the key/data pair and as many duplicate data items that can fit in a buffer the size of one database page in `CurrentMultiple`.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

```csharp
public bool MoveMultiple(
    uint recno,
    int BufferSize,
    LockingInfo info
)
```

Visual Basic (Declaration)

```vbnet
Public Function MoveMultiple (_
    recno As UInteger, _
    BufferSize As Integer, _
    info As LockingInfo _
) As Boolean
```

Visual C++

```cpp
public:
    bool MoveMultiple(
        unsigned int recno,
        int BufferSize,
        LockingInfo^ info
    )
```

Parameters

recno
  Type: `System::::UInt32`
  The specific numbered record of the database at which to position the cursor.

BufferSize
  Type: `System::::Int32`
  The size of a buffer to fill with duplicate data items. Must be at least the page size of the underlying database and be a multiple of 1024.
info

Type: BerkeleyDB::LockingInfo
The locking behavior to use

Return Value

True if the cursor was positioned successfully, false otherwise.
See Also

BTreeCursor Class
MoveMultiple Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
BTreeCursor::MoveMultipleKey Method

BTreeCursor Class  See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MoveMultipleKey(UInt32)</strong></td>
<td>Position the cursor at a specific key/data pair in the database, and store the key/data pair and as many ensuing key/data pairs that can fit in a buffer the size of one database page in <strong>CurrentMultipleKey</strong>.</td>
</tr>
<tr>
<td><strong>MoveMultipleKey(UInt32, LockingInfo)</strong></td>
<td>Position the cursor at a specific key/data pair in the database, and store the key/data pair and as many ensuing key/data pairs that can fit in a buffer the size of one database page in <strong>CurrentMultipleKey</strong>.</td>
</tr>
<tr>
<td><strong>MoveMultipleKey(UInt32, Int32)</strong></td>
<td>Position the cursor at a specific key/data pair in the database, and store the key/data pair and as many ensuing key/data pairs that can fit in a buffer the size of one database page in <strong>CurrentMultipleKey</strong>.</td>
</tr>
<tr>
<td><strong>MoveMultipleKey(DatabaseEntry, Boolean)</strong></td>
<td>Set the cursor to refer to key, and store that key and as many ensuing key/data pairs that can fit in a buffer the size of one database page in <strong>CurrentMultipleKey</strong>. (Inherited from <strong>Cursor</strong>.) Move the cursor to the specified key/data pair of the database, and store that key/data pair and as many ensuing key/data pairs</td>
</tr>
</tbody>
</table>
MoveMultipleKey(KeyValuePair<(Of (DatabaseEntry, DatabaseEntry)>)>, Boolean)

that can fit in a buffer the size of one database page in CurrentMultipleKey. The cursor is positioned to a key/data pair if both the key and data match the values provided on the key and data parameters. (Inherited from Cursor.)

MoveMultipleKey(UInt32, Int32, LockingInfo)

Position the cursor at a specific key/data pair in the database, and store the key/data pair and as many ensuing key/data pairs that can fit in a buffer the size of one database page in CurrentMultipleKey.

MoveMultipleKey(DatabaseEntry, Boolean, LockingInfo)

Set the cursor to refer to key, and store that key and as many ensuing key/data pairs that can fit in a buffer the size of one database page in CurrentMultipleKey. (Inherited from Cursor.)

MoveMultipleKey(DatabaseEntry, Boolean, Int32)

Set the cursor to refer to key, and store that key and as many ensuing key/data pairs that can fit in a buffer the size of BufferSize in CurrentMultipleKey. (Inherited from Cursor.)

MoveMultipleKey(KeyValuePair<(Of (DatabaseEntry, DatabaseEntry)>)>, Boolean, LockingInfo)

Move the cursor to the specified key/data pair of the database, and store that key/data pair and as many ensuing key/data pairs that can fit in a buffer the size of one database page in CurrentMultipleKey. The cursor is positioned to a key/data pair if both the key and data match the
MoveMultipleKey(KeyValuePair(Of <(DatabaseEntry, DatabaseEntry)>>, Boolean, Int32)

Move the cursor to the specified key/data pair of the database, and store that key/data pair and as many ensuing key/data pairs that can fit in a buffer the size of BufferSize in CurrentMultipleKey. The cursor is positioned to a key/data pair if both the key and data match the values provided on the key and data parameters. (Inherited from Cursor.)

MoveMultipleKey(DatabaseEntry, Boolean, Int32, LockingInfo)

Set the cursor to refer to key, and store that key and as many ensuing key/data pairs that can fit in a buffer the size of BufferSize in CurrentMultipleKey. (Inherited from Cursor.)

MoveMultipleKey(KeyValuePair(Of <(DatabaseEntry, DatabaseEntry)>), Boolean, Int32, LockingInfo)

Move the cursor to the specified key/data pair of the database, and store that key/data pair and as many ensuing key/data pairs that can fit in a buffer the size of BufferSize in CurrentMultipleKey. The cursor is positioned to a key/data pair if both the key and data match the values provided on the key and data parameters. (Inherited from Cursor.)
See Also

BTreeCursor Class
BTreeCursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Position the cursor at a specific key/data pair in the database, and store the key/data pair and as many ensuing key/data pairs that can fit in a buffer the size of one database page in **CurrentMultipleKey**.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

```csharp
public bool MoveMultipleKey(
    uint recno
)
```

Visual Basic (Declaration)

```vbnet
Public Function MoveMultipleKey ( _
    recno As UInteger _
) As Boolean
```

Visual C++

```cpp
public:
    bool MoveMultipleKey(
        unsigned int recno
    )
```

Parameters

recno

Type: `System::::UInt32`

The specific numbered record of the database at which to position the cursor.

Return Value

True if the cursor was positioned successfully, false otherwise.
See Also

BTreeCursor Class
MoveMultipleKey Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Position the cursor at a specific key/data pair in the database, and store the key/data pair and as many ensuing key/data pairs that can fit in a buffer the size of one database page in `CurrentMultipleKey`.

_namespace: BerkeleyDB_

Syntax

C#

```csharp
public bool MoveMultipleKey(
    uint recno,
    LockingInfo info
)
```

Visual Basic (Declaration)

```vbnet
Public Function MoveMultipleKey (_
    recno As UInteger, _
    info As LockingInfo _
) As Boolean
```

Visual C++

```cpp
public:
bool MoveMultipleKey(
    unsigned int recno,
    LockingInfo info
)
```

Parameters

recno
Type: `System::::UInt32`
The specific numbered record of the database at which to position the cursor.

info
Type: `BerkeleyDB::::LockingInfo`
The locking behavior to use

Return Value

True if the cursor was positioned successfully, false otherwise.
See Also

BTreeCursor Class
MoveMultipleKey Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
BTreeCursor::MoveMultipleKey Method (UInt32, Int32)

Position the cursor at a specific key/data pair in the database, and store the key/data pair and as many ensuing key/data pairs that can fit in a buffer the size of one database page in CurrentMultipleKey.

Namespace: BerkeleyDB
Syntax

C#

public bool MoveMultipleKey(
    uint recno,
    int BufferSize
)

Visual Basic (Declaration)

Public Function MoveMultipleKey (_
    recno As UInteger, _
    BufferSize As Integer _
) As Boolean

Visual C++

public:
    bool MoveMultipleKey(
        unsigned int recno, 
        int BufferSize
    )

Parameters

recno
Type: System::::UInt32
The specific numbered record of the database at which to position the cursor.

BufferSize
Type: System::::Int32
The size of a buffer to fill with key/data pairs. Must be at least the page size of the underlying database and be a multiple of 1024.

Return Value
True if the cursor was positioned successfully, false otherwise.
See Also

BTreeCursor Class
MoveMultipleKey Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Position the cursor at a specific key/data pair in the database, and store the key/data pair and as many ensuing key/data pairs that can fit in a buffer the size of one database page in `CurrentMultipleKey`.

**Namespace:**  [BerkeleyDB](https://berkeleydb.net/)

**Assembly:**  `libdb_dotnet48` (in `libdb_dotnet48.dll`) Version: 4.8.24.0
### Syntax

#### C#

```csharp
class YourClass
{
    public bool MoveMultipleKey(
        uint recno,
        int BufferSize,
        LockingInfo info
    )
}
```

#### Visual Basic (Declaration)

```vbnet
Public Function MoveMultipleKey ( _
    recno As UInteger, _
    BufferSize As Integer, _
    info As LockingInfo _
) As Boolean
```

#### Visual C++

```cpp
public:
    bool MoveMultipleKey(
        unsigned int recno,
        int BufferSize,
        LockingInfo^ info
    )
```

### Parameters

- **recno**
  - Type: `System::::UInt32`
  - The specific numbered record of the database at which to position the cursor.

- **BufferSize**
  - Type: `System::::Int32`
  - The size of a buffer to fill with key/data pairs. Must be at least the page size of the underlying database and be a multiple of 1024.
info
Type: BerkeleyDB::::LockingInfo
The locking behavior to use

Return Value

True if the cursor was positioned successfully, false otherwise.
See Also

BTreeCursor Class
MoveMultipleKey Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
BTreeCursor Class  See Also
### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MoveNext()</td>
<td>If the cursor is not yet initialized, MoveNext is identical to MoveFirst(). Otherwise, move the cursor to the next key/data pair of the database, and store that pair in Current. In the presence of duplicate key values, the value of Current.Key may not change. (Inherited from Cursor.)</td>
</tr>
<tr>
<td>MoveNext(LockingInfo)</td>
<td>If the cursor is not yet initialized, MoveNext is identical to MoveFirst(LockingInfo). Otherwise, move the cursor to the next key/data pair of the database, and store that pair in Current. In the presence of duplicate key values, the value of Current.Key may not change. (Inherited from Cursor.)</td>
</tr>
</tbody>
</table>
See Also

BTreeCursor Class  
BTreeCursor Members  
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
BTreeCursor Class  See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MoveNextDuplicate()()()</td>
<td>If the next key/data pair of the database is a duplicate data record for the current key/data pair, move the cursor to the next key/data pair in the database, and store that pair in <code>Current</code>. MoveNextDuplicate will return false if the next key/data pair of the database is not a duplicate data record for the current key/data pair. (Inherited from <code>Cursor</code>.)</td>
</tr>
<tr>
<td>MoveNextDuplicate(LockingInfo)()</td>
<td>MoveNextDuplicate will return false if the next key/data pair of the database is not a duplicate data record for the current key/data pair. (Inherited from <code>Cursor</code>.)</td>
</tr>
</tbody>
</table>
See Also

BTreeCursor Class
BTreeCursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
C#  Visual Basic  Visual C++  Include Protected Members  Include Inherited Members

Berkeley DB .NET API Documentation

BTreeCursor...:MoveNextDuplicateMultiple Method

BTreeCursor Class  See Also
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MoveNextDuplicateMultiple()()()</td>
<td>If the next key/data pair of the database is a duplicate data record for the current key/data pair, move the cursor to the next key/data pair in the database, and store that pair and as many duplicate data items that can fit in a buffer the size of one database page in CurrentMultiple. MoveNextDuplicateMultiple will return false if the next key/data pair of the database is not a duplicate data record for the current key/data pair. (Inherited from Cursor.)</td>
</tr>
<tr>
<td>MoveNextDuplicateMultiple(LockingInfo)</td>
<td>If the next key/data pair of the database is a duplicate data record for the current key/data pair, move the cursor to the next key/data pair in the database, and store that pair and as many duplicate data items that can fit in a buffer the size of one database page in CurrentMultiple. MoveNextDuplicateMultiple will return false if the next key/data pair of the database is not a duplicate data record for the current key/data pair.</td>
</tr>
</tbody>
</table>
MoveNextDuplicateMultiple(Int32)

If the next key/data pair of the database is a duplicate data record for the current key/data pair, then move cursor to the next key/data pair in the database, and store that pair and as many duplicate data items that can fit in a buffer the size of BufferSize in CurrentMultiple.

MoveNextDuplicateMultiple will return false if the next key/data pair of the database is not a duplicate data record for the current key/data pair.

(Inherited from Cursor.)

MoveNextDuplicateMultiple(Int32, LockingInfo)

MoveNextDuplicateMultiple will return false if the next key/data pair of the database is not a duplicate data record for the current key/data pair.

(Inherited from Cursor.)
See Also

BTreeCursor Class
BTreeCursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation
BTreeCursor::MoveNextDuplicateMultipleKey Method

BTreeCursor Class  See Also
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MoveNextDuplicateMultipleKey()</strong></td>
<td>If the next key/data pair of the database is a duplicate data record for the current key/data pair, move the cursor to the next key/data pair in the database, and store that pair and as many duplicate data items that can fit in a buffer the size of one database page in <code>CurrentMultipleKey</code>. <code>MoveNextDuplicateMultipleKey</code> will return false if the next key/data pair of the database is not a duplicate data record for the current key/data pair. (Inherited from <code>Cursor</code>.)</td>
</tr>
<tr>
<td><strong>MoveNextDuplicateMultipleKey(LockingInfo)</strong></td>
<td>If the next key/data pair of the database is a duplicate data record for the current key/data pair, move the cursor to the next key/data pair in the database, and store that pair and as many duplicate data items that can fit in a buffer the size of one database page in <code>CurrentMultipleKey</code>. <code>MoveNextDuplicateMultipleKey</code> will return false if the next key/data pair of the database is not a duplicate data record for the current key/data pair. (Inherited from <code>Cursor</code>.)</td>
</tr>
</tbody>
</table>
MoveNextDuplicateMultipleKey(Int32)

database is a duplicate data record for the current key/data pair, move the cursor to the next key/data pair in the database, and store that pair and as many duplicate data items that can fit in a buffer the size of BufferSize in CurrentMultipleKey. MoveNextDuplicateMultipleKey will return false if the next key/data pair of the database is not a duplicate data record for the current key/data pair. (Inherited from Cursor.)

MoveNextDuplicateMultipleKey(Int32, LockingInfo)

If the next key/data pair of the database is a duplicate data record for the current key/data pair, move the cursor to the next key/data pair in the database, and store that pair and as many duplicate data items that can fit in a buffer the size of BufferSize in CurrentMultipleKey. MoveNextDuplicateMultipleKey will return false if the next key/data pair of the database is not a duplicate data record for the current key/data pair. (Inherited from Cursor.)
See Also

BTreeCursor Class
BTreeCursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
BTreeCursor Class  See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MoveNextMultiple()()</td>
<td>If the cursor is not yet initialized, MoveNextMultiple is identical to MoveFirstMultiple(). Otherwise, move the cursor to the next key/data pair of the database, and store that pair and as many duplicate data items that can fit in a buffer the size of one database page in CurrentMultiple. In the presence of duplicate key values, the value of CurrentMultiple.Key may not change. (Inherited from Cursor.)</td>
</tr>
<tr>
<td>MoveNextMultiple(LockingInfo)</td>
<td>If the cursor is not yet initialized, MoveNextMultiple is identical to MoveFirstMultiple(LockingInfo). Otherwise, move the cursor to the next key/data pair of the database, and store that pair and as many duplicate data items that can fit in a buffer the size of one database page in CurrentMultiple. In the presence of duplicate key values, the value of CurrentMultiple.Key may not change. (Inherited from Cursor.)</td>
</tr>
<tr>
<td>MoveNextMultiple(Int32)</td>
<td>If the cursor is not yet initialized, MoveNextMultiple is identical to MoveFirstMultiple(Int32). Otherwise, move the cursor to the next key/data pair of the database, and store that pair and as many duplicate data items that can fit in a buffer the size of BufferSize in CurrentMultiple. In the presence of duplicate key values, the</td>
</tr>
</tbody>
</table>
MoveNextMultiple(Int32, LockingInfo)

value of `CurrentMultiple.Key` may not change.
(Inherited from `Cursor`.)

If the cursor is not yet initialized, `MoveNextMultiple` is identical to `MoveFirstMultiple(Int32, LockingInfo)`. Otherwise, move the cursor to the next key/data pair of the database, and store that pair and as many duplicate data items that can fit in a buffer the size of `BufferSize` in `CurrentMultiple`. In the presence of duplicate key values, the value of `CurrentMultiple.Key` may not change.
(Inherited from `Cursor`.)
See Also

BTreeCursor Class
BTreeCursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
C#  Visual Basic  Visual C++  Include Protected Members  Include Inherited Members
Berkeley DB .NET API Documentation
BTreeCursor...::MoveNextMultipleKey Method

BTreeCursor Class  See Also
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MoveNextMultipleKey()()()</td>
<td>If the cursor is not yet initialized, MoveNextMultipleKey is identical to MoveFirstMultipleKey(). Otherwise, move the cursor to the next key/data pair of the database, and store that pair and as many ensuing key/data pairs that can fit in a buffer the size of one database page in CurrentMultipleKey. In the presence of duplicate key values, the keys of CurrentMultipleKey may not change. (Inherited from Cursor.)</td>
</tr>
<tr>
<td>MoveNextMultipleKey(LockingInfo)</td>
<td>Ensuing key/data pairs that can fit in a buffer the size of one database page in CurrentMultipleKey. In the presence of duplicate key values, the keys of CurrentMultipleKey may not change. (Inherited from Cursor.)</td>
</tr>
<tr>
<td>MoveNextMultipleKey(Int32)</td>
<td>Ensuing key/data pairs that can fit in a buffer the size of one database page in CurrentMultipleKey. In the presence of duplicate key values, the keys of CurrentMultipleKey may not change. (Inherited from Cursor.)</td>
</tr>
</tbody>
</table>

If the cursor is not yet initialized, MoveNextMultipleKey is identical to MoveFirstMultipleKey(). Otherwise, move the cursor to the next key/data pair of the database, and store that pair and as many ensuing key/data pairs that can fit in a buffer the size of one database page in CurrentMultipleKey. In the presence of duplicate key values, the keys of CurrentMultipleKey may not change. (Inherited from Cursor.)
buffer the size of BufferSize in CurrentMultipleKey. In the presence of duplicate key values, the keys of CurrentMultipleKey may not change. (Inherited from Cursor.)

If the cursor is not yet initialized, MoveNextMultipleKey is identical to MoveFirstMultipleKey(Int32, LockingInfo). Otherwise, move the cursor to the next key/data pair of the database, and store that pair and as many ensuing key/data pairs that can fit in a buffer the size of BufferSize in CurrentMultipleKey. In the presence of duplicate key values, the keys of CurrentMultipleKey may not change.
(Inherited from Cursor.)
See Also

BTreeCursor Class
BTreeCursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
BTreeCursor Class  See Also

BTreeCursor...::MoveNextUnique Method

BTreeCursor Class  See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MoveNextUnique()()()</td>
<td>If the cursor is not yet initialized, MoveNextUnique is identical to MoveFirst(). Otherwise, move the cursor to the next non-duplicate key in the database, and store that key and associated datum in Current. MoveNextUnique will return false if no non-duplicate key/data pairs exist after the cursor position in the database. (Inherited from Cursor.)</td>
</tr>
<tr>
<td>MoveNextUnique(LockingInfo)</td>
<td>If the cursor is not yet initialized, MoveNextUnique is identical to MoveFirst(LockingInfo). Otherwise, move the cursor to the next non-duplicate key in the database, and store that key and associated datum in Current. MoveNextUnique will return false if no non-duplicate key/data pairs exist after the cursor position in the database. (Inherited from Cursor.)</td>
</tr>
</tbody>
</table>
See Also

BTreeCursor Class
BTreeCursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation

BTreeCursor::MoveNextUniqueMultiple Method

BTreeCursor Class  See Also
MoveNextUniqueMultiple()()

If the cursor is not yet initialized, MoveNextUniqueMultiple is identical to MoveFirstMultiple(). Otherwise, move the cursor to the next non-duplicate key in the database, and store that key and associated datum and as many duplicate data items that can fit in a buffer the size of one database page in CurrentMultiple.

MoveNextUniqueMultiple will return false if no non-duplicate key/data pairs exist after the cursor position in the database. (Inherited from Cursor.)

MoveNextUniqueMultiple(LockingInfo)

If the cursor is not yet initialized, MoveNextUniqueMultiple is identical to MoveFirstMultiple(LockingInfo). Otherwise, move the cursor to the next non-duplicate key in the database, and store that key and associated datum and as many duplicate data items that can fit in a buffer the size of one database page in CurrentMultiple.

MoveNextUniqueMultiple will return false if no non-duplicate key/data pairs exist after the cursor position in the database. (Inherited from Cursor.)
MoveNextUniqueMultiple(Int32)

MoveNextUniqueMultiple is identical to MoveFirstMultiple(Int32). Otherwise, move the cursor to the next non-duplicate key in the database, and store that key and associated datum and as many duplicate data items that can fit in a buffer the size of BufferSize in CurrentMultiple.

MoveNextUniqueMultiple will return false if no non-duplicate key/data pairs exist after the cursor position in the database. (Inherited from Cursor.)

If the cursor is not yet initialized, MoveNextUniqueMultiple is identical to MoveFirstMultiple(Int32, LockingInfo). Otherwise, move the cursor to the next non-duplicate key in the database, and store that key and associated datum and as many duplicate data items that can fit in a buffer the size of BufferSize in CurrentMultiple.

MoveNextUniqueMultiple will return false if no non-duplicate key/data pairs exist after the cursor position in the database. (Inherited from Cursor.)

MoveNextUniqueMultiple(Int32, LockingInfo)
See Also

BTreeCursor Class
BTreeCursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
BTreeCursor::MoveNextUniqueMultipleKey Method

BTreeCursor Class  See Also
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MoveNextUniqueMultipleKey()</strong></td>
<td>If the cursor is not yet initialized, MoveNextUniqueMultipleKey is identical to <strong>MoveFirstMultipleKey()</strong>. Otherwise, move the cursor to the next non-duplicate key in the database, and store that key and associated datum and as many ensuing key/data pairs that can fit in a buffer the size of one database page in <strong>CurrentMultipleKey</strong>. MoveNextUniqueMultipleKey returns false if no non-duplicate key/data pairs exist after the cursor position in the database. (Inherited from <strong>Cursor</strong>.)</td>
</tr>
<tr>
<td><strong>MoveNextUniqueMultipleKey(LockingInfo)</strong></td>
<td>If the cursor is not yet initialized, MoveNextUniqueMultipleKey is identical to <strong>MoveFirstMultipleKey(LockingInfo)</strong>. Otherwise, move the cursor to the next non-duplicate key in the database, and store that key and associated datum and as many ensuing key/data pairs that can fit in a buffer the size of one database page in <strong>CurrentMultipleKey</strong>. MoveNextUniqueMultipleKey returns false if no non-duplicate key/data pairs exist after the cursor position in the database. (Inherited from <strong>Cursor</strong>.)</td>
</tr>
</tbody>
</table>


MoveNextUniqueMultipleKey(Int32)

identical to MoveFirstMultipleKey(Int32).
Otherwise, move the cursor to the next non-duplicate key in the database, and store that key and associated datum and as many ensuing key/data pairs that can fit in a buffer the size of BufferSize in CurrentMultipleKey.

MoveNextUniqueMultipleKey will return false if no non-duplicate key/data pairs exist after the cursor position in the database.
(Inherited from Cursor.)

If the cursor is not yet initialized, MoveNextUniqueMultipleKey is identical to MoveFirstMultipleKey(Int32, LockingInfo). Otherwise, move the cursor to the next non-duplicate key in the database, and store that key and associated datum and as many ensuing key/data pairs that can fit in a buffer the size of BufferSize in CurrentMultipleKey.

MoveNextUniqueMultipleKey will return false if no non-duplicate key/data pairs exist after the cursor position in the database.
(Inherited from Cursor.)
See Also

BTreeCursor Class
BTreeCursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
BTreeCursor...::MovePrev Method

BTreeCursor Class   See Also
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MovePrev()()</td>
<td>If the cursor is not yet initialized, MovePrev is identical to MoveLast(). Otherwise, move the cursor to the previous key/data pair of the database, and store that pair in Current. In the presence of duplicate key values, the value of Current.Key may not change. (Inherited from Cursor.)</td>
</tr>
<tr>
<td>MovePrev(LockingInfo)</td>
<td>If the cursor is not yet initialized, MovePrev is identical to MoveLast(LockingInfo). Otherwise, move the cursor to the previous key/data pair of the database, and store that pair in Current. In the presence of duplicate key values, the value of Current.Key may not change. (Inherited from Cursor.)</td>
</tr>
</tbody>
</table>
See Also

BTreeCursor Class
BTreeCursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
BTreeCursor...::MovePrevDuplicate Method

See Also
<table>
<thead>
<tr>
<th><strong>Name</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>MovePrevDuplicate()()</td>
<td>If the previous key/data pair of the database is a duplicate data record for the current key/data pair, the cursor is moved to the previous key/data pair of the database, and that pair is stored in <strong>Current</strong>. MovePrevDuplicate will return false if the previous key/data pair of the database is not a duplicate data record for the current key/data pair. (Inherited from <strong>Cursor</strong>.)</td>
</tr>
<tr>
<td>MovePrevDuplicate(LockingInfo)</td>
<td>If the previous key/data pair of the database is a duplicate data record for the current key/data pair, the cursor is moved to the previous key/data pair of the database, and that pair is stored in <strong>Current</strong>. MovePrevDuplicate will return false if the previous key/data pair of the database is not a duplicate data record for the current key/data pair. (Inherited from <strong>Cursor</strong>.)</td>
</tr>
</tbody>
</table>
See Also

BTreeCursor Class
BTreeCursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
BTreeCursor Class  See Also

BTreeCursor...::MovePrevUnique Method
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MovePrevUnique()</td>
<td>If the cursor is not yet initialized, MovePrevUnique is identical to MoveLast(). Otherwise, move the cursor to the previous non-duplicate key in the database, and store that key and associated datum in Current. MovePrevUnique will return false if no non-duplicate key/data pairs exist after the cursor position in the database. (Inherited from Cursor.)</td>
</tr>
<tr>
<td>MovePrevUnique(LockingInfo)</td>
<td>If the cursor is not yet initialized, MovePrevUnique is identical to MoveLast(LockingInfo). Otherwise, move the cursor to the previous non-duplicate key in the database, and store that key and associated datum in Current. MovePrevUnique will return false if no non-duplicate key/data pairs exist after the cursor position in the database. (Inherited from Cursor.)</td>
</tr>
</tbody>
</table>
See Also

BTreeCursor Class
BTreeCursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
BTreeCursor Class  See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recno()()</td>
<td>Return the record number associated with the cursor's current position.</td>
</tr>
<tr>
<td>Recno(LockingInfo)</td>
<td>Return the record number associated with the cursor's current position.</td>
</tr>
</tbody>
</table>

See Also

BTreeCursor Class
BTreeCursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Return the record number associated with the cursor's current position.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public uint Recno()

Visual Basic (Declaration)

Public Function Recno As UInteger

Visual C++

public:
unsigned int Recno()

Return Value

The record number associated with the cursor.
See Also

BTreeCursor Class
Recno Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Return the record number associated with the cursor's current position.

**Namespace:**  BerkeleyDB  
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

```csharp
public uint Recno(
    LockingInfo info
)
```

Visual Basic (Declaration)

```vbnet
Public Function Recno (_
    info As LockingInfo _
) As UInteger
```

Visual C++

```cpp
public:
    unsigned int Recno(  
        LockingInfo^ info
    )
```

Parameters

info

Type: BerkeleyDB::LockingInfo
The locking behavior to use

Return Value

The record number associated with the cursor.
See Also

BTreeCursor Class
Recno Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
BTreeCursor Class  See Also
### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Refresh()()</td>
<td>Store the key/data pair to which the cursor refers in Current. (Inherited from Cursor.)</td>
</tr>
<tr>
<td>Refresh(LockingInfo)</td>
<td>Store the key/data pair to which the cursor refers in Current. (Inherited from Cursor.)</td>
</tr>
</tbody>
</table>
See Also

BTreeCursor Class
BTreeCursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
BTreeCursor Class
See Also

BTreeCursor RefreshMultiple Method
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RefreshMultiple()()()</td>
<td>Store the key/data pair to which the cursor refers and as many duplicate data items that can fit in a buffer the size of one database page in CurrentMultiple. (Inherited from Cursor.)</td>
</tr>
<tr>
<td>RefreshMultiple(LockingInfo)</td>
<td>Store the key/data pair to which the cursor refers and as many duplicate data items that can fit in a buffer the size of one database page in CurrentMultiple. (Inherited from Cursor.)</td>
</tr>
<tr>
<td>RefreshMultiple(Int32)</td>
<td>Store the key/data pair to which the cursor refers and as many duplicate data items that can fit in a buffer the size of BufferSize in CurrentMultiple. (Inherited from Cursor.)</td>
</tr>
<tr>
<td>RefreshMultiple(Int32, LockingInfo)</td>
<td>Store the key/data pair to which the cursor refers and as many duplicate data items that can fit in a buffer the size of BufferSize in CurrentMultiple. (Inherited from Cursor.)</td>
</tr>
</tbody>
</table>
See Also

BTreeCursor Class
BTreeCursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
BTreeCursor Class  See Also

BTreeCursor

RefreshMultipleKey Method
### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![RefreshMultipleKey()()()]</td>
<td>Store the key/data pair to which the cursor refers and as many ensuing key/data pairs that can fit in a buffer the size of one database page in CurrentMultipleKey. (Inherited from Cursor.)</td>
</tr>
<tr>
<td>![RefreshMultipleKey(LockingInfo)]</td>
<td>Store the key/data pair to which the cursor refers and as many ensuing key/data pairs that can fit in a buffer the size of one database page in CurrentMultipleKey. (Inherited from Cursor.)</td>
</tr>
<tr>
<td>![RefreshMultipleKey(Int32)]</td>
<td>Store the key/data pair to which the cursor refers and as many ensuing key/data pairs that can fit in a buffer the size of BufferSize in CurrentMultipleKey. (Inherited from Cursor.)</td>
</tr>
<tr>
<td>![RefreshMultipleKey(Int32, LockingInfo)]</td>
<td>Store the key/data pair to which the cursor refers and as many ensuing key/data pairs that can fit in a buffer the size of BufferSize in CurrentMultipleKey. (Inherited from Cursor.)</td>
</tr>
</tbody>
</table>
See Also

BTreeCursor Class
BTreeCursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `BTreeCursor` type exposes the following members.
# Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current</td>
<td>The key/data pair at which the cursor currently points. (Inherited from Cursor.)</td>
</tr>
<tr>
<td>CurrentMultiple</td>
<td>The key and multiple data items at which the cursor currently points. (Inherited from Cursor.)</td>
</tr>
<tr>
<td>CurrentMultipleKey</td>
<td>The multiple key and data items at which the cursor currently points. (Inherited from Cursor.)</td>
</tr>
<tr>
<td>Priority</td>
<td>The cache priority for pages referenced by the cursor. (Inherited from Cursor.)</td>
</tr>
</tbody>
</table>
See Also

BTreeCursor Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
A class representing a BTreeDatabase. The Btree format is a representation of a sorted, balanced tree structure.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

**C#**

```csharp
public class BTreeDatabase : Database
```

**Visual Basic (Declaration)**

```vbnet
Public Class BTreeDatabase _
    Inherits Database
```

**Visual C++**

```c++
public ref class BTreeDatabase : public Database
```
Inheritance Hierarchy

System::Object
BerkeleyDB::BaseDatabase
BerkeleyDB::Database
BerkeleyDB::BTreeDatabase
See Also

BTreeDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `BTreeDatabase` type exposes the following members.
Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overloaded.</td>
<td>Although closing a database will close any open cursors, it is recommended that applications explicitly close all their Cursor objects before closing the database. The reason why is that when the cursor is explicitly closed, the memory allocated for it is reclaimed; however, this will not happen if you close a database while cursors are still opened. The same rule, for the same reasons, hold true for Transaction objects. Simply make sure you resolve all your transaction objects before closing your database handle. Because key/data pairs are cached in memory, applications should make a point to always either close database handles or sync their data to disk (using Sync()() before exiting, to ensure that any data cached in main memory are reflected in the underlying file system. When called on a database that is the primary database for a secondary index, the primary database should be closed only after all secondary indices referencing it have been closed. When multiple threads are using the object concurrently, only a single thread may call the Close method. The object may not be accessed again after</td>
</tr>
</tbody>
</table>
Close is called, regardless of its outcome.

- **Compact** Overloaded.
- **Cursor** Overloaded.
- **Delete** Overloaded.

Release the resources held by this object, and close the database if it's still open.
(Inherited from **BaseDatabase**.)

- **Dispose**

Determines whether the specified **Object** is equal to the current **Object**.
(Inherited from **Object**.)

- **Exists** Overloaded.

Among other things, this method makes it possible for applications to request key and record counts without incurring the performance penalty of traversing the entire database.

- **FastStats**

The statistical information is described by the **BTreeStats**, **HashStats**, **QueueStats**, and **RecnoStats** classes.

- **Get** Overloaded.
- **GetBoth** Overloaded.
- **GetBothMultiple** Overloaded.

Serves as a hash function for a particular type.
(Inherited from **Object**.)

- **GetHashCode**
- **GetMultiple** Overloaded.

Gets the **Type** of the current instance.
(Inherited from **Object**.)

- **GetType**

Create a specialized join cursor for use in performing equality or natural joins on secondary indices.
(Inherited from **Database**.)

- **Join**

- **KeyRange** Overloaded.
Open Overloaded.

PrintFastStats

The statistical information is described by the BTreeStats, HashStats, QueueStats, and RecnoStats classes.

Overloaded.

PrintStats

The statistical information is described by the BTreeStats, HashStats, QueueStats, and RecnoStats classes.

Overloaded.

Put

If the database supports duplicates, add the new data value at the end of the duplicate set. If the database supports sorted duplicates, the new data value is inserted at the correct sorted location.

PutNoDuplicate Overloaded.

PutNoOverwrite Overloaded.

Stats

The statistical information is described by BTreeStats.

Sync

Flush any cached information to disk.
(Inherited from BaseDatabase.)

Returns a String that represents the current Object.
(Inherited from Object.)

Truncate

When called on a database configured with secondary indices, Truncate will truncate the primary database and all secondary indices. A count of the records discarded from the primary database is returned.

TruncateUnusedPages Overloaded.
### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AutoCommit</strong></td>
<td>If true, all database modification operations based on this object will be transactionally protected. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>CacheSize</strong></td>
<td>The size of the shared memory buffer pool -- that is, the cache. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>Compare</strong></td>
<td>The Btree key comparison function. The comparison function is called whenever it is necessary to compare a key specified by the application with a key currently stored in the tree.</td>
</tr>
<tr>
<td><strong>Compress</strong></td>
<td>The compression function used to store key/data pairs in the database.</td>
</tr>
<tr>
<td><strong>Creation</strong></td>
<td>The CreatePolicy with which this database was opened. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>DatabaseName</strong></td>
<td>The name of this database, if it has one. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>Decompress</strong></td>
<td>The decompression function used to retrieve key/data pairs from the database. If true, do checksum verification of pages read into the cache from the backing filestore. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>DoChecksum</strong></td>
<td></td>
</tr>
<tr>
<td><strong>DupCompare</strong></td>
<td>The duplicate data item comparison function.</td>
</tr>
<tr>
<td><strong>Duplicates</strong></td>
<td>Whether the insertion of duplicate data items in the database is permitted, and whether duplicates items are sorted. The algorithm used by the Berkeley DB library to perform encryption and decryption. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>EncryptAlgorithm</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Encrypted</strong></td>
<td>If true, encrypt all data stored in the database. (Inherited from BaseDatabase.)</td>
</tr>
</tbody>
</table>
**Endianness**
The database byte order.
(Inherited from [BaseDatabase](#).

**ErrorFeedback**
The mechanism for reporting detailed error messages to the application.
(Inherited from [BaseDatabase](#).

**ErrorPrefix**
The prefix string that appears before error messages issued by Berkeley DB.
(Inherited from [BaseDatabase](#).

**Feedback**
Monitor progress within long running operations.
(Inherited from [BaseDatabase](#).

**FileName**
The filename of this database, if it has one.
(Inherited from [BaseDatabase](#).

**FreeThreaded**
If true, the object is free-threaded; that is, concurrently usable by multiple threads in the address space.
(Inherited from [BaseDatabase](#).

**HasMultiple**
If true, the object references a physical file supporting multiple databases.
(Inherited from [BaseDatabase](#).

**InHostOrder**
If true, the underlying database files were created on an architecture of the same byte order as the current one. This information may be used to determine whether application data needs to be adjusted for this architecture or not.
(Inherited from [BaseDatabase](#).

**MinKeysPerPage**
The minimum number of key/data pairs intended to be stored on any single Btree leaf page.

If true, this database is not mapped into process memory.

**NoMMap**
See [MMapSize](#) for further information.

(Inherited from [BaseDatabase](#).

**NonDurableTxns**
If true, Berkeley DB will not write log records for this database.
(Inherited from [BaseDatabase](#).)
**Pagesize**
The database's current page size.
(Inherited from [BaseDatabase](#).)

The Btree prefix function. The prefix function is used to determine the amount by which keys stored on the Btree internal pages can be safely truncated without losing their uniqueness.

**PrefixCompare**
The cache priority for pages referenced by this object.
(Inherited from [BaseDatabase](#).)

If true, this database has been opened for reading only. Any attempt to modify items in the database will fail, regardless of the actual permissions of any underlying files.
(Inherited from [BaseDatabase](#).)

**Priority**
If true, this database supports transactional read operations with degree 1 isolation. Read operations on the database may request the return of modified but not yet committed data.
(Inherited from [BaseDatabase](#).)

**ReadOnly**
If true, this object supports retrieval from the Btree using record numbers.

**RecordNumbers**
If false, empty pages will not be coalesced into higher-level pages.

**ReverseSplit**
If true, this database has been opened in a transactional mode.
(Inherited from [BaseDatabase](#).)

**Transactional**
If true, the underlying file was physically truncated upon open, discarding all previous databases it might have held.
(Inherited from [BaseDatabase](#).)

**Truncated**
The type of the underlying access method (and file format). This value may be used to determine the type of the database after an `Open(String, DatabaseConfig)`.
(Inherited from [BaseDatabase](#).)

**Type**
If true, the database was opened with support for multiversion concurrency control.

**UseMVCC**
(Inherited from BaseDatabase.)
See Also

BTreeDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `BTreeDatabase` type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overloaded.</td>
<td>Although closing a database will close any open cursors, it is recommended that applications explicitly close all their Cursor objects before closing the database. The reason why is that when the cursor is explicitly closed, the memory allocated for it is reclaimed; however, this will not happen if you close a database while cursors are still opened. The same rule, for the same reasons, hold true for Transaction objects. Simply make sure you resolve all your transaction objects before closing your database handle. Because key/data pairs are cached in memory, applications should make a point to always either close database handles or sync their data to disk (using Sync()()) before exiting, to ensure that any data cached in main memory are reflected in the underlying file system. When called on a database that is the primary database for a secondary index, the primary database should be closed only after all secondary indices referencing it have been closed. When multiple threads are using the object concurrently, only a single thread may call the Close method. The object may not be accessed again after</td>
</tr>
</tbody>
</table>
Close is called, regardless of its outcome.

- **Compact**
  Overloaded.

- **Cursor**
  Overloaded.

- **Delete**
  Overloaded.
  Release the resources held by this object, and close the database if it's still open.
  (Inherited from BaseDatabase.)

- **Dispose**
  Determines whether the specified Object is equal to the current Object.
  (Inherited from Object.)

- **Equals**
  Overloaded.

- **Exists**
  Overloaded.

  Among other things, this method makes it possible for applications to request key and record counts without incurring the performance penalty of traversing the entire database.

- **FastStats**
  The statistical information is described by the BTreeStats, HashStats, QueueStats, and RecnoStats classes.

- **Get**
  Overloaded.

- **GetBoth**
  Overloaded.

- **GetBothMultiple**
  Overloaded.
  Serves as a hash function for a particular type.
  (Inherited from Object.)

- **GetHashCode**
  Overloaded.

- **GetType**
  Gets the Type of the current instance.
  (Inherited from Object.)
  Create a specialized join cursor for use in performing equality or natural joins on secondary indices.
  (Inherited from Database.)

- **KeyRange**
  Overloaded.
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>PrintFastStats</td>
<td>The statistical information is described by the <strong>BTreeStats</strong>, <strong>HashStats</strong>, <strong>QueueStats</strong>, and <strong>RecnoStats</strong> classes.</td>
</tr>
<tr>
<td>PrintStats</td>
<td>The statistical information is described by the <strong>BTreeStats</strong>, <strong>HashStats</strong>, <strong>QueueStats</strong>, and <strong>RecnoStats</strong> classes.</td>
</tr>
<tr>
<td>Put</td>
<td>If the database supports duplicates, add the new data value at the end of the duplicate set. If the database supports sorted duplicates, the new data value is inserted at the correct sorted location.</td>
</tr>
<tr>
<td>PutNoDuplicate</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>PutNoOverwrite</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>Stats</td>
<td>The statistical information is described by <strong>BTreeStats</strong>.</td>
</tr>
<tr>
<td>Sync</td>
<td>Flush any cached information to disk. (Inherited from <strong>BaseDatabase</strong>.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a <strong>String</strong> that represents the current <strong>Object</strong>. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td>Truncate</td>
<td>Overloaded. When called on a database configured with secondary indices, Truncate will truncate the primary database and all secondary indices. A count of the records discarded from the primary database is returned.</td>
</tr>
<tr>
<td>TruncateUnusedPages</td>
<td>Overloaded.</td>
</tr>
</tbody>
</table>
See Also

BTreeDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Although closing a database will close any open cursors, it is recommended that applications explicitly close all their Cursor objects before closing the database. The reason why is that when the cursor is explicitly closed, the memory allocated for it is reclaimed; however, this will not happen if you close a database while cursors are still opened.

The same rule, for the same reasons, hold true for Transaction objects. Simply make sure you resolve all your transaction objects before closing your database handle.

Because key/data pairs are cached in memory, applications should make a point to always either close database handles or sync their data to disk (using Sync() before exiting, to ensure that any data cached in main memory are reflected in the underlying file system.

When called on a database that is the primary database for a secondary index, the primary database should be closed only after all secondary indices referencing it have been closed.

When multiple threads are using the object concurrently, only a single thread may call the Close method.

The object may not be accessed again after Close is called, regardless of its outcome.
### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Close()</td>
<td>Flush any cached database information to disk, close any open Cursor() objects, free any allocated resources, and close any underlying files. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>Close(Boolean)</td>
<td>Optionally flush any cached database information to disk, close any open Cursor() objects, free any allocated resources, and close any underlying files. (Inherited from BaseDatabase.)</td>
</tr>
</tbody>
</table>
See Also

BTreeDatabase Class
BTreeDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation

BTreeDatabase::Compact Method

BTreeDatabase Class  See Also
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compact(CompactConfig)</td>
<td>Compact the database, and optionally return unused database pages to the underlying filesystem.</td>
</tr>
<tr>
<td>Compact(CompactConfig, Transaction)</td>
<td>Compact the database, and optionally return unused database pages to the underlying filesystem.</td>
</tr>
</tbody>
</table>
See Also

BTreeDatabase Class
BTreeDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Compact the database, and optionally return unused database pages to the underlying filesystem.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

```csharp
public CompactData Compact(
    CompactConfig cdata
)
```

Visual Basic (Declaration)

```vbnet
Public Function Compact (_
    cdata As CompactConfig _
) As CompactData
```

Visual C++

```cpp
public: CompactData^ Compact(
    CompactConfig^ cdata
)
```

Parameters

cdata

Type: BerkeleyDB:::CompactConfig

Compact configuration parameters

Return Value

Compact operation statistics
Remarks

If the operation occurs in a transactional database, the operation will be implicitly transaction protected using multiple transactions. These transactions will be periodically committed to avoid locking large sections of the tree. Any deadlocks encountered cause the compaction operation to be retried from the point of the last transaction commit.
See Also

BTreeDatabase Class
Compact Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Compact the database, and optionally return unused database pages to the underlying filesystem.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
## Syntax

### C#

```csharp
public CompactData Compact(
    CompactConfig cdata,
    Transaction txn
)
```

### Visual Basic (Declaration)

```vbnet
Public Function Compact ( _
    cdata As CompactConfig, _
    txn As Transaction _
) As CompactData
```

### Visual C++

```cpp
public:
CompactData^ Compact(
    CompactConfig^ cdata,
    Transaction^ txn
)
```

## Parameters

**cdata**
- Type: `BerkeleyDB::::CompactConfig`
- Compact configuration parameters

**txn**
- Type: `BerkeleyDB::::Transaction`
- If the operation is part of an application-specified transaction, txn is a Transaction object returned from `BeginTransaction()`; if the operation is part of a Berkeley DB Concurrent Data Store group, txn is a handle returned from `BeginCDSGroup()`; otherwise null.

## Return Value
Compact operation statistics
Remarks

If txn is non-null, then the operation is performed using that transaction. In this event, large sections of the tree may be locked during the course of the transaction.

If txn is null, but the operation occurs in a transactional database, the operation will be implicitly transaction protected using multiple transactions. These transactions will be periodically committed to avoid locking large sections of the tree. Any deadlocks encountered cause the compaction operation to be retried from the point of the last transaction commit.
See Also

BTreeDatabase Class
Compact Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
BTreeDatabase::Cursor Method

See Also

BTreeDatabase Class
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Cursor()()</code></td>
<td>Create a database cursor.</td>
</tr>
<tr>
<td><code>Cursor(CursorConfig)</code></td>
<td>Create a database cursor with the given configuration.</td>
</tr>
<tr>
<td><code>Cursor(Transaction)</code></td>
<td>Create a transactionally protected database cursor.</td>
</tr>
<tr>
<td><code>Cursor(CursorConfig, Transaction)</code></td>
<td>Create a transactionally protected database cursor with the given configuration.</td>
</tr>
</tbody>
</table>
See Also

BTreeDatabase Class
BTreeDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Create a database cursor.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public BTreeCursor Cursor()

Visual Basic (Declaration)

Public Function Cursor As BTreeCursor

Visual C++

public: BTreeCursor^ Cursor()

Return Value

A newly created cursor
See Also

BTreeDatabase Class
Cursor Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Create a database cursor with the given configuration.

**Namespace:**BerkeleyDB

**Assembly:**libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

```csharp
public BTreeCursor Cursor(
    CursorConfig cfg
)
```

Visual Basic (Declaration)

```vbnet
Public Function Cursor ( _
    cfg As CursorConfig _
) As BTreeCursor
```

Visual C++

```cpp
public:
BTreeCursor^ Cursor(
    CursorConfig^ cfg
)
```

Parameters

cfg

Type: BerkeleyDB::::CursorConfig
The configuration properties for the cursor.

Return Value

A newly created cursor
See Also

BTreeDatabase Class
Cursor Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Create a transactionally protected database cursor.

**Namespace:**  [BerkeleyDB](https://docs.oracle.com/cd/E11882_01/appdev.112/e10776/)

**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public BTreeCursor Cursor(
    Transaction txn
)

Visual Basic (Declaration)

Public Function Cursor ( _
    txn As Transaction _
) As BTreeCursor

Visual C++

public:
BTreeCursor^ Cursor(
    Transaction^ txn
)

Parameters

txn

Type: BerkeleyDB::Transaction
The transaction context in which the cursor may be used.

Return Value

A newly created cursor
See Also

BTreeDatabase Class
Cursor Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Create a transactionally protected database cursor with the given configuration.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

```csharp
public BTreeCursor Cursor(
    CursorConfig cfg,
    Transaction txn
)
```

Visual Basic (Declaration)

```vbnet
Public Function Cursor ( _
    cfg As CursorConfig, _
    txn As Transaction _
) As BTreeCursor
```

Visual C++

```cpp
public:
BTreeCursor^ Cursor(
    CursorConfig^ cfg,
    Transaction^ txn
)
```

Parameters

`cfg`
- Type: `BerkeleyDB::::CursorConfig`
  - The configuration properties for the cursor.

`txn`
- Type: `BerkeleyDB::::Transaction`
  - The transaction context in which the cursor may be used.

Return Value

A newly created cursor
See Also

BTreeDatabase Class
Cursor Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
BTreeDatabase Class  See Also
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delete(DatabaseEntry)</td>
<td>Remove key/data pairs from the database. The key/data pair associated with key is discarded from the database. In the presence of duplicate key values, all records associated with the designated key will be discarded. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>Delete(DatabaseEntry, Transaction)</td>
<td>Remove key/data pairs from the database. The key/data pair associated with key is discarded from the database. In the presence of duplicate key values, all records associated with the designated key will be discarded. (Inherited from BaseDatabase.)</td>
</tr>
</tbody>
</table>
See Also

BTreeDatabase Class
BTreeDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation

BTreeDatabase...::Exists Method

BTreeDatabase Class  See Also
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exists(DatabaseEntry)</td>
<td>Check whether key appears in the database. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>Exists(DatabaseEntry, Transaction)</td>
<td>Check whether key appears in the database. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>Exists(DatabaseEntry, Transaction, LockingInfo)</td>
<td>Check whether key appears in the database. (Inherited from BaseDatabase.)</td>
</tr>
</tbody>
</table>
See Also

BTreeDatabase Class
BTreeDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Among other things, this method makes it possible for applications to request key and record counts without incurring the performance penalty of traversing the entire database.

The statistical information is described by the BTreeStats, HashStats, QueueStats, and RecnoStats classes.
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FastStats()()()</td>
<td>Return the database statistical information which does not require traversal of the database.</td>
</tr>
<tr>
<td>FastStats(Transaction)</td>
<td>Return the database statistical information which does not require traversal of the database.</td>
</tr>
<tr>
<td>FastStats(Transaction, Isolation)</td>
<td>Return the database statistical information which does not require traversal of the database.</td>
</tr>
</tbody>
</table>
See Also

BTreeDatabase Class
BTreeDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Return the database statistical information which does not require traversal of the database.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public BTreeStats FastStats()

Visual Basic (Declaration)

Public Function FastStats As BTreeStats

Visual C++

public:
BTreeStats FastStats()

Return Value

The database statistical information which does not require traversal of the database.
See Also

BTreeDatabase Class
FastStats Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Return the database statistical information which does not require traversal of the database.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

```csharp
public BTreeStats FastStats(
    Transaction txn
)
```

Visual Basic (Declaration)

```vbnet
Public Function FastStats (_
    txn As Transaction _
) As BTreeStats
```

Visual C++

```cpp
public:
BTreeStats^ FastStats(
    Transaction^ txn
)
```

Parameters

txn

Type: BerkeleyDB::Transaction
If the operation is part of an application-specified transaction, txn is a
Transaction object returned from BeginTransaction(); if the operation is
part of a Berkeley DB Concurrent Data Store group, txn is a handle
returned from BeginCDSGroup(); otherwise null.

Return Value

The database statistical information which does not require traversal of the
database.
See Also

BTreeDatabase Class
FastStats Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Return the database statistical information which does not require traversal of the database.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
**Syntax**

**C#**

```csharp
public BTreeStats FastStats(
  Transaction txn,
  Isolation isoDegree
)
```

**Visual Basic (Declaration)**

```vbnet
Public Function FastStats ( _
  txn As Transaction, _
  isoDegree As Isolation _
) As BTreeStats
```

**Visual C++**

```c++
public:
BTreeStats^ FastStats(
  Transaction^ txn,
  Isolation isoDegree
)
```

**Parameters**

**txn**
Type: `BerkeleyDB::::Transaction`
If the operation is part of an application-specified transaction, `txn` is a Transaction object returned from `BeginTransaction()`; if the operation is part of a Berkeley DB Concurrent Data Store group, `txn` is a handle returned from `BeginCDSGroup()`; otherwise null.

**isoDegree**
Type: `BerkeleyDB::::Isolation`
The level of isolation for database reads. `DEGREE_ONE` will be silently ignored for databases which did not specify `ReadUncommitted`. 
**Return Value**

The database statistical information which does not require traversal of the database.
See Also

BTreeDatabase Class
FastStats Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation
BTreeDatabase...::.Get Method
BTreeDatabase Class  See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Get(DatabaseEntry)</td>
<td>Retrieve a key/data pair from the database. In the presence of duplicate key values, Get will return the first data item for key. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>Get(UInt32)</td>
<td>Retrieve a specific numbered key/data pair from the database.</td>
</tr>
<tr>
<td>Get(DatabaseEntry, Transaction)</td>
<td>Retrieve a key/data pair from the database. In the presence of duplicate key values, Get will return the first data item for key. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>Get(UInt32, Transaction)</td>
<td>Retrieve a specific numbered key/data pair from the database.</td>
</tr>
<tr>
<td>Get(DatabaseEntry, Transaction, LockingInfo)</td>
<td>Retrieve a key/data pair from the database. In the presence of duplicate key values, Get will return the first data item for key. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>Get(UInt32, Transaction, LockingInfo)</td>
<td>Retrieve a specific numbered key/data pair from the database.</td>
</tr>
</tbody>
</table>
See Also

BTreeDatabase Class
BTreeDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
BTreeDatabase.Get Method (UInt32)

BTreeDatabase Class  See Also

Retrieve a specific numbered key/data pair from the database.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

```csharp
public KeyValuePair<DatabaseEntry, DatabaseEntry> Get(
    uint recno
)
```

Visual Basic (Declaration)

```vbnet
Public Function Get ( _
    recno As UInteger _
) As KeyValuePair(Of DatabaseEntry, DatabaseEntry)
```

Visual C++

```cpp
public:
    KeyValuePair<DatabaseEntry^, DatabaseEntry^> Get(
        unsigned int recno
    )
```

Parameters

recno

Type: `System::::UInt32`

The record number of the record to be retrieved.

Return Value

A `KeyValuePair<Of <(TKey, TValue)>>` whose Key parameter is key and whose Value parameter is the retrieved data.
See Also

BTreeDatabase Class
Get Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
BTreeDatabase...::Get Method (UInt32, Transaction)

BTreeDatabase Class  See Also

Retrieve a specific numbered key/data pair from the database.

Namespace:  BerkeleyDB
Syntax

C#

```csharp
public KeyValuePair<DatabaseEntry, DatabaseEntry> Get(
    uint recno,
    Transaction txn
)
```

Visual Basic (Declaration)

```vbnet
Public Function Get (_
    recno As UInteger, _
    txn As Transaction _
) As KeyValuePair(Of DatabaseEntry, DatabaseEntry)
```

Visual C++

```cpp
public:
KeyValuePair<DatabaseEntry^, DatabaseEntry^> Get(
    unsigned int recno,
    Transaction^ txn
)
```

Parameters

recno

Type: System::::UInt32
The record number of the record to be retrieved.

txn

Type: BerkeleyDB::::Transaction
txn is a Transaction object returned from BeginTransaction(); if the operation is part of a Berkeley DB Concurrent Data Store group, txn is a handle returned from BeginCDSGroup(); otherwise null.

Return Value
A `KeyValuePair(Of (TKey, TValue)>)` whose Key parameter is key and whose Value parameter is the retrieved data.
See Also

BTreeDatabase Class
Get Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
BTreeDatabase...::Get Method (UInt32, Transaction, LockingInfo)

BTreeDatabase Class  See Also

Retrieve a specific numbered key/data pair from the database.

Namespace:  BerkeleyDB
Syntax

C#

public KeyValuePair<DatabaseEntry, DatabaseEntry> Get(
    uint recno,
    Transaction txn,
    LockingInfo info
)

Visual Basic (Declaration)

Public Function Get ( _
    recno As UInteger, _
    txn As Transaction, _
    info As LockingInfo _
) As KeyValuePair(Of DatabaseEntry, DatabaseEntry)

Visual C++

public:
    KeyValuePair<DatabaseEntry^, DatabaseEntry^> Get(
        unsigned int recno,
        Transaction^ txn,
        LockingInfo^ info
    )

Parameters

recno
Type: System::Uint32
The record number of the record to be retrieved.

txn
Type: BerkeleyDB::Transaction
txn is a Transaction object returned from BeginTransaction(); if the
operation is part of a Berkeley DB Concurrent Data Store group, txn is a
handle returned from BeginCDSGroup(); otherwise null.
info

Type: BerkeleyDB::::LockingInfo
The locking behavior to use.

Return Value

A KeyValuePair<Of <(TKey, TValue)>> whose Key parameter is key and whose Value parameter is the retrieved data.
See Also

BTreeDatabase Class
Get Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
BTreeDatabase

::

.GetBoth Method

BTreeDatabase Class  See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetBoth(DatabaseEntry, DatabaseEntry)</td>
<td>Retrieve a key/data pair from the database which matches key and data. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>GetBoth(DatabaseEntry, DatabaseEntry, Transaction)</td>
<td>Retrieve a key/data pair from the database which matches key and data. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>GetBoth(DatabaseEntry, DatabaseEntry, Transaction, LockingInfo)</td>
<td>Retrieve a key/data pair from the database which matches key and data. (Inherited from BaseDatabase.)</td>
</tr>
</tbody>
</table>
See Also

BTreeDatabase Class
BTreeDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetBothMultiple(DatabaseEntry, DatabaseEntry)</td>
<td>If a key/data pair in the database matches key and data, return the key and all duplicate data items. (Inherited from Database.)</td>
</tr>
<tr>
<td>GetBothMultiple(DatabaseEntry, DatabaseEntry, Int32)</td>
<td>If a key/data pair in the database matches key and data, return the key and all duplicate data items. (Inherited from Database.)</td>
</tr>
<tr>
<td>GetBothMultiple(DatabaseEntry, DatabaseEntry, Int32, Transaction)</td>
<td>If a key/data pair in the database matches key and data, return the key and all duplicate data items. (Inherited from Database.)</td>
</tr>
<tr>
<td>GetBothMultiple(DatabaseEntry, DatabaseEntry, Int32, Transaction, LockingInfo)</td>
<td>If a key/data pair in the database matches key and data, return the key and all duplicate data items. (Inherited from Database.)</td>
</tr>
</tbody>
</table>
See Also

BTreeDatabase Class
BTreeDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
BTreeDatabase Class

See Also

Berkeley DB .NET API Documentation
BTreeDatabase...:::GetMultiple Method
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>GetMultiple(UInt32)</code></td>
<td>Retrieve a key and all duplicate data items from the database. (Inherited from <code>Database</code>.)</td>
</tr>
<tr>
<td><code>GetMultiple(DatabaseEntry)</code></td>
<td></td>
</tr>
<tr>
<td><code>GetMultiple(UInt32, Int32)</code></td>
<td>Retrieve a key and all duplicate data items from the database. (Inherited from <code>Database</code>.)</td>
</tr>
<tr>
<td><code>GetMultiple(DatabaseEntry, Int32)</code></td>
<td></td>
</tr>
<tr>
<td><code>GetMultiple(UInt32, Int32, Transaction)</code></td>
<td>Retrieve a key and all duplicate data items from the database. (Inherited from <code>Database</code>.)</td>
</tr>
<tr>
<td><code>GetMultiple(DatabaseEntry, Int32, Transaction)</code></td>
<td></td>
</tr>
<tr>
<td><code>GetMultiple(UInt32, Int32, Transaction, LockingInfo)</code></td>
<td>Retrieve a key and all duplicate data items from the database. (Inherited from <code>Database</code>.)</td>
</tr>
<tr>
<td><code>GetMultiple(DatabaseEntry, Int32, Transaction, LockingInfo)</code></td>
<td></td>
</tr>
</tbody>
</table>
See Also

BTreeDatabase Class
BTreeDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
BTreeDatabase..::.GetMultiple Method (UInt32)

Namespace: BerkeleyDB
Syntax

C#

```csharp
public KeyValuePair<DatabaseEntry, MultipleDatabaseEntry> GetMultiple(uint recno)
```

Visual Basic (Declaration)

```vbnet
Public Function GetMultiple(_
    recno As UInteger _
) As KeyValuePair(Of DatabaseEntry, MultipleDatabaseEntry)
```

Visual C++

```cpp
public:
    KeyValuePair<DatabaseEntry^, MultipleDatabaseEntry^> GetMultiple(?
        unsigned int recno
    )
```

Parameters

recno

Type: `System::::UInt32`
See Also

BTreeDatabase Class
GetMultiple Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Namespace: BerkeleyDB
Syntax

C#

```csharp
public KeyValuePair<DatabaseEntry, MultipleDatabaseEntry> GetMultiple(
    uint recno,
    int BufferSize
)
```

Visual Basic (Declaration)

```vbnet
Public Function GetMultiple ( _
    recno As UInteger, _
    BufferSize As Integer _
) As KeyValuePair(Of DatabaseEntry, MultipleDatabaseEntry)
```

Visual C++

```cpp
public:
    KeyValuePair<DatabaseEntry^, MultipleDatabaseEntry^> GetMultiple(
        unsigned int recno,
        int BufferSize
    )
```

Parameters

recno
    Type: System::::UInt32

BufferSize
    Type: System::::Int32
See Also

BTreeDatabase Class
GetMultiple Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
BTreeDatabase...::GetMultiple Method (UInt32, Int32, Transaction)

BTreeDatabase Class  See Also

Namespace: BerkeleyDB
Syntax

C#

public KeyValuePair<DatabaseEntry, MultipleDatabaseEntry> GetMultiple(uint recno,
                int BufferSize,
                Transaction txn
            )

Visual Basic (Declaration)

Public Function GetMultiple (_
                recno As UInteger, _
                BufferSize As Integer, _
                txn As Transaction _
            ) As KeyValuePair(Of DatabaseEntry, MultipleDatabaseEntry)

Visual C++

public:
KeyValuePair<DatabaseEntry^, MultipleDatabaseEntry^> GetMultiple(
    unsigned int recno,
    int BufferSize,
    Transaction^ txn
)

Parameters

recno
    Type: System::::UInt32

BufferSize
    Type: System::::Int32

txn
    Type: BerkeleyDB::::Transaction
See Also

BTreeDatabase Class
GetMultiple Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
BTreeDatabase...::.GetMultiple Method (UInt32, Int32, Transaction, LockingInfo)

BTreeDatabase Class  See Also

Namespace:  BerkeleyDB
Syntax

C#

public KeyValuePair<DatabaseEntry, MultipleDatabaseEntry> GetMultiple(uint recno,
    int BufferSize,
    Transaction txn,
    LockingInfo info
)

Visual Basic (Declaration)

Public Function GetMultiple (_
    recno As UInteger, _
    BufferSize As Integer, _
    txn As Transaction, _
    info As LockingInfo _
) As KeyValuePair(Of DatabaseEntry, MultipleDatabaseEntry)

Visual C++

public:
    KeyValuePair<DatabaseEntry^, MultipleDatabaseEntry^> GetMultiple( _
    unsigned int recno,
    int BufferSize,
    Transaction^ txn,
    LockingInfo^ info
)

Parameters

recno
    Type: System::::UInt32

BufferSize
    Type: System::::Int32

txn
    Type: BerkeleyDB::::Transaction
info
    Type: BerkeleyDB::::LockingInfo
See Also

BTreeDatabase Class
GetMultiple Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation
BTreeDatabase...:::KeyRange Method
BTreeDatabase Class
See Also
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>KeyRange(DatabaseEntry)</td>
<td>Return an estimate of the proportion of keys that are less than, equal to, and greater than the specified key.</td>
</tr>
<tr>
<td>KeyRange(DatabaseEntry, Transaction)</td>
<td>Return an estimate of the proportion of keys that are less than, equal to, and greater than the specified key.</td>
</tr>
</tbody>
</table>
See Also

BTreeDatabase Class
BTreeDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Return an estimate of the proportion of keys that are less than, equal to, and greater than the specified key.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
**Syntax**

**C#**

```csharp
public KeyRange KeyRange(
    DatabaseEntry key
)
```

**Visual Basic (Declaration)**

```vbnet
Public Function KeyRange ( _
    key As DatabaseEntry _
) As KeyRange
```

**Visual C++**

```cpp
public: 
   KeyRange^ KeyRange(
       DatabaseEntry^ key
   )
```

**Parameters**

key

Type: BerkeleyDB::DatabaseEntry

The key to search for

**Return Value**

An estimate of the proportion of keys that are less than, equal to, and greater than the specified key.
See Also

BTreeDatabase Class
KeyRange Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Return an estimate of the proportion of keys that are less than, equal to, and greater than the specified key.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
**Syntax**

**C#**

```csharp
public KeyRange KeyRange(
    DatabaseEntry key,
    Transaction txn
)
```

**Visual Basic (Declaration)**

```vbnet
Public Function KeyRange (_
    key As DatabaseEntry, _
    txn As Transaction _
) As KeyRange
```

**Visual C++**

```cpp
public:
KeyRange^ KeyRange(
    DatabaseEntry^ key,
    Transaction^ txn
)
```

**Parameters**

**key**
- Type: `BerkeleyDB::::DatabaseEntry`
- The key to search for

**txn**
- Type: `BerkeleyDB::::Transaction`
- If the operation is part of an application-specified transaction, txn is a Transaction object returned from `BeginTransaction()();` if the operation is part of a Berkeley DB Concurrent Data Store group, txn is a handle returned from `BeginCDSGroup()();` otherwise null.

**Return Value**
An estimate of the proportion of keys that are less than, equal to, and greater than the specified key.
See Also

BTreeDatabase Class
KeyRange Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation

BTreeDatabase...::::Open Method

BTreeDatabase Class  See Also
### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Open(String, BTreeDatabaseConfig)</code></td>
<td>Instantiate a new BTreeDatabase object and open the database represented by Filename.</td>
</tr>
<tr>
<td><code>Open(String, BTreeDatabaseConfig, Transaction)</code></td>
<td>Instantiate a new BTreeDatabase object and open the database represented by Filename.</td>
</tr>
<tr>
<td><code>Open(String, String, BTreeDatabaseConfig)</code></td>
<td>Instantiate a new BTreeDatabase object and open the database represented by Filename and DatabaseName.</td>
</tr>
<tr>
<td><code>Open(String, String, BTreeDatabaseConfig, Transaction)</code></td>
<td>Instantiate a new BTreeDatabase object and open the database represented by Filename and DatabaseName.</td>
</tr>
</tbody>
</table>
See Also

BTreeDatabase Class
BTreeDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
BTreeDatabase Class

Instantiate a new BTreeDatabase object and open the database represented by Filename.

Namespace: BerkeleyDB
## Syntax

### C#

```csharp
public static BTreeDatabase Open(
    string Filename,
    BTreeDatabaseConfig cfg
)
```

### Visual Basic (Declaration)

```vbnet
Public Shared Function Open (
    _
    Filename As String,
    _
    cfg As BTreeDatabaseConfig
) As BTreeDatabase
```

### Visual C++

```cpp
public:
static BTreeDatabase^ Open(
    String^ Filename,
    BTreeDatabaseConfig^ cfg
)
```

## Parameters

### Filename

Type: `System::String`

The name of an underlying file that will be used to back the database. In-memory databases never intended to be preserved on disk may be created by setting this parameter to null.

### cfg

Type: `BerkeleyDB::BTreeDatabaseConfig`

The database's configuration

## Return Value
A new, open database object
Remarks

If Filename is null, the database is strictly temporary and cannot be opened by any other thread of control, thus the database can only be accessed by sharing the single database object that created it, in circumstances where doing so is safe.

If AutoCommit is set, the operation will be implicitly transaction protected. Note that transactionally protected operations on a database object requires the object itself be transactionally protected during its open.
See Also

BTreeDatabase Class
Open Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Instantiate a new BTreeDatabase object and open the database represented by Filename.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public static BTreeDatabase Open(
    string Filename,
    BTreeDatabaseConfig cfg,
    Transaction txn
)

Visual Basic (Declaration)

Public Shared Function Open (_
    Filename As String,
    cfg As BTreeDatabaseConfig,
    txn As Transaction_
) As BTreeDatabase

Visual C++

public:
static BTreeDatabase^ Open(
    String^ Filename,
    BTreeDatabaseConfig^ cfg,
    Transaction^ txn
)

Parameters

Filename
Type: System::String
The name of an underlying file that will be used to back the database. In-memory databases never intended to be preserved on disk may be created by setting this parameter to null.

cfg
Type: BerkeleyDB::BTreeDatabaseConfig
The database's configuration
txn

Type: BerkeleyDB::::Transaction
If the operation is part of an application-specified transaction, txn is a Transaction object returned from BeginTransaction(); if the operation is part of a Berkeley DB Concurrent Data Store group, txn is a handle returned from BeginCDSGroup(); otherwise null.

Return Value

A new, open database object
Remarks

If Filename is null, the database is strictly temporary and cannot be opened by any other thread of control, thus the database can only be accessed by sharing the single database object that created it, in circumstances where doing so is safe.

If txn is null, but AutoCommit is set, the operation will be implicitly transaction protected. Note that transactionally protected operations on a database object requires the object itself be transactionally protected during its open. Also note that the transaction must be committed before the object is closed.
See Also

BTreeDatabase Class
Open Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Instantiate a new BTreeDatabase object and open the database represented by Filename and DatabaseName.

**Namespace:**  BerkeleyDB  
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

```csharp
public static BTreeDatabase Open(
    string Filename,
    string DatabaseName,
    BTreeDatabaseConfig cfg
)
```

**Visual Basic (Declaration)**

```vbnet
Public Shared Function Open ( _
    Filename As String, _
    DatabaseName As String, _
    cfg As BTreeDatabaseConfig _
) As BTreeDatabase
```

**Visual C++**

```cpp
public:
static BTreeDatabase^ Open(
    String^ Filename, _
    String^ DatabaseName, _
    BTreeDatabaseConfig^ cfg
)
```

**Parameters**

**Filename**

Type: `System::::String`

The name of an underlying file that will be used to back the database. In-memory databases never intended to be preserved on disk may be created by setting this parameter to null.

**DatabaseName**

Type: `System::::String`

This parameter allows applications to have multiple databases in a single file. Although no DatabaseName needs to be specified, it is an error to
attempt to open a second database in a file that was not initially created using a database name.

cfg
  Type: BerkeleyDB:::BTreerDatabaseConfig
The database's configuration

Return Value

A new, open database object
Remarks

If both Filename and DatabaseName are null, the database is strictly temporary and cannot be opened by any other thread of control, thus the database can only be accessed by sharing the single database object that created it, in circumstances where doing so is safe. If Filename is null and DatabaseName is non-null, the database can be opened by other threads of control and will be replicated to client sites in any replication group.

If **AutoCommit** is set, the operation will be implicitly transaction protected. Note that transactionally protected operations on a database object requires the object itself be transactionally protected during its open.
See Also

BTreeDatabase Class
Open Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
BTreeDatabase..:::Open Method (String, String, BTreeDatabaseConfig, Transaction)

BTreeDatabase Class  See Also

Instantiate a new BTreeDatabase object and open the database represented by Filename and DatabaseName.

Namespace:  BerkeleyDB
Syntax

C#

public static BTreeDatabase Open(
    string Filename,
    string DatabaseName,
    BTreeDatabaseConfig cfg,
    Transaction txn
)

Visual Basic (Declaration)

Public Shared Function Open ( _
    Filename As String, _
    DatabaseName As String, _
    cfg As BTreeDatabaseConfig, _
    txn As Transaction _
) As BTreeDatabase

Visual C++

public:
    static BTreeDatabase^ Open(
        String^ Filename,
        String^ DatabaseName,
        BTreeDatabaseConfig^ cfg,
        Transaction^ txn
    )

Parameters

Filename
    Type: System::String
    The name of an underlying file that will be used to back the database. In-memory databases never intended to be preserved on disk may be created by setting this parameter to null.

DatabaseName
    Type: System::String
This parameter allows applications to have multiple databases in a single file. Although no DatabaseName needs to be specified, it is an error to attempt to open a second database in a file that was not initially created using a database name.

cfg
Type: BerkeleyDB::BTreeDatabaseConfig
The database's configuration

txn
Type: BerkeleyDB::Transaction
If the operation is part of an application-specified transaction, txn is a Transaction object returned from BeginTransaction(); if the operation is part of a Berkeley DB Concurrent Data Store group, txn is a handle returned from BeginCDSGroup(); otherwise null.

**Return Value**

A new, open database object
Remarks

If both Filename and DatabaseName are null, the database is strictly temporary and cannot be opened by any other thread of control, thus the database can only be accessed by sharing the single database object that created it, in circumstances where doing so is safe. If Filename is null and DatabaseName is non-null, the database can be opened by other threads of control and will be replicated to client sites in any replication group.

If txn is null, but AutoCommit is set, the operation will be implicitly transaction protected. Note that transactionally protected operations on a database object requires the object itself be transactionally protected during its open. Also note that the transaction must be committed before the object is closed.
See Also

BTreeDatabase Class
Open Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The statistical information is described by the BTreeStats, HashStats, QueueStats, and RecnoStats classes.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PrintFastStats()()()</td>
<td>Display the database statistical information which does not require traversal of the database. (Inherited from <code>BaseDatabase</code>.)</td>
</tr>
<tr>
<td>PrintFastStats(Boolean)</td>
<td>Display the database statistical information which does not require traversal of the database. (Inherited from <code>BaseDatabase</code>.)</td>
</tr>
</tbody>
</table>
See Also

BTreeDatabase Class
BTreeDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
BTreeDatabase Class  See Also

The statistical information is described by the BTreeStats, HashStats, QueueStats, and RecnoStats classes.
### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>PrintStats()</code></td>
<td>Display the database statistical information. (Inherited from <code>BaseDatabase</code>.)</td>
</tr>
<tr>
<td><code>PrintStats(Boolean)</code></td>
<td>Display the database statistical information. (Inherited from <code>BaseDatabase</code>.)</td>
</tr>
</tbody>
</table>
See Also

BTreeDatabase Class
BTreeDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If the database supports duplicates, add the new data value at the end of the duplicate set. If the database supports sorted duplicates, the new data value is inserted at the correct sorted location.
# Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Store the key/data pair in the database, replacing any previously existing key if duplicates are disallowed, or adding a duplicate data item if duplicates are allowed. (Inherited from Database.)</td>
<td>Put(DatabaseEntry, DatabaseEntry)</td>
</tr>
<tr>
<td>Store the key/data pair in the database, replacing any previously existing key if duplicates are disallowed, or adding a duplicate data item if duplicates are allowed. (Inherited from Database.)</td>
<td>Put(DatabaseEntry, DatabaseEntry, Transaction)</td>
</tr>
</tbody>
</table>
See Also

BTreeDatabase Class
BTreeDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
C#  Visual Basic
Visual C++
Include Protected Members
Include Inherited Members
Berkeley DB .NET API Documentation
BTreeDatabase:::.PutNoDuplicate Method
BTreeDatabase Class  See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PutNoDuplicate(DatabaseEntry, DatabaseEntry)</td>
<td>Store the key/data pair in the database only if it does not already appear in the database.</td>
</tr>
<tr>
<td>PutNoDuplicate(DatabaseEntry, DatabaseEntry, Transaction)</td>
<td>Store the key/data pair in the database only if it does not already appear in the database.</td>
</tr>
</tbody>
</table>
See Also

BTreeDatabase Class
BTreeDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Store the key/data pair in the database only if it does not already appear in the database.

Namespace: BerkeleyDB
Syntax

C#

public void PutNoDuplicate(
    DatabaseEntry key,
    DatabaseEntry data
)

Visual Basic (Declaration)

Public Sub PutNoDuplicate (_
    key As DatabaseEntry, _
    data As DatabaseEntry _
)

Visual C++

public:
void PutNoDuplicate(
    DatabaseEntry^ key,
    DatabaseEntry^ data
)

Parameters

key
    Type: BerkeleyDB::::DatabaseEntry
    The key to store in the database

data
    Type: BerkeleyDB::::DatabaseEntry
    The data item to store in the database
See Also

BTreeDatabase Class
PutNoDuplicate Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Store the key/data pair in the database only if it does not already appear in the database.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public void PutNoDuplicate(
    DatabaseEntry key,
    DatabaseEntry data,
    Transaction txn
)

Visual Basic (Declaration)

Public Sub PutNoDuplicate ( _
    key As DatabaseEntry, _
    data As DatabaseEntry, _
    txn As Transaction _
)

Visual C++

public:
void PutNoDuplicate(
    DatabaseEntry^ key,
    DatabaseEntry^ data,
    Transaction^ txn
)

Parameters

key
Type: BerkeleyDB::DatabaseEntry
The key to store in the database

data
Type: BerkeleyDB::DatabaseEntry
The data item to store in the database

txn
Type: BerkeleyDB::Transaction
If the operation is part of an application-specified transaction, txn is a Transaction object returned from \texttt{BeginTransaction()}; if the operation is part of a Berkeley DB Concurrent Data Store group, txn is a handle returned from \texttt{BeginCDSGroup()}; otherwise null.
See Also

BTreeDatabase Class
PutNoDuplicate Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation

BTreeDatabase:::PutNoOverwrite Method

BTreeDatabase Class  See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PutNoOverwrite</strong>(<em>DatabaseEntry</em>, <em>DatabaseEntry</em>)</td>
<td>Store the key/data pair in the database, only if the key does not already appear in the database. (Inherited from <code>Database</code>.)</td>
</tr>
<tr>
<td><strong>PutNoOverwrite</strong>(<em>DatabaseEntry</em>, <em>DatabaseEntry</em>, <em>Transaction</em>)</td>
<td>Store the key/data pair in the database, only if the key does not already appear in the database. (Inherited from <code>Database</code>.)</td>
</tr>
</tbody>
</table>
See Also

- **BTreeDatabase Class**
- **BTreeDatabase Members**
- **BerkeleyDB Namespace**

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The statistical information is described by BTreeStats.
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stats()</td>
<td>Return the database statistical information for this database.</td>
</tr>
<tr>
<td>Stats(Transaction)</td>
<td>Return the database statistical information for this database.</td>
</tr>
<tr>
<td>Stats(Transaction, Isolation)</td>
<td>Return the database statistical information for this database.</td>
</tr>
</tbody>
</table>
See Also

BTreeDatabase Class
BTreeDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation
BTreeDatabase..:::Stats Method

Return the database statistical information for this database.

Namespace: BerkeleyDB
Syntax

C#

```csharp
public BTreeStats Stats()
```

Visual Basic (Declaration)

```vbnet
Public Function Stats As BTreeStats
```

Visual C++

```cpp
public:
BTreeStats^ Stats()
```

ReturnValue

Database statistical information.
See Also

BTreeDatabase Class
Stats Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Return the database statistical information for this database.

**Namespace:**  [BerkeleyDB](https://example.com)
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
### Syntax

**C#**

```csharp
public BTreeStats Stats(
    Transaction txn
)
```

**Visual Basic (Declaration)**

```vbnet
Public Function Stats (_
    txn As Transaction _
) As BTreeStats
```

**Visual C++**

```cpp
public: BTreeStats^ Stats(
    Transaction^ txn
)
```

### Parameters

**txn**

Type: `BerkeleyDB::Transaction`
If the operation is part of an application-specified transaction, txn is a Transaction object returned from `BeginTransaction()`; if the operation is part of a Berkeley DB Concurrent Data Store group, txn is a handle returned from `BeginCDSGroup()`; otherwise null.

### Return Value

Database statistical information.
See Also

BTreeDatabase Class
Stats Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
BTreeDatabase..::.Stats Method (Transaction, Isolation)

BTreeDatabase Class  See Also

Return the database statistical information for this database.

Namespace:  BerkeleyDB
## Syntax

### C#

```csharp
public BTreeStats Stats(
    Transaction txn,
    Isolation isoDegree
)
```

### Visual Basic (Declaration)

```vbnet
Public Function Stats (_
    txn As Transaction, _
    isoDegree As Isolation _
) As BTreeStats
```

### Visual C++

```cpp
public:
BTreeStats^ Stats(
    Transaction^ txn,    
    Isolation isoDegree
)
```

## Parameters

### txn

Type: `BerkeleyDB::Transaction`

If the operation is part of an application-specified transaction, `txn` is a Transaction object returned from `BeginTransaction()`; if the operation is part of a Berkeley DB Concurrent Data Store group, `txn` is a handle returned from `BeginCDSGroup()`; otherwise null.

### isoDegree

Type: `BerkeleyDB::Isolation`

The level of isolation for database reads. `DEGREE_ONE` will be silently ignored for databases which did not specify `ReadUncommitted`.
**Return Value**

Database statistical information.
See Also

BTreeDatabase Class
Stats Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
When called on a database configured with secondary indices, Truncate will truncate the primary database and all secondary indices. A count of the records discarded from the primary database is returned.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Truncate()()</td>
<td>Empty the database, discarding all records it contains. (Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
<tr>
<td>Truncate(Transaction)</td>
<td>Empty the database, discarding all records it contains. (Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
</tbody>
</table>
See Also

BTreeDatabase Class
BTreeDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation
BTreeDatabase...:::TruncateUnusedPages Method

BTreeDatabase Class   See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TruncateUnusedPages()()()</td>
<td>Return pages to the filesystem that are already free and at the end of the file.</td>
</tr>
<tr>
<td>TruncateUnusedPages(Transaction)</td>
<td>Return pages to the filesystem that are already free and at the end of the file.</td>
</tr>
</tbody>
</table>
See Also

BTreeDatabase Class
BTreeDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
BTreeDatabase Class  See Also

Return pages to the filesystem that are already free and at the end of the file.

Namespace:  BerkeleyDB
Syntax

C#

public uint TruncateUnusedPages()

Visual Basic (Declaration)

Public Function TruncateUnusedPages As UInteger

Visual C++

public: unsigned int TruncateUnusedPages()

ReturnValue

The number of database pages returned to the filesystem
See Also

BTreeDatabase Class
TruncateUnusedPages Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Return pages to the filesystem that are already free and at the end of the file.

**Namespace:** BerkeleyDB
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public uint TruncateUnusedPages(
    Transaction txn
)

Visual Basic (Declaration)

Public Function TruncateUnusedPages ( _
    txn As Transaction _
) As UInteger

Visual C++

public:
    unsigned int TruncateUnusedPages(
        Transaction^ txn
    )

Parameters

txn

Type: BerkeleyDB::::Transaction

If the operation is part of an application-specified transaction, txn is a Transaction object returned from `BeginTransaction()`; if the operation is part of a Berkeley DB Concurrent Data Store group, txn is a handle returned from `BeginCDSGroup()`; otherwise null.

Return Value

The number of database pages returned to the filesystem
See Also

BTreeDatabase Class
TruncateUnusedPages Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation
BTreeDatabase Properties

BTreeDatabase Class  See Also

The BTreeDatabase type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AutoCommit</strong></td>
<td>If true, all database modification operations based on this object will be transactionally protected. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>CacheSize</strong></td>
<td>The size of the shared memory buffer pool -- that is, the cache. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>Compare</strong></td>
<td>The Btree key comparison function. The comparison function is called whenever it is necessary to compare a key specified by the application with a key currently stored in the tree.</td>
</tr>
<tr>
<td><strong>Compress</strong></td>
<td>The compression function used to store key/data pairs in the database.</td>
</tr>
<tr>
<td><strong>Creation</strong></td>
<td>The CreatePolicy with which this database was opened. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>DatabaseName</strong></td>
<td>The name of this database, if it has one. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>Decompress</strong></td>
<td>The decompression function used to retrieve key/data pairs from the database. If true, do checksum verification of pages read into the cache from the backing filestore. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>DoChecksum</strong></td>
<td>The algorithm used by the Berkeley DB library to perform encryption and decryption. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>DupCompare</strong></td>
<td>The duplicate data item comparison function. Whether the insertion of duplicate data items in the database is permitted, and whether duplicates items are sorted.</td>
</tr>
<tr>
<td><strong>Duplicates</strong></td>
<td>If true, encrypt all data stored in the database. (Inherited from BaseDatabase.)</td>
</tr>
</tbody>
</table>
**Endianness**
The database byte order.
(Inherited from [BaseDatabase](#).)

**ErrorFeedback**
The mechanism for reporting detailed error messages to the application.
(Inherited from [BaseDatabase](#).)

**ErrorPrefix**
The prefix string that appears before error messages issued by Berkeley DB.
(Inherited from [BaseDatabase](#).)

**Feedback**
Monitor progress within long running operations.
(Inherited from [BaseDatabase](#).)

**FileName**
The filename of this database, if it has one.
(Inherited from [BaseDatabase](#).)

**FreeThreaded**
If true, the object is free-threaded; that is, concurrently usable by multiple threads in the address space.
(Inherited from [BaseDatabase](#).)

**HasMultiple**
If true, the object references a physical file supporting multiple databases.
(Inherited from [BaseDatabase](#).)

**InHostOrder**
If true, the underlying database files were created on an architecture of the same byte order as the current one. This information may be used to determine whether application data needs to be adjusted for this architecture or not.
(Inherited from [BaseDatabase](#).)

**MinKeysPerPage**
The minimum number of key/data pairs intended to be stored on any single Btree leaf page.

**NonDurableTxns**
If true, Berkeley DB will not write log records for this database.
(Inherited from [BaseDatabase](#).)

**NoMMap**
See [MMapSize](#) for further information.

(Inherited from [BaseDatabase](#).)
| **Pagesize** | The database's current page size. (Inherited from [BaseDatabase](#).) |
| **PrefixCompare** | The Btree prefix function. The prefix function is used to determine the amount by which keys stored on the Btree internal pages can be safely truncated without losing their uniqueness. |
| **Priority** | The cache priority for pages referenced by this object. (Inherited from [BaseDatabase](#).) |
| **ReadOnly** | If true, this database has been opened for reading only. Any attempt to modify items in the database will fail, regardless of the actual permissions of any underlying files. (Inherited from [BaseDatabase](#).) |
| **ReadUncommitted** | If true, this database supports transactional read operations with degree 1 isolation. Read operations on the database may request the return of modified but not yet committed data. (Inherited from [BaseDatabase](#).) |
| **RecordNumbers** | If true, this object supports retrieval from the Btree using record numbers. |
| **ReverseSplit** | If false, empty pages will not be coalesced into higher-level pages. If true, this database has been opened in a transactional mode. (Inherited from [BaseDatabase](#).) |
| **Transactional** | If true, the underlying file was physically truncated upon open, discarding all previous databases it might have held. (Inherited from [BaseDatabase](#).) |
| **Truncated** | The type of the underlying access method (and file format). This value may be used to determine the type of the database after an [Open(String, DatabaseConfig)](##). (Inherited from [BaseDatabase](#).) |
| **Type** | If true, the database was opened with support for multiversion concurrency control. |
| **UseMVCC** | |
(Inherited from BaseDatabase.)
See Also

BTreeDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The Btree key comparison function. The comparison function is called whenever it is necessary to compare a key specified by the application with a key currently stored in the tree.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public EntryComparisonDelegate Compare { get; private set; }

Visual Basic (Declaration)

Public Property Compare As EntryComparisonDelegate

Visual C++

public:
property EntryComparisonDelegate^ Compare {
    EntryComparisonDelegate^ get ();
    void set (EntryComparisonDelegate^ value);
}
See Also

BTreeDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The compression function used to store key/data pairs in the database.

**Namespace**: BerkeleyDB

**Assembly**: libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public BTreeCompressDelegate Compress { get; private set; }

Visual Basic (Declaration)

Public Property Compress As BTreeCompressDelegate

Visual C++

public:
property BTreeCompressDelegate^ Compress {
    BTreeCompressDelegate^ get ();
    void set (BTreeCompressDelegate^ value);
}

See Also

BTreeDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The decompression function used to retrieve key/data pairs from the database.

**Namespace:**  [BerkeleyDB](#)  
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public BTreeDecompressDelegate Decompress { get; private set; }

Visual Basic (Declaration)

Public Property Decompress As BTreeDecompressDelegate

Visual C++

public:
property BTreeDecompressDelegate^ Decompress {
BTreeDecompressDelegate^ get ();
void set (BTreeDecompressDelegate^ value);
}
See Also

**BTreeDatabase Class**
**BerkeleyDB Namespace**

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The duplicate data item comparison function.

**Namespace:** [BerkeleyDB](https://www.berkeleydb.com)

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public EntryComparisonDelegate DupCompare { get; private set; }

Visual Basic (Declaration)

Public Property DupCompare As EntryComparisonDelegate

Visual C++

public:
property EntryComparisonDelegate^ DupCompare {
    EntryComparisonDelegate^ get ();
    void set (EntryComparisonDelegate^ value);
}
See Also

BTreeDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Whether the insertion of duplicate data items in the database is permitted, and whether duplicates items are sorted.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public DuplicatesPolicy Duplicates { get; }

Visual Basic (Declaration)

Public ReadOnly Property Duplicates As DuplicatesPolicy

Visual C++

public:
property DuplicatesPolicy Duplicates {
    DuplicatesPolicy get ();
}

}
See Also

BTreeDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
BTreeDatabase::MinKeysPerPage Property

BTreeDatabase Class  See Also

The minimum number of key/data pairs intended to be stored on any single Btree leaf page.

Namespace: BerkeleyDB
Syntax

C#

public uint MinKeysPerPage { get; }

Visual Basic (Declaration)

Public ReadOnly Property MinKeysPerPage AsUInteger

Visual C++

public:
property unsigned int MinKeysPerPage {
    unsigned int get ();
}

See Also

BTreeDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The Btree prefix function. The prefix function is used to determine the amount by which keys stored on the Btree internal pages can be safely truncated without losing their uniqueness.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public EntryComparisonDelegate PrefixCompare { get; private set; }

Visual Basic (Declaration)

Public Property PrefixCompare As EntryComparisonDelegate

Visual C++

public:
property EntryComparisonDelegate^ PrefixCompare {
EntryComparisonDelegate^ get ();
void set (EntryComparisonDelegate^ value);
}
See Also

BTreeDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, this object supports retrieval from the Btree using record numbers.

Namespace: BerkeleyDB
Syntax

C#

public bool RecordNumbers { get; }

Visual Basic (Declaration)

Public ReadOnly Property RecordNumbers As Boolean

Visual C++

public:
property bool RecordNumbers {
    bool get ();
}

See Also

BTreeDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If false, empty pages will not be coalesced into higher-level pages.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

```csharp
public bool ReverseSplit { get; }
```

Visual Basic (Declaration)

```vbnet
Public ReadOnly Property ReverseSplit As Boolean
```

Visual C++

```cpp
public:
property bool ReverseSplit {
    bool get ();
}
```
See Also

BTreeDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
A class representing configuration parameters for BTreeDatabase

Namespace: BerkeleyDB
Syntax

C#

public class BTreeDatabaseConfig : DatabaseConfig

Visual Basic (Declaration)

Public Class BTreeDatabaseConfig _
    Inherits DatabaseConfig

Visual C++

public ref class BTreeDatabaseConfig : public DatabaseConfig
Inheritance Hierarchy

System:::Object
BerkeleyDB:::DatabaseConfig
BerkeleyDB:::BTreeDatabaseConfig
See Also

BTreeDatabaseConfig Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `BTreeDatabaseConfig` type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BTreeDatabaseConfig</td>
<td>Create a new BTreeDatabaseConfig object</td>
</tr>
</tbody>
</table>
**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object?view=netframework-4.7.2">Object</a> is equal to the current <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object?view=netframework-4.7.2">Object</a>. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object?view=netframework-4.7.2">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object?view=netframework-4.7.2">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <a href="https://docs.microsoft.com/en-us/dotnet/api/system.type?view=netframework-4.7.2">Type</a> of the current instance. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object?view=netframework-4.7.2">Object</a>.)</td>
</tr>
<tr>
<td><strong>SetCompression</strong></td>
<td>Overloaded. Set the password and algorithm used by the Berkeley DB library to perform encryption and decryption. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.data.db2.databaseconfig?view=netframework-4.7.2">DatabaseConfig</a>.)</td>
</tr>
<tr>
<td><strong>SetEncryption</strong></td>
<td>Returns a <a href="https://docs.microsoft.com/en-us/dotnet/api/system.string?view=netframework-4.7.2">String</a> that represents the current <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object?view=netframework-4.7.2">Object</a>. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object?view=netframework-4.7.2">Object</a>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td></td>
</tr>
</tbody>
</table>
# Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AutoCommit</strong></td>
<td>Enclose the open call within a transaction. If the call succeeds, the open operation will be recoverable and all subsequent database modification operations based on this handle will be transactionally protected. If the call fails, no database will have been created. (Inherited from <a href="#">DatabaseConfig</a>).</td>
</tr>
<tr>
<td><strong>BTreeCompare</strong></td>
<td>The Btree key comparison function.</td>
</tr>
<tr>
<td><strong>BTreePrefixCompare</strong></td>
<td>The Btree prefix function.</td>
</tr>
<tr>
<td><strong>ByteOrder</strong></td>
<td>The byte order for integers in the stored database metadata. The host byte order of the machine where the Berkeley DB library was compiled is the default value. (Inherited from <a href="#">DatabaseConfig</a>).</td>
</tr>
<tr>
<td><strong>CacheSize</strong></td>
<td>The size of the shared memory buffer pool -- that is, the cache. (Inherited from <a href="#">DatabaseConfig</a>).</td>
</tr>
<tr>
<td><strong>Creation</strong></td>
<td>The policy for how to handle database creation. If true, do checksum verification of pages read into the cache from the backing filestore. (Inherited from <a href="#">DatabaseConfig</a>).</td>
</tr>
<tr>
<td><strong>DoChecksum</strong></td>
<td>The duplicate data item comparison function. Policy for duplicate data items in the database; that is, insertion when the key of the key/data pair being inserted already exists in the database will be successful.</td>
</tr>
<tr>
<td><strong>Duplicates</strong></td>
<td>The Berkeley DB environment within which to create a database. If null, the database will be created stand-alone; that is, it is not part of any Berkeley DB environment. (Inherited from <a href="#">DatabaseConfig</a>).</td>
</tr>
<tr>
<td><strong>Env</strong></td>
<td></td>
</tr>
<tr>
<td><strong>ErrorFeedback</strong></td>
<td>The mechanism for reporting error messages to the application. (Inherited from <code>DatabaseConfig</code>.)</td>
</tr>
<tr>
<td><strong>ErrorPrefix</strong></td>
<td>The prefix string that appears before error messages issued by Berkeley DB. (Inherited from <code>DatabaseConfig</code>.)</td>
</tr>
<tr>
<td><strong>Feedback</strong></td>
<td>(Inherited from <code>DatabaseConfig</code>.)</td>
</tr>
<tr>
<td><strong>FreeThreaded</strong></td>
<td>Cause the database object to be free-threaded; that is, concurrently usable by multiple threads in the address space. (Inherited from <code>DatabaseConfig</code>.)</td>
</tr>
<tr>
<td><strong>NoMMap</strong></td>
<td>Do not map this database into process memory. (Inherited from <code>DatabaseConfig</code>.)</td>
</tr>
<tr>
<td><strong>NonDurableTxns</strong></td>
<td>If true, Berkeley DB will not write log records for this database. (Inherited from <code>DatabaseConfig</code>.)</td>
</tr>
<tr>
<td><strong>NoReverseSplitting</strong></td>
<td>Turn reverse splitting in the Btree on or off.</td>
</tr>
<tr>
<td><strong>Priority</strong></td>
<td>The cache priority for pages referenced by the database. (Inherited from <code>DatabaseConfig</code>.)</td>
</tr>
<tr>
<td><strong>ReadOnly</strong></td>
<td>Open the database for reading only. Any attempt to modify items in the database will fail, regardless of the actual permissions of any underlying files. (Inherited from <code>DatabaseConfig</code>.)</td>
</tr>
<tr>
<td><strong>ReadUncommitted</strong></td>
<td>Support transactional read operations with degree 1 isolation. (Inherited from <code>DatabaseConfig</code>.)</td>
</tr>
<tr>
<td><strong>Truncate</strong></td>
<td>Physically truncate the underlying file, discarding all previous databases it might have held. (Inherited from <code>DatabaseConfig</code>.)</td>
</tr>
<tr>
<td><strong>UseMVCC</strong></td>
<td>Open the database with support for multiversion concurrency control. (Inherited from <code>DatabaseConfig</code>.)</td>
</tr>
<tr>
<td><strong>UseRecordNumbers</strong></td>
<td>If true, support retrieval from the Btree using record numbers.</td>
</tr>
</tbody>
</table>
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Compress</strong></td>
<td>The compression function used to store key/data pairs in the database.</td>
</tr>
<tr>
<td><strong>Decompress</strong></td>
<td>The decompression function used to retrieve key/data pairs from the database.</td>
</tr>
<tr>
<td><strong>EncryptAlgorithm</strong></td>
<td>The algorithm used to perform encryption and decryption. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td><strong>EncryptionPassword</strong></td>
<td>The password used to perform encryption and decryption. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td><strong>MinKeysPerPage</strong></td>
<td>The minimum number of key/data pairs intended to be stored on any single Btree leaf page.</td>
</tr>
<tr>
<td><strong>PageSize</strong></td>
<td>The size of the pages used to hold items in the database, in bytes. (Inherited from DatabaseConfig.)</td>
</tr>
</tbody>
</table>
See Also

BTreeDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Create a new BTreeDatabaseConfig object

**Namespace:**  BerkeleyDB  
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
**Syntax**

**C#**

```csharp
public BTreeDatabaseConfig()
```

**Visual Basic (Declaration)**

```vbnet
Public Sub New
```

**Visual C++**

```cpp
public:
BTreeDatabaseConfig()
```
See Also

BTreeDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `BTreeDatabaseConfig` type exposes the following members.
## Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AutoCommit</strong></td>
<td>Enclose the open call within a transaction. If the call succeeds, the open operation will be recoverable and all subsequent database modification operations based on this handle will be transactionally protected. If the call fails, no database will have been created. (Inherited from <a href="#">DatabaseConfig</a>).</td>
</tr>
<tr>
<td><strong>BTreeCompare</strong></td>
<td>The Btree key comparison function.</td>
</tr>
<tr>
<td><strong>BTreePrefixCompare</strong></td>
<td>The Btree prefix function.</td>
</tr>
<tr>
<td><strong>ByteOrder</strong></td>
<td>The byte order for integers in the stored database metadata. The host byte order of the machine where the Berkeley DB library was compiled is the default value. (Inherited from <a href="#">DatabaseConfig</a>).</td>
</tr>
<tr>
<td><strong>CacheSize</strong></td>
<td>The size of the shared memory buffer pool -- that is, the cache. (Inherited from <a href="#">DatabaseConfig</a>).</td>
</tr>
<tr>
<td><strong>Creation</strong></td>
<td>The policy for how to handle database creation. If true, do checksum verification of pages read into the cache from the backing filestore. (Inherited from <a href="#">DatabaseConfig</a>).</td>
</tr>
<tr>
<td><strong>DoChecksum</strong></td>
<td></td>
</tr>
<tr>
<td><strong>DuplicateCompare</strong></td>
<td>The duplicate data item comparison function.</td>
</tr>
<tr>
<td><strong>Duplicates</strong></td>
<td>Policy for duplicate data items in the database; that is, insertion when the key of the key/data pair being inserted already exists in the database will be successful. The Berkeley DB environment within which to create a database. If null, the database will be created stand-alone; that is, it is not part of any Berkeley DB environment. (Inherited from <a href="#">DatabaseConfig</a>).</td>
</tr>
<tr>
<td><strong>Env</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Attribute</strong></td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------</td>
</tr>
<tr>
<td><strong>ErrorFeedback</strong></td>
<td>The mechanism for reporting error messages to the application. (Inherited from <code>DatabaseConfig</code>.)</td>
</tr>
<tr>
<td><strong>ErrorPrefix</strong></td>
<td>The prefix string that appears before error messages issued by Berkeley DB. (Inherited from <code>DatabaseConfig</code>.)</td>
</tr>
<tr>
<td><strong>Feedback</strong></td>
<td>Cause the database object to be free-threaded; that is, concurrently usable by multiple threads in the address space. (Inherited from <code>DatabaseConfig</code>.)</td>
</tr>
<tr>
<td><strong>FreeThreaded</strong></td>
<td>Do not map this database into process memory. (Inherited from <code>DatabaseConfig</code>.)</td>
</tr>
<tr>
<td><strong>NoMMap</strong></td>
<td>If true, Berkeley DB will not write log records for this database. (Inherited from <code>DatabaseConfig</code>.)</td>
</tr>
<tr>
<td><strong>NonDurableTxns</strong></td>
<td>Turn reverse splitting in the Btree on or off.</td>
</tr>
<tr>
<td><strong>NoReverseSplitting</strong></td>
<td>The cache priority for pages referenced by the database. (Inherited from <code>DatabaseConfig</code>.)</td>
</tr>
<tr>
<td><strong>Priority</strong></td>
<td>Open the database for reading only. Any attempt to modify items in the database will fail, regardless of the actual permissions of any underlying files. (Inherited from <code>DatabaseConfig</code>.)</td>
</tr>
<tr>
<td><strong>ReadOnly</strong></td>
<td>Support transactional read operations with degree 1 isolation. (Inherited from <code>DatabaseConfig</code>.)</td>
</tr>
<tr>
<td><strong>ReadUncommitted</strong></td>
<td>Physically truncate the underlying file, discarding all previous databases it might have held. (Inherited from <code>DatabaseConfig</code>.)</td>
</tr>
<tr>
<td><strong>Truncate</strong></td>
<td>Open the database with support for multiversion concurrency control. (Inherited from <code>DatabaseConfig</code>.)</td>
</tr>
<tr>
<td><strong>UseMVCC</strong></td>
<td>If true, support retrieval from the Btree using record numbers.</td>
</tr>
<tr>
<td><strong>UseRecordNumbers</strong></td>
<td></td>
</tr>
</tbody>
</table>


See Also

BTreeDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation

BTreeDatabaseConfig...:::BTreeCompare Field

BTreeDatabaseConfig Class  See Also

The Btree key comparison function.

Namespace:  BerkeleyDB
Syntax

C#

public EntryComparisonDelegate BTreeCompare

Visual Basic (Declaration)

Public BTreeCompare As EntryComparisonDelegate

Visual C++

public:
EntryComparisonDelegate^ BTreeCompare
Remarks

The comparison function is called whenever it is necessary to compare a key specified by the application with a key currently stored in the tree.

If no comparison function is specified, the keys are compared lexically, with shorter keys collating before longer keys.

If the database already exists, the comparison function must be the same as that historically used to create the database or corruption can occur.
See Also

BTreeDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation

BTreeDatabaseConfig...:::BTreePrefixCompare Field

BTreeDatabaseConfig Class  See Also

The Btree prefix function.

Namespace:  BerkeleyDB
Syntax

C#

public EntryComparisonDelegate BTreePrefixCompare

Visual Basic (Declaration)

Public BTreePrefixCompare As EntryComparisonDelegate

Visual C++

public:
EntryComparisonDelegate^ BTreePrefixCompare
Remarks

The prefix function is used to determine the amount by which keys stored on the Btree internal pages can be safely truncated without losing their uniqueness. See the Btree prefix comparison section of the Berkeley DB Reference Guide for more details about how this works. The usefulness of this is data-dependent, but can produce significantly reduced tree sizes and search times in some data sets.

If no prefix function or key comparison function is specified by the application, a default lexical comparison function is used as the prefix function. If no prefix function is specified and BTreeCompare is specified, no prefix function is used. It is an error to specify a prefix function without also specifying BTreeCompare.

If the database already exists, the prefix function must be the same as that historically used to create the database or corruption can occur.
See Also

BTreeDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The policy for how to handle database creation.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

**C#**

```csharp
public CreatePolicy Creation
```

**Visual Basic (Declaration)**

```vbnet
Public Creation As CreatePolicy
```

**Visual C++**

```cpp
public:
CreatePolicy Creation
```
**Remarks**

If the database does not already exist and `NEVER` is set, `Open(String, BTreeDatabaseConfig)` will fail.
See Also

BTreeDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The duplicate data item comparison function.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public EntryComparisonDelegate DuplicateCompare

Visual Basic (Declaration)

Public DuplicateCompare As EntryComparisonDelegate

Visual C++

public:
EntryComparisonDelegate^ DuplicateCompare
Remarks

The comparison function is called whenever it is necessary to compare a data item specified by the application with a data item currently stored in the database. Setting DuplicateCompare implies setting Duplicates to SORTED.

If no comparison function is specified, the data items are compared lexically, with shorter data items collating before longer data items.

If the database already exists when Open(String, BTreeDatabaseConfig) is called, the delegate must be the same as that historically used to create the database or corruption can occur.
See Also

BTreeDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Policy for duplicate data items in the database; that is, insertion when the key of the key/data pair being inserted already exists in the database will be successful.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
### Syntax

**C#**

public `DuplicatesPolicy` Duplicates

**Visual Basic (Declaration)**

Public Duplicates As `DuplicatesPolicy`

**Visual C++**

public: `DuplicatesPolicy` Duplicates
Remarks

The ordering of duplicates in the database for **UNSORTED** is determined by the order of insertion, unless the ordering is otherwise specified by use of a cursor operation or a duplicate sort function. The ordering of duplicates in the database for **SORTED** is determined by the duplicate comparison function. If the application does not specify a comparison function using `DuplicateCompare`, a default lexical comparison will be used.

**SORTED** is preferred to **UNSORTED** for performance reasons. **UNSORTED** should only be used by applications wanting to order duplicate data items manually.

If the database already exists, the value of Duplicates must be the same as the existing database or an error will be returned.

It is an error to specify `UseRecordNumbers` and anything other than **NONE**.
See Also

BTreeDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
BTreeDatabaseConfig::NoReverseSplitting Field

Turn reverse splitting in the Btree on or off.

Namespace: BerkeleyDB
Syntax

C#

public bool NoReverseSplitting

Visual Basic (Declaration)

Public NoReverseSplitting As Boolean

Visual C++

public:

bool NoReverseSplitting
Remarks

As pages are emptied in a database, the Berkeley DB Btree implementation attempts to coalesce empty pages into higher-level pages in order to keep the database as small as possible and minimize search time. This can hurt performance in applications with cyclical data demands; that is, applications where the database grows and shrinks repeatedly. For example, because Berkeley DB does page-level locking, the maximum level of concurrency in a database of two pages is far smaller than that in a database of 100 pages, so a database that has shrunk to a minimal size can cause severe deadlocking when a new cycle of data insertion begins.
See Also

BTreeDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, support retrieval from the Btree using record numbers.

**Namespace:**  BerkeleyDB

**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
## Syntax

**C#**

```csharp
public bool UseRecordNumbers
```

**Visual Basic (Declaration)**

```vbnet
Public UseRecordNumbers As Boolean
```

**Visual C++**

```cpp
public:
    bool UseRecordNumbers
```
Remarks

Logical record numbers in Btree databases are mutable in the face of record insertion or deletion. See Renumber for further discussion.

Maintaining record counts within a Btree introduces a serious point of contention, namely the page locations where the record counts are stored. In addition, the entire database must be locked during both insertions and deletions, effectively single-threading the database for those operations. Specifying UseRecordNumbers can result in serious performance degradation for some applications and data sets.

It is an error to specify UseRecordNumbers and anything other than NONE.

If the database already exists, the value of UseRecordNumbers must be the same as the existing database or an error will be returned.
See Also

BTreeDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The **BTreeDatabaseConfig** type exposes the following members.
### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>SetCompression</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>SetEncryption</strong></td>
<td>Set the password and algorithm used by the Berkeley DB library to perform encryption and decryption. (Inherited from <a href="#">DatabaseConfig</a>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a <a href="#">String</a> that represents the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
</tbody>
</table>
See Also

BTreeDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
BTreeDatabaseConfig::SetCompression Method

See Also
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SetCompression()()()</td>
<td>Enable compression of the key/data pairs stored in the database, using the default compression and decompression functions.</td>
</tr>
<tr>
<td>SetCompression(BTreeCompressDelegate,</td>
<td>Enable compression of the key/data pairs stored in the database, using the specified compression and decompression functions.</td>
</tr>
<tr>
<td>BTreeDecompressDelegate)</td>
<td></td>
</tr>
</tbody>
</table>
See Also

BTreeDatabaseConfig Class
BTreeDatabaseConfig Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Enable compression of the key/data pairs stored in the database, using the default compression and decompression functions.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
**Syntax**

**C#**

```csharp
public void SetCompression()
```

**Visual Basic (Declaration)**

```vbnet
Public Sub SetCompression
```

**Visual C++**

```cpp
public:
void SetCompression()
```
Remarks

The default functions perform prefix compression on keys, and prefix compression on data items for duplicate keys.
See Also

BTreeDatabaseConfig Class
SetCompression Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Enable compression of the key/data pairs stored in the database, using the specified compression and decompression functions.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) **Version:** 4.8.24.0
Syntax

**C#**

```csharp
public void SetCompression(
    BTreeCompressDelegate compression,
    BTreeDecompressDelegate decompression
)
```

**Visual Basic (Declaration)**

```vbnet
Public Sub SetCompression ( _
    compression As BTreeCompressDelegate, _
    decompression As BTreeDecompressDelegate _
)
```

**Visual C++**

```cpp
public:
void SetCompression(
    BTreeCompressDelegate^ compression,
    BTreeDecompressDelegate^ decompression
)
```

**Parameters**

**compression**
- Type: `BerkeleyDB::::BTreeCompressDelegate`
- The compression function

**decompression**
- Type: `BerkeleyDB::::BTreeDecompressDelegate`
- The decompression function
See Also

BTreeDatabaseConfig Class
SetCompression Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The **BTreeDatabaseConfig** type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compress</td>
<td>The compression function used to store key/data pairs in the database.</td>
</tr>
<tr>
<td>Decompress</td>
<td>The decompression function used to retrieve key/data pairs from the database.</td>
</tr>
<tr>
<td>EncryptAlgorithm</td>
<td>The algorithm used to perform encryption and decryption.</td>
</tr>
<tr>
<td>EncryptionPassword</td>
<td>(Inherited from <a href="#">DatabaseConfig</a> )</td>
</tr>
<tr>
<td>MinKeysPerPage</td>
<td>The minimum number of key/data pairs intended to be stored on any single Btree leaf page.</td>
</tr>
<tr>
<td>PageSize</td>
<td>The size of the pages used to hold items in the database, in bytes.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">DatabaseConfig</a> )</td>
</tr>
</tbody>
</table>
See Also

BTreeDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
BTreeDatabaseConfig::Compress Property

The compression function used to store key/data pairs in the database.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public BTreeCompressDelegate Compress { get; }

Visual Basic (Declaration)

Public ReadOnly Property Compress As BTreeCompressDelegate

Visual C++

public:
property BTreeCompressDelegate^ Compress {
    BTreeCompressDelegate^ get ();
}
See Also

BTreeDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
BTreeDatabaseConfig...::Decompress Property

BTreeDatabaseConfig Class  See Also

The decompression function used to retrieve key/data pairs from the database.

Namespace:  BerkeleyDB
Syntax

C#

public BTreeDecompressDelegate Decompress { get; }

Visual Basic (Declaration)

Public ReadOnly Property Decompress As BTreeDecompressDelegate

Visual C++

public:
property BTreeDecompressDelegate^ Decompress {
    BTreeDecompressDelegate^ get ();
}

See Also

BTreeDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The minimum number of key/data pairs intended to be stored on any single Btree leaf page.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public uint MinKeysPerPage { get; set; }

Visual Basic (Declaration)

Public Property MinKeysPerPage As UInteger

Visual C++

public:
property unsigned int MinKeysPerPage {
    unsigned int get ();
    void set (unsigned int value);
}
Remarks

This value is used to determine if key or data items will be stored on overflow pages instead of Btree leaf pages. For more information on the specific algorithm used, see the Berkeley DB Reference Guide. The value specified must be at least 2; if not explicitly set, a value of 2 is used.

If the database already exists, MinKeysPerPage will be ignored.
See Also

BTreeDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
A function to decompress a key/data pair from a supplied buffer.

**Namespace:**  BerkeleyDB  
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public delegate KeyValuePair<
    DatabaseEntry, DatabaseEntry>
    BTreeDecompressDelegate(
    DatabaseEntry prevKey,
    DatabaseEntry prevData,
    byte[] compressed,
    out uint bytesRead
    )

Visual Basic (Declaration)

Public Delegate Function BTreeDecompressDelegate ( _
    prevKey As DatabaseEntry, _
    prevData As DatabaseEntry, _
    compressed As Byte(), _
    <OutAttribute> ByRef bytesRead As UInteger _
    ) As KeyValuePair(Of DatabaseEntry, DatabaseEntry)

Visual C++

public delegate KeyValuePair<
    DatabaseEntry^, DatabaseEntry^>
    BTreeDecompressDelegate(
    DatabaseEntry^ prevKey,
    DatabaseEntry^ prevData,
    array<uchar>^ compressed,
    [OutAttribute] unsigned int% bytesRead
    )

Parameters

prevKey
Type: BerkeleyDB:::DatabaseEntry
The key immediately preceding the key being decompressed.

prevData
Type: BerkeleyDB:::DatabaseEntry
The data associated with prevKey.

compressed
Type: array< System::::Byte >[]
The data stored in the tree, that is, the compressed data.

bytesRead
Type: System::::UInt32 %
The number of bytes read from compressed.

Return Value

Two new DatabaseEntry objects representing the decompressed key/data pair.
See Also

BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Statistical information about a BTreeDatabase

**Namespace:**  BerkeleyDB

**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public class BTreeStats

Visual Basic (Declaration)

Public Class BTreeStats

Visual C++

public ref class BTreeStats
Inheritance Hierarchy

System..:::Object
BerkeleyDB..:::BTreeStats
See Also

BTreeStats Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The **BTreeStats** type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified <em>Object</em> is equal to the current <em>Object</em>.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <em>Object</em>.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as a hash function for a particular type.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <em>Object</em>.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the <em>Type</em> of the current instance.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <em>Object</em>.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a <em>String</em> that represents the current <em>Object</em>.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <em>Object</em>.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-------------------------------------------------------</td>
</tr>
<tr>
<td>DuplicatePages</td>
<td>Duplicate pages.</td>
</tr>
<tr>
<td>DuplicatePagesFreeBytes</td>
<td>Bytes free in duplicate pages.</td>
</tr>
<tr>
<td>EmptyPages</td>
<td>Empty pages.</td>
</tr>
<tr>
<td>FreePages</td>
<td>Pages on the free list.</td>
</tr>
<tr>
<td>InternalPages</td>
<td>Internal pages.</td>
</tr>
<tr>
<td>InternalPagesFreeBytes</td>
<td>Bytes free in internal pages.</td>
</tr>
<tr>
<td>LeafPages</td>
<td>Leaf pages.</td>
</tr>
<tr>
<td>LeafPagesFreeBytes</td>
<td>Bytes free in leaf pages.</td>
</tr>
<tr>
<td>Levels</td>
<td>Tree levels.</td>
</tr>
<tr>
<td>MagicNumber</td>
<td>Magic number.</td>
</tr>
<tr>
<td>MetadataFlags</td>
<td>Metadata flags.</td>
</tr>
<tr>
<td>MinKey</td>
<td>Minkey value.</td>
</tr>
<tr>
<td>nData</td>
<td>Number of data items.</td>
</tr>
<tr>
<td>nKeys</td>
<td>Number of unique keys.</td>
</tr>
<tr>
<td>nPages</td>
<td>Page count.</td>
</tr>
<tr>
<td>OverflowPages</td>
<td>Overflow pages.</td>
</tr>
<tr>
<td>OverflowPagesFreeBytes</td>
<td>Bytes free in overflow pages.</td>
</tr>
<tr>
<td>PageSize</td>
<td>Page size.</td>
</tr>
<tr>
<td>Version</td>
<td>Version number.</td>
</tr>
</tbody>
</table>
See Also

BTreeStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The **BTreeStats** type exposes the following members.
### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a> is equal to the current <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <a href="https://docs.microsoft.com/en-us/dotnet/api/system.type">Type</a> of the current instance. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a <a href="https://docs.microsoft.com/en-us/dotnet/api/system.string">String</a> that represents the current <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
</tbody>
</table>
See Also

BTreeStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `BTreeStats` type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DuplicatePages</td>
<td>Duplicate pages.</td>
</tr>
<tr>
<td>DuplicatePagesFreeBytes</td>
<td>Bytes free in duplicate pages.</td>
</tr>
<tr>
<td>EmptyPages</td>
<td>Empty pages.</td>
</tr>
<tr>
<td>FreePages</td>
<td>Pages on the free list.</td>
</tr>
<tr>
<td>InternalPages</td>
<td>Internal pages.</td>
</tr>
<tr>
<td>InternalPagesFreeBytes</td>
<td>Bytes free in internal pages.</td>
</tr>
<tr>
<td>LeafPages</td>
<td>Leaf pages.</td>
</tr>
<tr>
<td>LeafPagesFreeBytes</td>
<td>Bytes free in leaf pages.</td>
</tr>
<tr>
<td>Levels</td>
<td>Tree levels.</td>
</tr>
<tr>
<td>MagicNumber</td>
<td>Magic number.</td>
</tr>
<tr>
<td>MetadataFlags</td>
<td>Metadata flags.</td>
</tr>
<tr>
<td>MinKey</td>
<td>Minkey value.</td>
</tr>
<tr>
<td>nData</td>
<td>Number of data items.</td>
</tr>
<tr>
<td>nKeys</td>
<td>Number of unique keys.</td>
</tr>
<tr>
<td>nPages</td>
<td>Page count.</td>
</tr>
<tr>
<td>OverflowPages</td>
<td>Overflow pages.</td>
</tr>
<tr>
<td>OverflowPagesFreeBytes</td>
<td>Bytes free in overflow pages.</td>
</tr>
<tr>
<td>PageSize</td>
<td>Page size.</td>
</tr>
<tr>
<td>Version</td>
<td>Version number.</td>
</tr>
</tbody>
</table>
See Also

BTreeStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Duplicate pages.

**Namespace:** BerkeleyDB
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

```csharp
public uint DuplicatePages { get; }
```

Visual Basic (Declaration)

Public ReadOnly Property DuplicatePages AsUInteger

Visual C++

```cpp
public:
property unsigned int DuplicatePages {
    unsigned int get ();
}
```
See Also

BTTreeStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Bytes free in duplicate pages.

**Namespace:**  BerkeleyDB  
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
## Syntax

### C#

```csharp
public ulong DuplicatePagesFreeBytes { get; }
```

### Visual Basic (Declaration)

```vbnet
Public ReadOnly Property DuplicatePagesFreeBytes As ULong
```

### Visual C++

```cpp
public:
    property unsigned long long DuplicatePagesFreeBytes {
        unsigned long long get ();
    }
```
See Also

BTreeStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Empty pages.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
# Syntax

**C#**

```csharp
public uint EmptyPages { get; }
```

**Visual Basic (Declaration)**

```
Public Readonly Property EmptyPages As UInteger
```

**Visual C++**

```cpp
public:
    property unsigned int EmptyPages {
        unsigned int get();
    }
```
See Also

BTreeStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
BTreeStats.FreePages Property

Pages on the free list.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

**C#**

```csharp
public uint FreePages { get; }
```

**Visual Basic (Declaration)**

```vbnet
Public ReadOnly Property FreePages As UInteger
```

**Visual C++**

```cpp
public:
    property unsigned int FreePages {
        unsigned int get ();
    }
```
See Also

BTreeStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
BTreeStats.Class

InternalPages Property

Namespace: BerkeleyDB

Internal pages.
Syntax

C#

```csharp
public uint InternalPages { get; }
```

Visual Basic (Declaration)

```vbnet
Public ReadOnly Property InternalPages AsUInteger
```

Visual C++

```cpp
public:
property unsigned int InternalPages {
    unsigned int get ();
}
```
See Also

BTreeStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Bytes free in internal pages.

Namespace: BerkeleyDB
Syntax

C#

public ulong InternalPagesFreeBytes { get; }

Visual Basic (Declaration)

Public ReadOnly Property InternalPagesFreeBytes As ULong

Visual C++

public:
property unsigned long long InternalPagesFreeBytes {
    unsigned long long get ();
}

See Also

BTreeStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Leaf pages.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public uint LeafPages { get; }

Visual Basic (Declaration)

Public ReadOnly Property LeafPages As UInteger

Visual C++

public:
property unsigned int LeafPages {
    unsigned int get ();
}
See Also

BTreeStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
BTreeStats.LeafPagesFreeBytes Property

Bytes free in leaf pages.

**Namespace:**  [BerkeleyDB](https://www.oracle.com/database/berkeley-db/javadocs/bdb-4.8.24.0/index.html)

**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

```csharp
public ulong LeafPagesFreeBytes { get; }
```

Visual Basic (Declaration)

```vbnet
Public ReadOnly Property LeafPagesFreeBytes As ULong
```

Visual C++

```cpp
public:
    property unsigned long long LeafPagesFreeBytes {
        unsigned long long get ();
    }
```
See Also

BTreeStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Tree levels.

Namespace: BerkeleyDB
Syntax

C#

public uint Levels { get; }

Visual Basic (Declaration)

Public Readonly Property Levels As UInteger

Visual C++

public:
property unsigned int Levels {
    unsigned int get ();
}


See Also

BTreeStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Magic number.

**Namespace:**  BerkeleyDB  
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public uint MagicNumber { get; }

Visual Basic (Declaration)

Public ReadOnly Property MagicNumber As UInteger

Visual C++

public:
property unsigned int MagicNumber {
    unsigned int get ();
}
See Also

BTreeStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
BTreeStats.MetadataFlags Property

Metadata flags.

Namespace: BerkeleyDB
Syntax

C#

public uint MetadataFlags { get; }

Visual Basic (Declaration)

Public ReadOnly Property MetadataFlags As UInteger

Visual C++

public:
property unsigned int MetadataFlags {
    unsigned int get ();
}

See Also

BTreeStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Minkey value.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public uint MinKey { get; }

Visual Basic (Declaration)

Public ReadOnly Property MinKey As UInteger

Visual C++

public:
property unsigned int MinKey {
    unsigned int get ();
}

See Also

BTreeStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
BTreeStats.nData Property

Number of data items.

Namespace: BerkeleyDB
Syntax

C#

public uint nData { get; }

Visual Basic (Declaration)

Public ReadOnly Property nData AsUInteger

Visual C++

public:
property unsigned int nData {
    unsigned int get ();
}

See Also

BTreeStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
BTreeStats::nKeys Property

Number of unique keys.

Namespace:  BerkeleyDB
Syntax

C#

public uint nKeys { get; }

Visual Basic (Declaration)

Public ReadOnly Property nKeys AsUInteger

Visual C++

public:
property unsigned int nKeys {
    unsigned int get ();
}
See Also

BTreeStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
BTreeStats

::

nPages Property

BTreeStats Class  See Also

Page count.

Namespace:  BerkeleyDB
Syntax

C#

public uint nPages { get; }

Visual Basic (Declaration)

Public ReadOnly Property nPages As UInteger

Visual C++

public:
property unsigned int nPages {
    unsigned int get ();
}
See Also

BTreeStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Overflow pages.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
**Syntax**

**C#**

```csharp
public uint OverflowPages { get; }
```

**Visual Basic (Declaration)**

```vbnet
Public ReadOnly Property OverflowPages AsUInteger
```

**Visual C++**

```c++
public:
property unsigned int OverflowPages {
    unsigned int get ();
}
```
See Also

BTreeStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Bytes free in overflow pages.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

```c#
public ulong OverflowPagesFreeBytes { get; }
```

Visual Basic (Declaration)

```vbnet
Public ReadOnly Property OverflowPagesFreeBytes As ULong
```

Visual C++

```cpp
public:
  property unsigned long long OverflowPagesFreeBytes {
    unsigned long long get ();
  }
```
See Also

BTreeStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
BTreeStats::PageSize Property

Page size.

Namespace: BerkeleyDB
**Syntax**

**C#**

```csharp
public uint PageSize { get; }
```

**Visual Basic (Declaration)**

```vbnet
Public ReadOnly Property PageSize As UInteger
```

**Visual C++**

```cpp
public:
property unsigned int PageSize {
    unsigned int get ();
}
```
See Also

BTreeStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
BTreeStats Class

Version number.

Namespace: BerkeleyDB
Syntax

C#

public uint Version { get; }

Visual Basic (Declaration)

Public ReadOnly Property Version AsUInteger

Visual C++

public:
property unsigned int Version {
    unsigned int get ();
}

See Also

BTreeStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
A class to represent the database byte order.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
**Syntax**

C#

```csharp
public class ByteOrder
```

**Visual Basic (Declaration)**

```vbnet
Public Class ByteOrder
```

**Visual C++**

```cpp
public ref class ByteOrder
```
Inheritance Hierarchy

System..:::Object
BerkeleyDB..:::ByteOrder
See Also

ByteOrder Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `ByteOrder` type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td>FromConst</td>
<td>Convert from the integer constant used to represent byte order in the C library to its corresponding <a href="#">ByteOrder</a> object.</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a <a href="#">String</a> that represents the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
</tbody>
</table>
### Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>BIG_ENDIAN</code></td>
<td>Big endian byte order</td>
</tr>
<tr>
<td><code>LITTLE_ENDIAN</code></td>
<td>Little endian byte order</td>
</tr>
<tr>
<td><code>MACHINE</code></td>
<td>The host byte order of the machine where the Berkeley DB library was compiled.</td>
</tr>
</tbody>
</table>
See Also

[ByteOrder Class]
[BerkeleyDB Namespace]

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The **ByteOrder** type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIG_ENDIAN</td>
<td>Big endian byte order</td>
</tr>
<tr>
<td>LITTLE_ENDIAN</td>
<td>Little endian byte order</td>
</tr>
</tbody>
</table>
| MACHINE        | The host byte order of the machine where the Berkeley DB library was compiled.
See Also

ByteOrder Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Big endian byte order

**Namespace**: BerkeleyDB
**Assembly**: libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public static ByteOrder BIG_ENDIAN

Visual Basic (Declaration)

Public Shared BIG_ENDIAN As ByteOrder

Visual C++

public:
static ByteOrder^ BIG_ENDIAN
See Also

ByteOrder Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Little endian byte order

**Namespace:**  BerkeleyDB  
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll)  
**Version:**  4.8.24.0
Syntax

C#

public static ByteOrder LITTLE_ENDIAN

Visual Basic (Declaration)

Public Shared LITTLE_ENDIAN As ByteOrder

Visual C++

public:
static ByteOrder^ LITTLE_ENDIAN
See Also

ByteOrder Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The host byte order of the machine where the Berkeley DB library was compiled.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
## Syntax

### C#

public static ByteOrder MACHINE

### Visual Basic (Declaration)

Public Shared MACHINE As ByteOrder

### Visual C++

public:
static ByteOrder^ MACHINE
See Also

ByteOrder Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The **ByteOrder** type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a> is equal to the current <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td><strong>FromConst</strong></td>
<td>Convert from the integer constant used to represent byte order in the C library to its corresponding <a href="https://docs.microsoft.com/en-us/dotnet/api/system.byteorder">ByteOrder</a> object.</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <a href="https://docs.microsoft.com/en-us/dotnet/api/system.type">Type</a> of the current instance. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a <a href="https://docs.microsoft.com/en-us/dotnet/api/system.string">String</a> that represents the current <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
</tbody>
</table>
See Also

[ByteOrder Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Convert from the integer constant used to represent byte order in the C library to its corresponding ByteOrder object.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

```csharp
public static ByteOrder FromConst(
    int order
)
```

Visual Basic (Declaration)

```vbnet
Public Shared Function FromConst ( _
    order As Integer _
) As ByteOrder
```

Visual C++

```cpp
public:
static ByteOrder^ FromConst(
    int order
)
```

Parameters

order
  Type: `System::::Int32`
  The C library constant

Return Value

The ByteOrder object corresponding to the given constant
See Also

ByteOrder Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
A class to represent information about the Berkeley DB cache

Namespace: BerkeleyDB
Syntax

**C#**

```csharp
public class CacheInfo
```

**Visual Basic (Declaration)**

```vbnet
Public Class CacheInfo
```

**Visual C++**

```cpp
public ref class CacheInfo
```
Inheritance Hierarchy

System..:::Object
BerkeleyDB..:::CacheInfo
See Also

CacheInfo Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `CacheInfo` type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CacheInfo</td>
<td>Create a new CacheInfo object. The size of the cache is set to gbytes gigabytes plus bytes and spread over numCaches separate caches.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified Object is equal to the current Object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as a hash function for a particular type.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a String that represents the current Object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
</tbody>
</table>
### Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bytes</td>
<td>The number of bytes in the cache</td>
</tr>
<tr>
<td>Gigabytes</td>
<td>The number of gigabytes in the cache</td>
</tr>
<tr>
<td>NCaches</td>
<td>The number of caches</td>
</tr>
</tbody>
</table>
See Also

CacheInfo Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Create a new CacheInfo object. The size of the cache is set to gbytes gigabytes plus bytes and spread over numCaches separate caches.

**Namespace:** BerkeleyDB
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public CacheInfo(
    uint gbytes,
    uint bytes,
    int numCaches
)

Visual Basic (Declaration)

Public Sub New ( _
    gbytes As UInteger, _
    bytes As UInteger, _
    numCaches As Integer _
)

Visual C++

public:
CacheInfo(
    unsigned int gbytes,
    unsigned int bytes,
    int numCaches
)

Parameters

gbytes
    Type: System::::UInt32
    The number of gigabytes in the cache

bytes
    Type: System::::UInt32
    The number of bytes in the cache

numCaches
    Type: System::::Int32
The number of caches
See Also

CacheInfo Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation

CacheInfo Fields

The CacheInfo type exposes the following members.
### Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bytes</td>
<td>The number of bytes in the cache</td>
</tr>
<tr>
<td>Gigabytes</td>
<td>The number of gigabytes in the cache</td>
</tr>
<tr>
<td>NCaches</td>
<td>The number of caches</td>
</tr>
</tbody>
</table>
See Also

CacheInfo Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
CacheInfo...::Bytes Field

CacheInfo Class  See Also

The number of bytes in the cache

Namespace:  BerkeleyDB
Syntax

C#

public uint Bytes

Visual Basic (Declaration)

Public Bytes As UInteger

Visual C++

public:
unsigned int Bytes
See Also

CacheInfo Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
CacheInfo Class  See Also

The number of gigabytes in the cache

Namespace:  BerkeleyDB
Syntax

C#

public uint Gigabytes

Visual Basic (Declaration)

Public Gigabytes AsUInteger

Visual C++

public:
unsigned int Gigabytes
See Also

CacheInfo Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The number of caches

Namespace: BerkeleyDB
Syntax

C#

public int NCaches

Visual Basic (Declaration)

Public NCaches As Integer

Visual C++

public:
    int NCaches
See Also

CacheInfo Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `CacheInfo` type exposes the following members.
**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a <a href="#">String</a> that represents the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
</tbody>
</table>
See Also

CacheInfo Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
A class to represent cache priority for database pages

Namespace: BerkeleyDB
**Syntax**

**C#**

```csharp
public class CachePriority
```

**Visual Basic (Declaration)**

```vbnet
Public Class CachePriority
```

**Visual C++**

```cpp
public ref class CachePriority
```
Inheritance Hierarchy

System:::Object
BerkeleyDB:::CachePriority
See Also

CachePriority Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `CachePriority` type exposes the following members.
# Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a <a href="#">String</a> that represents the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
</tbody>
</table>
## Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEFAULT</td>
<td>The default priority.</td>
</tr>
<tr>
<td>HIGH</td>
<td>The next highest priority.</td>
</tr>
<tr>
<td>LOW</td>
<td>The next lowest priority.</td>
</tr>
<tr>
<td>VERY_HIGH</td>
<td>The highest priority: pages are the least likely to be discarded.</td>
</tr>
<tr>
<td>VERY_LOW</td>
<td>The lowest priority: pages are the most likely to be discarded.</td>
</tr>
</tbody>
</table>
See Also

CachePriority Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `CachePriority` type exposes the following members.
## Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEFAULT</td>
<td>The default priority.</td>
</tr>
<tr>
<td>HIGH</td>
<td>The next highest priority.</td>
</tr>
<tr>
<td>LOW</td>
<td>The next lowest priority.</td>
</tr>
<tr>
<td>VERY_HIGH</td>
<td>The highest priority: pages are the least likely to be discarded.</td>
</tr>
<tr>
<td>VERY_LOW</td>
<td>The lowest priority: pages are the most likely to be discarded.</td>
</tr>
</tbody>
</table>
See Also

CachePriority Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
CachePriority..::..DEFAULT Field

See Also

The default priority.

Namespace:  BerkeleyDB
Syntax

C#

public static CachePriority DEFAULT

Visual Basic (Declaration)

Public Shared DEFAULT As CachePriority

Visual C++

public:
static CachePriority^ DEFAULT
See Also

CachePriority Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The next highest priority.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public static CachePriority HIGH

Visual Basic (Declaration)

Public Shared HIGH As CachePriority

Visual C++

public:
static CachePriority^ HIGH
See Also

CachePriority Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
CachePriority...:::LOW Field

**CachePriority Class**  See Also

The next lowest priority.

**Namespace:**  BerkeleyDB  
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
C#  
public static CachePriority LOW

Visual Basic (Declaration)  
Public Shared LOW As CachePriority

Visual C++  
public:  
static CachePriority^ LOW
See Also

CachePriority Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
CachePriority.

VERY_HIGH Field

CachePriority Class  See Also

The highest priority: pages are the least likely to be discarded.

Namespace:  BerkeleyDB
Syntax

C#

```csharp
public static CachePriority VERY_HIGH
```

Visual Basic (Declaration)

```vbnet
Public Shared VERY_HIGH As CachePriority
```

Visual C++

```cpp
public:
static CachePriority^ VERY_HIGH
```
See Also

CachePriority Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The lowest priority: pages are the most likely to be discarded.

Namespace:  BerkeleyDB
Syntax

C#

public static CachePriority VERY_LOW

Visual Basic (Declaration)

Public Shared VERY_LOW As CachePriority

Visual C++

public:
static CachePriority^ VERY_LOW
See Also

CachePriority Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `CachePriority` type exposes the following members.
<table>
<thead>
<tr>
<th><strong>Name</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified Object is equal to the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a String that represents the current Object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>
See Also

CachePriority Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
A class to represent configuration settings for Compact(CompactConfig) and Compact(CompactConfig).

Namespace: BerkeleyDB
**Syntax**

**C#**

public class CompactConfig

**Visual Basic (Declaration)**

Public Class CompactConfig

**Visual C++**

public ref class CompactConfig
Inheritance Hierarchy

System:::Object
BerkeleyDB:::CompactConfig
See Also

CompactConfig Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
C# Visual Basic
Visual C++
Include Protected Members
Include Inherited Members
Berkeley DB .NET API Documentation
CompactConfig Members

CompactConfig Class
Constructors
Methods
Fields
Properties
See Also

The CompactConfig type exposes the following members.
# Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CompactConfig</td>
<td>Create a new CompactConfig object</td>
</tr>
</tbody>
</table>
# Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a>.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <a href="#">Type</a> of the current instance.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a <a href="#">String</a> that represents the current <a href="#">Object</a>.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>returnEnd</td>
<td>Return the database key marking the end of the compaction operation in a Btree or Recno database. This is generally the first key of the page where the operation stopped.</td>
</tr>
<tr>
<td>start</td>
<td>If non-null, the starting point for compaction. Compaction will start at the smallest key greater than or equal to start. If null, compaction will start at the beginning of the database.</td>
</tr>
<tr>
<td>stop</td>
<td>If non-null, the stopping point for compaction. Compaction will stop at the page with the smallest key greater than stop. If null, compaction will stop at the end of the database.</td>
</tr>
<tr>
<td>TruncatePages</td>
<td>If true, return pages to the filesystem when possible. If false, pages emptied as a result of compaction will be placed on the free list for re-use, but never returned to the filesystem.</td>
</tr>
</tbody>
</table>
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FillPercentage</td>
<td>If non-zero, this provides the goal for filling pages, specified as a percentage between 1 and 100. Any page not at or above this percentage full will be considered for compaction. The default behavior is to consider every page for compaction, regardless of its page fill percentage.</td>
</tr>
<tr>
<td>Pages</td>
<td>If non-zero, compaction will complete after the specified number of pages have been freed. If non-zero, and no Transaction is specified, this parameter identifies the lock timeout used for implicit transactions, in microseconds.</td>
</tr>
</tbody>
</table>
See Also

CompactConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Create a new CompactConfig object

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

**C#**

public CompactConfig()

**Visual Basic (Declaration)**

Public Sub New

**Visual C++**

public:
CompactConfig()
See Also

CompactConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
C#  Visual Basic
Visual C++
Include Protected Members
Include Inherited Members
Berkeley DB .NET API Documentation
CompactConfig Fields

CompactConfig Class  See Also

The CompactConfig type exposes the following members.
### Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>returnEnd</td>
<td>Return the database key marking the end of the compaction operation in a Btree or Recno database. This is generally the first key of the page where the operation stopped.</td>
</tr>
<tr>
<td>start</td>
<td>If non-null, the starting point for compaction. Compaction will start at the smallest key greater than or equal to start. If null, compaction will start at the beginning of the database.</td>
</tr>
<tr>
<td>stop</td>
<td>If non-null, the stopping point for compaction. Compaction will stop at the page with the smallest key greater than stop. If null, compaction will stop at the end of the database.</td>
</tr>
<tr>
<td>TruncatePages</td>
<td>If true, return pages to the filesystem when possible. If false, pages emptied as a result of compaction will be placed on the free list for re-use, but never returned to the filesystem.</td>
</tr>
</tbody>
</table>
See Also

CompactConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
CompactConfig..:::returnEnd Field

CompactConfig Class  See Also

Return the database key marking the end of the compaction operation in a Btree or Recno database. This is generally the first key of the page where the operation stopped.

Namespace:  BerkeleyDB
Syntax

C#

public bool returnEnd

Visual Basic (Declaration)

Public returnEnd As Boolean

Visual C++

public:
    bool returnEnd
See Also

CompactConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If non-null, the starting point for compaction. Compaction will start at the smallest key greater than or equal to `start`. If null, compaction will start at the beginning of the database.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public DatabaseEntry start

Visual Basic (Declaration)

Public start As DatabaseEntry

Visual C++

public: DatabaseEntry^ start
See Also

CompactConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
CompactConfig::stop Field

If non-null, the stopping point for compaction. Compaction will stop at the page with the smallest key greater than stop. If null, compaction will stop at the end of the database.

Namespace: BerkeleyDB
Syntax

C#
public DatabaseEntry stop

Visual Basic (Declaration)
Public stop As DatabaseEntry

Visual C++
public:
DatabaseEntry^ stop
See Also

CompactConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, return pages to the filesystem when possible. If false, pages emptied as a result of compaction will be placed on the free list for re-use, but never returned to the filesystem.

Namespace: BerkeleyDB
Syntax

C#

public bool TruncatePages

Visual Basic (Declaration)

Public TruncatePages As Boolean

Visual C++

public:

bool TruncatePages
Remarks

Note that only pages at the end of a file can be returned to the filesystem. Because of the one-pass nature of the compaction algorithm, any unemptied page near the end of the file inhibits returning pages to the file system. A repeated call to `Compact(CompactConfig)` or `Compact(CompactConfig)` with a low `FillPercentage` may be used to return pages in this case.
See Also

CompactConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `CompactConfig` type exposes the following members.
# Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a <a href="#">String</a> that represents the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
</tbody>
</table>
See Also

CompactConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `CompactConfig` type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FillPercentage</td>
<td>If non-zero, this provides the goal for filling pages, specified as a percentage between 1 and 100. Any page not at or above this percentage full will be considered for compaction. The default behavior is to consider every page for compaction, regardless of its page fill percentage.</td>
</tr>
<tr>
<td>Pages</td>
<td>If non-zero, compaction will complete after the specified number of pages have been freed.</td>
</tr>
<tr>
<td>Transaction</td>
<td>If non-zero, and no <a href="#">Transaction</a> is specified, this parameter identifies the lock timeout used for implicit transactions, in microseconds.</td>
</tr>
</tbody>
</table>
See Also

CompactConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If non-zero, this provides the goal for filling pages, specified as a percentage between 1 and 100. Any page not at or above this percentage full will be considered for compaction. The default behavior is to consider every page for compaction, regardless of its page fill percentage.
Syntax

C#

public uint FillPercentage { get; set; }

Visual Basic (Declaration)

Public Property FillPercentage AsUInteger

Visual C++

public:
property unsigned int FillPercentage {
    unsigned int get ();
    void set (unsigned int value);
}
See Also

CompactConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If non-zero, compaction will complete after the specified number of pages have been freed.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

```csharp
public uint Pages { get; set; }
```

Visual Basic (Declaration)

```vbnet
Public Property Pages As UInteger
```

Visual C++

```cpp
public:
property unsigned int Pages {
    unsigned int get ();
    void set (unsigned int value);
}
```
See Also

CompactConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If non-zero, and no Transaction is specified, this parameter identifies the lock timeout used for implicit transactions, in microseconds.

Namespace: BerkeleyDB
Syntax

C#

public uint Timeout { get; set; }

Visual Basic (Declaration)

Public Property Timeout As UInteger

Visual C++

public:
property unsigned int Timeout {
    unsigned int get ();
    void set (unsigned int value);
}
See Also

CompactConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
A class for representing compact operation statistics

**Namespace:**  [BerkeleyDB](https://www berkleydb.com)

**Assembly:**  `libdb_dotnet48` (in `libdb_dotnet48.dll`) Version: 4.8.24.0
Syntax

C#

public class CompactData

Visual Basic (Declaration)

Public Class CompactData

Visual C++

public ref class CompactData
Inheritance Hierarchy

System:::Object
BerkeleyDB:::CompactData
See Also

CompactData Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
CompactData Members

The `CompactData` type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a <a href="#">String</a> that represents the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Deadlocks</td>
<td>If no Transaction parameter was specified, the number of deadlocks which occurred.</td>
</tr>
<tr>
<td>End</td>
<td>The database key marking the end of the compaction operation. This is generally the first key of the page where the operation stopped and is only non-null if returnEnd was true.</td>
</tr>
<tr>
<td>Levels</td>
<td>The number of levels removed from the Btree or Recno database during the compaction phase.</td>
</tr>
<tr>
<td>PagesExamined</td>
<td>The number of database pages reviewed during the compaction phase.</td>
</tr>
<tr>
<td>PagesFreed</td>
<td>The number of database pages freed during the compaction phase.</td>
</tr>
<tr>
<td>PagesTruncated</td>
<td>The number of database pages returned to the filesystem.</td>
</tr>
</tbody>
</table>
See Also

CompactData Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The **CompactData** type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="https://www.example.com/Object">Object</a> is equal to the current <a href="https://www.example.com/Object">Object</a>. (Inherited from <a href="https://www.example.com/Object">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="https://www.example.com/Object">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <a href="https://www.example.com/Type">Type</a> of the current instance. (Inherited from <a href="https://www.example.com/Object">Object</a>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a <a href="https://www.example.com/String">String</a> that represents the current <a href="https://www.example.com/Object">Object</a>. (Inherited from <a href="https://www.example.com/Object">Object</a>.)</td>
</tr>
</tbody>
</table>
See Also

CompactData Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The CompactData type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Deadlocks</strong></td>
<td>If no Transaction parameter was specified, the number of deadlocks which occurred.</td>
</tr>
<tr>
<td><strong>End</strong></td>
<td>The database key marking the end of the compaction operation. This is generally the first key of the page where the operation stopped and is only non-null if returnEnd was true.</td>
</tr>
<tr>
<td><strong>Levels</strong></td>
<td>The number of levels removed from the Btree or Recno database during the compaction phase.</td>
</tr>
<tr>
<td><strong>PagesExamined</strong></td>
<td>The number of database pages reviewed during the compaction phase.</td>
</tr>
<tr>
<td><strong>PagesFreed</strong></td>
<td>The number of database pages freed during the compaction phase.</td>
</tr>
<tr>
<td><strong>PagesTruncated</strong></td>
<td>The number of database pages returned to the filesystem.</td>
</tr>
</tbody>
</table>
See Also

CompactData Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If no Transaction parameter was specified, the number of deadlocks which occurred.

Namespace: BerkeleyDB
Syntax

C#

public uint Deadlocks { get; }

Visual Basic (Declaration)

Public ReadOnly Property Deadlocks AsUInteger

Visual C++

public:
property unsigned int Deadlocks {
    unsigned int get ();
}

See Also

CompactData Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
CompactData::End Property

The database key marking the end of the compaction operation. This is generally the first key of the page where the operation stopped and is only non-null if returnEnd was true.

Namespace: BerkeleyDB
Syntax

C#

public DatabaseEntry End { get; }

Visual Basic (Declaration)

Public ReadOnly Property End As DatabaseEntry

Visual C++

public:
property DatabaseEntry^ End { DatabaseEntry^ get (); }
See Also

CompactData Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The number of levels removed from the Btree or Recno database during the compaction phase.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public uint Levels { get; }

Visual Basic (Declaration)

Public ReadOnly Property Levels As UInteger

Visual C++

public:
property unsigned int Levels {
        unsigned int get();
}

See Also

CompactData Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The number of database pages reviewed during the compaction phase.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public uint PagesExamined { get; }

Visual Basic (Declaration)

Public ReadOnly Property PagesExamined As UInteger

Visual C++

public:
property unsigned int PagesExamined {
    unsigned int get ();
}

See Also

CompactData Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The number of database pages freed during the compaction phase.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public uint PagesFreed { get; }

Visual Basic (Declaration)

Public ReadOnly Property PagesFreed AsUInteger

Visual C++

public:
property unsigned int PagesFreed {
    unsigned int get ();
}

See Also

CompactData Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
CompactData.PagesTruncated Property

The number of database pages returned to the filesystem.

Namespace: BerkeleyDB
Syntax

C#

public uint PagesTruncated { get; }

Visual Basic (Declaration)

Public ReadOnly Property PagesTruncated As UInteger

Visual C++

public:
property unsigned int PagesTruncated {
    unsigned int get ();
}
See Also

CompactData Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
See Also

The policy for how to handle database creation.

Namespace: BerkeleyDB
Syntax

C#

public enum CreatePolicy

Visual Basic (Declaration)

Public Enumeration CreatePolicy

Visual C++

public enum class CreatePolicy
<table>
<thead>
<tr>
<th>Member name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NEVER</td>
<td>Never create the database.</td>
</tr>
<tr>
<td>IF_NEEDED</td>
<td>Create the database if it does not already exist.</td>
</tr>
<tr>
<td>ALWAYS</td>
<td>Do not open the database and return an error if it already exists.</td>
</tr>
</tbody>
</table>
See Also

BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
A class representing database cursors, which allow for traversal of database records.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public class Cursor : BaseCursor, IDisposable,
    IEnumerable<KeyValuePair<DatabaseEntry, DatabaseEntry>>, IEnumerable

Visual Basic (Declaration)

Public Class Cursor
    Inherits BaseCursor
    Implements IDisposable, IEnumerable(Of KeyValuePair(Of DatabaseEntry), DatabaseEntry)

Visual C++

public ref class Cursor : public BaseCursor,
    IDisposable, IEnumerable<KeyValuePair<DatabaseEntry>, DatabaseEntry}
Inheritance Hierarchy

System..::.Object
BerkeleyDB..::.BaseCursor
BerkeleyDB...::.Cursor
   BerkeleyDB..::.BTreeCursor
   BerkeleyDB..::.HashCursor
   BerkeleyDB..::.RecnoCursor
See Also

Cursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The **Cursor** type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Add</strong></td>
<td>Stores the key/data pair in the database.</td>
</tr>
<tr>
<td><strong>Close</strong></td>
<td>Discard the cursor. It is possible for the Close() method to throw a <a href="https://example.com">DeadlockException</a>, signaling that any enclosing transaction should be aborted. If the application is already intending to abort the transaction, this error should be ignored, and the application should proceed. After Close has been called, regardless of its result, the object may not be used again. (Inherited from BaseCursor.)</td>
</tr>
<tr>
<td><strong>Compare</strong></td>
<td>Compare this cursor's position to another's. (Inherited from BaseCursor.)</td>
</tr>
<tr>
<td><strong>Count</strong></td>
<td>Returns a count of the number of data items for the key to which the cursor refers. (Inherited from BaseCursor.)</td>
</tr>
<tr>
<td><strong>Delete</strong></td>
<td>Delete the key/data pair to which the cursor refers. Release the resources held by this object, and close the cursor if it's still open. (Inherited from BaseCursor.)</td>
</tr>
<tr>
<td><strong>Dispose</strong></td>
<td></td>
</tr>
</tbody>
</table>


Duplicate
Create a new cursor that uses the same transaction and locker ID as the original cursor.

Equals
Determines whether the specified Object is equal to the current Object. (Inherited from Object.)

GetEnumerator
Returns an enumerator that iterates through the Cursor.

GetHashCode
Serves as a hash function for a particular type. (Inherited from Object.)

GetType
Gets the Type of the current instance. (Inherited from Object.)

Move
Overloaded.

MoveFirst
Overloaded.

MoveMultiple
Overloaded.

If positioning the cursor fails, CurrentMultiple will contain an empty KeyValuePair(Of (Of TKey, TValue>).

MoveFirstMultiple
Overloaded.

MoveFirstMultipleKey
Overloaded.

MoveLast
Overloaded.

MoveMultiple
Overloaded.

MoveMultipleKey
Overloaded.

MoveNext
Overloaded.

MoveNextDuplicate
Overloaded.

MoveNextDuplicateMultiple
Overloaded.

MoveNextDuplicateMultipleKey
Overloaded.

MoveNextMultiple
Overloaded.

MoveNextMultipleKey
Overloaded.

MoveNextUnique
Overloaded.

MoveNextUniqueMultiple
Overloaded.

MoveNextUniqueMultipleKey
Overloaded.

MovePrev
Overloaded.

MovePrevDuplicate
Overloaded.
- **MovePrevUnique**
  Overloaded.
  Overwrite the data of the key/data pair to which the cursor refers with the specified data item.

- **Overwrite**

- **Refresh**
  Overloaded.

- **RefreshMultiple**
  Overloaded.

- **RefreshMultipleKey**
  Overloaded.

- **ToString**
  Returns a **String** that represents the current **Object**.
  (Inherited from **Object**.)
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current</td>
<td>The key/data pair at which the cursor currently points.</td>
</tr>
<tr>
<td>CurrentMultiple</td>
<td>The key and multiple data items at which the cursor currently points.</td>
</tr>
<tr>
<td>CurrentMultipleKey</td>
<td>The multiple key and data items at which the cursor currently points.</td>
</tr>
<tr>
<td>Priority</td>
<td>The cache priority for pages referenced by the cursor.</td>
</tr>
</tbody>
</table>
See Also

Cursor Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `Cursor` type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add</td>
<td>Stores the key/data pair in the database.</td>
</tr>
</tbody>
</table>
| Close | Discard the cursor.  
It is possible for the Close() method to throw a `DeadlockException`, signaling that any enclosing transaction should be aborted. If the application is already intending to abort the transaction, this error should be ignored, and the application should proceed.  
After Close has been called, regardless of its result, the object may not be used again.  
(Inherited from `BaseCursor`.)|
| Compare | Compare this cursor's position to another's.  
(Inherited from `BaseCursor`). |
| Count | Returns a count of the number of data items for the key to which the cursor refers.  
(Inherited from `BaseCursor`). |
| Delete | Delete the key/data pair to which the cursor refers.  
Release the resources held by this object, and close the cursor if it's still open.  
(Inherited from `BaseCursor`). |
<p>| Dispose | |</p>
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Duplicate</td>
<td>Create a new cursor that uses the same transaction and locker ID as the original cursor.</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified Object is equal to the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetEnumerator</td>
<td>Returns an enumerator that iterates through the Cursor.</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as a hash function for a particular type. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>Move</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>MoveFirst</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>MoveFirstMultiple</td>
<td>If positioning the cursor fails, CurrentMultiple will contain an empty KeyValuePair&lt;TKey, TValue&gt;.</td>
</tr>
<tr>
<td>MoveFirstMultipleKey</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>MoveLast</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>MoveMultiple</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>MoveMultipleKey</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>MoveNext</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>MoveNextDuplicate</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>MoveNextDuplicateMultiple</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>MoveNextDuplicateMultipleKey</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>MoveNextMultiple</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>MoveNextMultipleKey</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>MoveNextUnique</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>MoveNextUniqueMultiple</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>MoveNextUniqueMultipleKey</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>MovePrev</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>MovePrevDuplicate</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>MovePrevUnique</td>
<td>Overloaded. Overwrite the data of the key/data pair to which the cursor refers with the specified data item.</td>
</tr>
<tr>
<td>Overwrite</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>Refresh</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>RefreshMultiple</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>RefreshMultipleKey</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a <strong>String</strong> that represents the current <strong>Object</strong>. (Inherited from <strong>Object</strong>.)</td>
</tr>
</tbody>
</table>
See Also

Cursor Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Stores the key/data pair in the database.

**Namespace:** [BerkeleyDB](http://example.com)

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
# Syntax

## C#

```csharp
public void Add(
    KeyValuePair<DatabaseEntry, DatabaseEntry> pair
)
```

## Visual Basic (Declaration)

```vbnet
Public Sub Add (_
    pair As KeyValuePair(Of DatabaseEntry, DatabaseEntry) _
)
```

## Visual C++

```cpp
public:
    void Add(
        KeyValuePair<DatabaseEntry^, DatabaseEntry^> pair
    )
```

### Parameters

**pair**

- **Type:** `System.Collections.Generic.KeyValuePair(Of DatabaseEntry, DatabaseEntry)`
- The key/data pair to be stored in the database.
Remarks

If the underlying database supports duplicate data items, and if the key already exists in the database and a duplicate sort function has been specified, the inserted data item is added in its sorted location. If the key already exists in the database and no duplicate sort function has been specified, the inserted data item is added as the first of the data items for that key.
See Also

Cursor Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Delete the key/data pair to which the cursor refers.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

**C#**

`public void Delete()`

**Visual Basic (Declaration)**

`Public Sub Delete`  

**Visual C++**

```cpp
public:
void Delete();
```
Remarks

The cursor position is unchanged after a delete, and subsequent calls to cursor functions expecting the cursor to refer to an existing key will fail.
## Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>BerkeleyDB::KeyEmptyException</code></td>
<td>The element has already been deleted.</td>
</tr>
</tbody>
</table>
See Also

Cursor Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Create a new cursor that uses the same transaction and locker ID as the original cursor.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

```csharp
public Cursor Duplicate(
    bool keepPosition
)
```

Visual Basic (Declaration)

```vbnet
Public Function Duplicate ( _
    keepPosition As Boolean _
) As Cursor
```

Visual C++

```cpp
public: Cursor^ Duplicate(
    bool keepPosition
)
```

Parameters

keepPosition
Type: System::Boolean
If true, the newly created cursor is initialized to refer to the same position in the database as the original cursor (if any) and hold the same locks (if any). If false, or the original cursor does not hold a database position and locks, the created cursor is uninitialized and will behave like a cursor newly created by Cursor().

Return Value

A newly created cursor
Remarks

This is useful when an application is using locking and requires two or more cursors in the same thread of control.
See Also

Cursor Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Returns an enumerator that iterates through the Cursor.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public IEnumerator<KeyValuePair<DatabaseEntry, DatabaseEntry>> GetEnumerator

Visual Basic (Declaration)

Public Function GetEnumerator As IEnumerator(Of KeyValuePair(Of DatabaseEntry, DatabaseEntry))

Visual C++

public:
virtual IEnumerator<KeyValuePair<DatabaseEntry^, DatabaseEntry^>> GetEnumerator()

ReturnValue

An enumerator for the Cursor.

Implements

IEnumerator<(Of (T)>)::.GetEnumerator()()
Remarks

The enumerator will begin at the cursor's current position (or the first record if
the cursor has not yet been positioned) and iterate forwards (i.e. in the direction
of MoveNext()) over the remaining records.
See Also

Cursor Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
C#  Visual Basic  Visual C++  Include Protected Members  Include Inherited Members
Berkeley DB .NET API Documentation  Cursor....Move Method

Cursor Class  See Also
### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Move(DatabaseEntry, Boolean)</strong></td>
<td>Set the cursor to refer to key, and store the datum associated with the given key in Current. In the presence of duplicate key values, the first data item in the set of duplicates is stored in Current.</td>
</tr>
<tr>
<td><strong>Move(KeyValuePair&lt;Of &lt;(DatabaseEntry, DatabaseEntry)&gt;&gt;, Boolean)</strong></td>
<td>Move the cursor to the specified key/data pair of the database. The cursor is positioned to a key/data pair if both the key and data match the values provided on the key and data parameters.</td>
</tr>
<tr>
<td><strong>Move(DatabaseEntry, Boolean, LockingInfo)</strong></td>
<td>Set the cursor to refer to key, and store the datum associated with the given key in Current. In the presence of duplicate key values, the first data item in the set of duplicates is stored in Current.</td>
</tr>
<tr>
<td><strong>Move(KeyValuePair&lt;Of &lt;(DatabaseEntry, DatabaseEntry)&gt;&gt;, Boolean, LockingInfo)</strong></td>
<td>Move the cursor to the specified key/data pair of the database. The cursor is positioned to a key/data pair if both the key and data match the values provided on the key and data parameters.</td>
</tr>
</tbody>
</table>
See Also

Cursor Class  
Cursor Members  
BerkeleyDB Namespace  

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Set the cursor to refer to key, and store the datum associated with the given key in `Current`. In the presence of duplicate key values, the first data item in the set of duplicates is stored in `Current`.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool Move(
    DatabaseEntry key,
    bool exact
)

Visual Basic (Declaration)

Public Function Move ( _
    key As DatabaseEntry, _
    exact As Boolean _
) As Boolean

Visual C++

public:
    bool Move(
        DatabaseEntry^ key,
        bool exact
    )

Parameters

key
    Type: BerkeleyDB::DatabaseEntry
    The key at which to position the cursor

exact
    Type: System::Boolean
    If true, require the given key to match the key in the database exactly. If false, position the cursor at the smallest key greater than or equal to the specified key, permitting partial key matches and range searches.

Return Value
True if the cursor was positioned successfully, false otherwise.
Remarks

If positioning the cursor fails, Current will contain an empty KeyValuePair<Of <(TKey, TValue)> >.
See Also

Cursor Class
Move Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Move the cursor to the specified key/data pair of the database. The cursor is positioned to a key/data pair if both the key and data match the values provided on the key and data parameters.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
## Syntax

### C#

```csharp
public bool Move(
    KeyValuePair<DatabaseEntry, DatabaseEntry> pair,
    bool exact
)
```

### Visual Basic (Declaration)

```vbnet
Public Function Move (_
    pair As KeyValuePair(Of DatabaseEntry, DatabaseEntry), _
    exact As Boolean _
) As Boolean
```

### Visual C++

```cpp
public:
    bool Move(
        KeyValuePair<DatabaseEntry^, DatabaseEntry^> pair,
        bool exact
    )
```

## Parameters

### pair
Type: `System.Collections.Generic::KeyValuePair<DatabaseEntry, DatabaseEntry>`
The key/data pair at which to position the cursor.

### exact
Type: `System::::Boolean`
If true, require the given key and data to match the key and data in the database exactly. If false, position the cursor at the smallest data value which is greater than or equal to the value provided by pair.Value (as determined by the comparison function).
**Return Value**

True if the cursor was positioned successfully, false otherwise.
Remarks

If positioning the cursor fails, `Current` will contain an empty `KeyValuePair<(Of <(TKey, TValue>)>)`.

If this flag is specified on a database configured without sorted duplicate support, the value of exact is ignored.
See Also

Cursor Class
Move Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Set the cursor to refer to key, and store the datum associated with the given key in `Current`. In the presence of duplicate key values, the first data item in the set of duplicates is stored in `Current`.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool Move(
    DatabaseEntry key,
    bool exact,
    LockingInfo info
)

Visual Basic (Declaration)

Public Function Move (_
    key As DatabaseEntry, _
    exact As Boolean, _
    info As LockingInfo _
) As Boolean

Visual C++

public:
bool Move(
    DatabaseEntry key,
    bool exact,
    LockingInfo info
)

Parameters

key
Type: BerkeleyDB:::DatabaseEntry
The key at which to position the cursor

exact
Type: System:::Boolean
If true, require the given key to match the key in the database exactly. If false, position the cursor at the smallest key greater than or equal to the specified key, permitting partial key matches and range searches.
info
  Type: BerkeleyDB::::LockingInfo
  The locking behavior to use.

Return Value

True if the cursor was positioned successfully, false otherwise.
Remarks

If positioning the cursor fails, `Current` will contain an empty `KeyValuePair<(Of <(TKey, TValue)>)>`. 
See Also

Cursor Class
Move Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Move the cursor to the specified key/data pair of the database. The cursor is positioned to a key/data pair if both the key and data match the values provided on the key and data parameters.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

```csharp
public bool Move(
    KeyValuePair<DatabaseEntry, DatabaseEntry> pair,
    bool exact,
    LockingInfo info
)
```

Visual Basic (Declaration)

```vbnet
Public Function Move (_
    pair As KeyValuePair(Of DatabaseEntry, DatabaseEntry), _
    exact As Boolean, _
    info As LockingInfo _
) As Boolean
```

Visual C++

```cpp
public:
    bool Move(
        KeyValuePair<DatabaseEntry, DatabaseEntry>^ pair,
        bool^ exact,
        LockingInfo^ info
    )
```

Parameters

pair
Type: `System.Collections.Generic:::KeyValuePair(Of System::::DatabaseEntry, System::::DatabaseEntry)`)`
The key/data pair at which to position the cursor.

exact
Type: `System::::Boolean`
If true, require the given key and data to match the key and data in the
database exactly. If false, position the cursor at the smallest data value
which is greater than or equal to the value provided by pair.Value (as
determined by the comparison function).

info
Type: BerkeleyDB::::LockingInfo
The locking behavior to use.

Return Value

True if the cursor was positioned successfully, false otherwise.
Remarks

If positioning the cursor fails, `Current` will contain an empty `KeyValuePair<(Of (TKey, TValue))>.

If this flag is specified on a database configured without sorted duplicate support, the value of exact is ignored.
See Also

Cursor Class
Move Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Cursor.

::

MoveFirst Method

See Also

Cursor Class
# Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MoveFirst()</td>
<td>Set the cursor to refer to the first key/data pair of the database, and store that pair in <code>Current</code>.</td>
</tr>
<tr>
<td>MoveFirst(LockingInfo)</td>
<td>If the first key has duplicate values, the first data item in the set of duplicates is stored in <code>Current</code>.</td>
</tr>
</tbody>
</table>

Set the cursor to refer to the first key/data pair of the database, and store that pair in `Current`.

If the first key has duplicate values, the first data item in the set of duplicates is stored in `Current`.
See Also

Cursor Class
Cursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
MoveFirst Method

Set the cursor to refer to the first key/data pair of the database, and store that pair in Current. If the first key has duplicate values, the first data item in the set of duplicates is stored in Current.

Namespace: BerkeleyDB
Syntax

**C#**

```csharp
public bool MoveFirst()
```

**Visual Basic (Declaration)**

```vbnet
Public Function MoveFirst As Boolean
```

**Visual C++**

```cpp
public:
bool MoveFirst()
```

**Return Value**

True if the cursor was positioned successfully, false otherwise.
Remarks

If positioning the cursor fails, `Current` will contain an empty `KeyValuePair<(Of<br/>(TKey, TValue)>).`
See Also

Cursor Class
MoveFirst Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Set the cursor to refer to the first key/data pair of the database, and store that pair in `Current`. If the first key has duplicate values, the first data item in the set of duplicates is stored in `Current`.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

```csharp
public bool MoveFirst(
    LockingInfo info
)
```

**Visual Basic (Declaration)**

```
Public Function MoveFirst ( _
    info As LockingInfo _
) As Boolean
```

**Visual C++**

```c++
public:
bool MoveFirst(
    LockingInfo^ info
)
```

**Parameters**

info

Type: BerkeleyDB::::LockingInfo
The locking behavior to use.

**Return Value**

True if the cursor was positioned successfully, false otherwise.
Remarks

If positioning the cursor fails, **Current** will contain an empty **KeyValuePair**<**(TKey, TValue)>**.
See Also

Cursor Class
MoveFirst Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If positioning the cursor fails, `CurrentMultiple` will contain an empty `KeyValuePair(Of (TKey, TValue)>).`
<table>
<thead>
<tr>
<th>Overload List</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Name</strong></td>
</tr>
<tr>
<td>MoveFirstMultiple()()</td>
</tr>
<tr>
<td>MoveFirstMultiple(LockingInfo)</td>
</tr>
<tr>
<td>MoveFirstMultiple(Int32)</td>
</tr>
<tr>
<td>MoveFirstMultiple(Int32, LockingInfo)</td>
</tr>
</tbody>
</table>
See Also

Cursor Class
Cursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Set the cursor to refer to the first key/data pair of the database, and store that key and as many duplicate data items that can fit in a buffer the size of one database page in `CurrentMultiple`.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool MoveFirstMultiple()

Visual Basic (Declaration)

Public Function MoveFirstMultiple As Boolean

Visual C++

public:
bool MoveFirstMultiple()

Return Value

True if the cursor was positioned successfully, false otherwise.
See Also

Cursor Class
MoveFirstMultiple Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Set the cursor to refer to the first key/data pair of the database, and store that key and as many duplicate data items that can fit in a buffer the size of one database page in `CurrentMultiple`.

**Namespace**: BerkeleyDB  
**Assembly**: libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

```csharp
public bool MoveFirstMultiple(
    LockingInfo info
)
```

Visual Basic (Declaration)

```vbnet
Public Function MoveFirstMultiple ( _
    info As LockingInfo _
) As Boolean
```

Visual C++

```cpp
public:
bool MoveFirstMultiple(
    LockingInfo^ info
)
```

Parameters

info

Type: BerkeleyDB::LockingInfo

The locking behavior to use.

Return Value

True if the cursor was positioned successfully, false otherwise.
See Also

Cursor Class
MoveFirstMultiple Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Set the cursor to refer to the first key/data pair of the database, and store that key and as many duplicate data items that can fit in a buffer the size of BufferSize in CurrentMultiple.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

```csharp
public bool MoveFirstMultiple(
    int BufferSize
)
```

Visual Basic (Declaration)

```vbnet
Public Function MoveFirstMultiple ( _
    BufferSize As Integer _
) As Boolean
```

Visual C++

```cpp
public:
  bool MoveFirstMultiple(
    int BufferSize
)
```

Parameters

BufferSize
  Type: System::Int32
  The size of a buffer to fill with duplicate data items. Must be at least the page size of the underlying database and be a multiple of 1024.

Return Value

True if the cursor was positioned successfully, false otherwise.
See Also

Cursor Class
MoveFirstMultiple Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Set the cursor to refer to the first key/data pair of the database, and store that key and as many duplicate data items that can fit in a buffer the size of BufferSize in CurrentMultiple.

Namespace: BerkeleyDB
Syntax

C#

public bool MoveFirstMultiple(
    int BufferSize,
    LockingInfo info
)

Visual Basic (Declaration)

Public Function MoveFirstMultiple ( _
    BufferSize As Integer, _
    info As LockingInfo _
) As Boolean

Visual C++

public:
    bool MoveFirstMultiple(
        int BufferSize,
        LockingInfo^ info
    )

Parameters

BufferSize
    Type: System::Int32
    The size of a buffer to fill with duplicate data items. Must be at least the page size of the underlying database and be a multiple of 1024.

info
    Type: BerkeleyDB::LockingInfo
    The locking behavior to use.

Return Value

True if the cursor was positioned successfully, false otherwise.
See Also

Cursor Class
MoveFirstMultiple Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Cursor Class  See Also
### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MoveFirstMultipleKey()()</td>
<td>Set the cursor to refer to the first key/data pair of the database, and store that pair and as many ensuing key/data pairs that can fit in a buffer the size of one database page in CurrentMultipleKey.</td>
</tr>
<tr>
<td>MoveFirstMultipleKey(LockingInfo)</td>
<td>Set the cursor to refer to the first key/data pair of the database, and store that pair and as many ensuing key/data pairs that can fit in a buffer the size of one database page in CurrentMultipleKey.</td>
</tr>
<tr>
<td>MoveFirstMultipleKey(Int32)</td>
<td>Set the cursor to refer to the first key/data pair of the database, and store that pair and as many ensuing key/data pairs that can fit in a buffer the size of BufferSize in CurrentMultipleKey.</td>
</tr>
<tr>
<td>MoveFirstMultipleKey(Int32, LockingInfo)</td>
<td>Set the cursor to refer to the first key/data pair of the database, and store that pair and as many ensuing key/data pairs that can fit in a buffer the size of BufferSize in CurrentMultipleKey.</td>
</tr>
</tbody>
</table>
See Also

Cursor Class
Cursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Set the cursor to refer to the first key/data pair of the database, and store that pair and as many ensuing key/data pairs that can fit in a buffer the size of one database page in `CurrentMultipleKey`.

**Namespace:**  BerkeleyDB  
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#
public bool MoveFirstMultipleKey()

Visual Basic (Declaration)
Public Function MoveFirstMultipleKey As Boolean

Visual C++

public:
bool MoveFirstMultipleKey()

Return Value

True if the cursor was positioned successfully, false otherwise.
See Also

Cursor Class
MoveFirstMultipleKey Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Set the cursor to refer to the first key/data pair of the database, and store that pair and as many ensuing key/data pairs that can fit in a buffer the size of one database page in CurrentMultipleKey.

Namespace: BerkeleyDB
C#

```csharp
public bool MoveFirstMultipleKey(
    LockingInfo info
)
```

Visual Basic (Declaration)

```vbnet
Public Function MoveFirstMultipleKey ( _
    info As LockingInfo _
) As Boolean
```

Visual C++

```cpp
public:
    bool MoveFirstMultipleKey(
        LockingInfo^ info
    )
```

Parameters

- info
  - Type: `BerkeleyDB::LockingInfo`
  - The locking behavior to use.

Return Value

- True if the cursor was positioned successfully, false otherwise.
See Also

Cursor Class
MoveFirstMultipleKey Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Set the cursor to refer to the first key/data pair of the database, and store that pair and as many ensuing key/data pairs that can fit in a buffer the size of BufferSize in `CurrentMultipleKey`.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

```csharp
public bool MoveFirstMultipleKey(
    int BufferSize
)
```

Visual Basic (Declaration)

```vbnet
Public Function MoveFirstMultipleKey ( _
    BufferSize As Integer _
) As Boolean
```

Visual C++

```cpp
public:
    bool MoveFirstMultipleKey(
        int BufferSize
    )
```

Parameters

BufferSize

Type: `System::::Int32`

The size of a buffer to fill with key/data pairs. Must be at least the page size of the underlying database and be a multiple of 1024.

Return Value

True if the cursor was positioned successfully, false otherwise.
See Also

Cursor Class
MoveFirstMultipleKey Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Set the cursor to refer to the first key/data pair of the database, and store that pair and as many ensuing key/data pairs that can fit in a buffer the size of BufferSize in CurrentMultipleKey.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

```csharp
public bool MoveFirstMultipleKey(
    int BufferSize,
    LockingInfo info
)
```

Visual Basic (Declaration)

```vbnet
Public Function MoveFirstMultipleKey ( _
    BufferSize As Integer, _
    info As LockingInfo _
) As Boolean
```

Visual C++

```cpp
public:
    bool MoveFirstMultipleKey(
        int BufferSize,
        LockingInfo^ info
    )
```

Parameters

BufferSize
Type: `System::::Int32`
The size of a buffer to fill with key/data pairs. Must be at least the page size of the underlying database and be a multiple of 1024.

info
Type: `BerkeleyDB::::LockingInfo`
The locking behavior to use.

Return Value

True if the cursor was positioned successfully, false otherwise.
See Also

Cursor Class
MoveFirstMultipleKey Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Cursor Class  See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MoveLast()()()</td>
<td>Set the cursor to refer to the last key/data pair of the database, and store that pair in <strong>Current</strong>. If the last key has duplicate values, the last data item in the set of duplicates is stored in <strong>Current</strong>.</td>
</tr>
<tr>
<td>MoveLast(LockingInfo)</td>
<td>Set the cursor to refer to the last key/data pair of the database, and store that pair in <strong>Current</strong>. If the last key has duplicate values, the last data item in the set of duplicates is stored in <strong>Current</strong>.</td>
</tr>
</tbody>
</table>
See Also

Cursor Class
Cursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Set the cursor to refer to the last key/data pair of the database, and store that pair in `Current`. If the last key has duplicate values, the last data item in the set of duplicates is stored in `Current`.

**Namespace:** BerkeleyDB  
**Assembly:** `libdb_dotnet48` (in `libdb_dotnet48.dll`) Version: 4.8.24.0
Syntax

C#

public bool MoveLast()

Visual Basic (Declaration)

Public Function MoveLast As Boolean

Visual C++

public:
bool MoveLast()

Return Value

True if the cursor was positioned successfully, false otherwise.
Remarks

If positioning the cursor fails, Current will contain an empty KeyValuePair<(Of <(TKey, TValue)>)>.
See Also

Cursor Class
MoveLast Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Set the cursor to refer to the last key/data pair of the database, and store that pair in `Current`. If the last key has duplicate values, the last data item in the set of duplicates is stored in `Current`.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

```csharp
public bool MoveLast(
    LockingInfo info
)
```

Visual Basic (Declaration)

```vbnet
Public Function MoveLast (_
    info As LockingInfo _
) As Boolean
```

Visual C++

```cpp
public:
bool MoveLast(
    LockingInfo^ info

```

Parameters

info

Type: BerkeleyDB::::LockingInfo

The locking behavior to use.

Return Value

True if the cursor was positioned successfully, false otherwise.
Remarks

If positioning the cursor fails, Current will contain an empty KeyValuePair<(TKey, TValue)>.
See Also

Cursor Class
MoveLast Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Cursor Class

See Also
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MoveMultiple(DatabaseEntry, Boolean)</td>
<td>Set the cursor to refer to key, and store that key and as many duplicate data items associated with the given key that can fit in a buffer the size of one database page in CurrentMultiple.</td>
</tr>
<tr>
<td>MoveMultiple(KeyValuePair&lt;Of (DatabaseEntry, DatabaseEntry)&gt;, Boolean)</td>
<td>Move the cursor to the specified key/data pair of the database, and store that key/data pair and as many duplicate data items associated with the given key that can fit in a buffer the size of one database page in CurrentMultiple. The cursor is positioned to a key/data pair if both the key and data match the values provided on the key and data parameters.</td>
</tr>
<tr>
<td>MoveMultiple(DatabaseEntry, Boolean, LockingInfo)</td>
<td>Set the cursor to refer to key, and store that key and as many duplicate data items associated with the given key that can fit in a buffer the size of one database page in CurrentMultiple.</td>
</tr>
<tr>
<td>MoveMultiple(DatabaseEntry, Boolean, Int32)</td>
<td>Set the cursor to refer to key, and store that key and as many duplicate data items associated with the given key that can fit in a buffer the size of BufferSize in CurrentMultiple. Move the cursor to the specified key/data pair of the database, and store that key/data pair and as many duplicate data items associated with</td>
</tr>
</tbody>
</table>
**MoveMultiple(KeyValuePair(Of DatabaseEntry, DatabaseEntry), Boolean, LockingInfo)**

Move the cursor to the specified key/data pair of the database, and store that key/data pair and as many duplicate data items associated with the given key that can fit in a buffer the size of BufferSize in **CurrentMultiple**. The cursor is positioned to a key/data pair if both the key and data match the values provided on the key and data parameters.

**MoveMultiple(DatabaseEntry, Boolean, Int32, LockingInfo)**

Set the cursor to refer to key, and store that key and as many duplicate data items associated with the given key that can fit in a buffer the size of BufferSize in **CurrentMultiple**.

**MoveMultiple(KeyValuePair(Of DatabaseEntry, DatabaseEntry), Boolean, Int32, LockingInfo)**

Move the cursor to the specified key/data pair of the database, and store that key/data pair and as many duplicate data items associated with the given key that can fit in a buffer the size of BufferSize in **CurrentMultiple**. The cursor is positioned to a key/data pair if both the key and data match the values provided on the key and data parameters.
See Also

Cursor Class
Cursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Set the cursor to refer to key, and store that key and as many duplicate data items associated with the given key that can fit in a buffer the size of one database page in `CurrentMultiple`.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool MoveMultiple(
    DatabaseEntry key,
    bool exact
)

Visual Basic (Declaration)

Public Function MoveMultiple (_
    key As DatabaseEntry, _
    exact As Boolean _
) As Boolean

Visual C++

public:
bool MoveMultiple(
    DatabaseEntry^ key,
    bool exact
)

Parameters

key
Type: BerkeleyDB::DatabaseEntry
The key at which to position the cursor

exact
Type: System::Boolean
If true, require the given key to match the key in the database exactly. If false, position the cursor at the smallest key greater than or equal to the specified key, permitting partial key matches and range searches.

Return Value
True if the cursor was positioned successfully, false otherwise.
See Also

Cursor Class
MoveMultiple Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Move the cursor to the specified key/data pair of the database, and store that key/data pair and as many duplicate data items associated with the given key that can fit in a buffer the size of one database page in `CurrentMultiple`. The cursor is positioned to a key/data pair if both the key and data match the values provided on the key and data parameters.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

```csharp
public bool MoveMultiple(
    KeyValuePair<DatabaseEntry, DatabaseEntry> pair,
    bool exact
)
```

Visual Basic (Declaration)

```vbnet
Public Function MoveMultiple (_
    pair As KeyValuePair(Of DatabaseEntry, DatabaseEntry), _
    exact As Boolean _
) As Boolean
```

Visual C++

```cpp
public:
bool MoveMultiple(
    KeyValuePair<DatabaseEntry^, DatabaseEntry^> pair,
    bool exact
)
```

Parameters

pair

Type: System.Collections.Generic.KeyValuePair<DatabaseEntry, DatabaseEntry>

The key/data pair at which to position the cursor.

exact

Type: System::Boolean

If true, require the given key and data to match the key and data in the database exactly. If false, position the cursor at the smallest data value which is greater than or equal to the value provided by pair.Value (as determined by the comparison function).
Return Value

True if the cursor was positioned successfully, false otherwise.
See Also

Cursor Class
MoveMultiple Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Cursor::MoveMultiple Method (DatabaseEntry, Boolean, LockingInfo)

Set the cursor to refer to key, and store that key and as many duplicate data items associated with the given key that can fit in a buffer the size of one database page in CurrentMultiple.

Namespace: BerkeleyDB
### Syntax

**C#**

```csharp
public bool MoveMultiple(
    DatabaseEntry key,
    bool exact,
    LockingInfo info
)
```

**Visual Basic (Declaration)**

```vbnet
Public Function MoveMultiple (_
    key As DatabaseEntry, _
    exact As Boolean, _
    info As LockingInfo _
) As Boolean
```

**Visual C++**

```c++
public:
    bool MoveMultiple(
        DatabaseEntry^ key,
        bool exact,
        LockingInfo^ info
    )
```

### Parameters

**key**

Type: `BerkeleyDB::DatabaseEntry`

The key at which to position the cursor.

**exact**

Type: `System::Boolean`

If true, require the given key to match the key in the database exactly. If false, position the cursor at the smallest key greater than or equal to the specified key, permitting partial key matches and range searches.
info

Type: BerkeleyDB::<LockingInfo>
The locking behavior to use.

**Return Value**

True if the cursor was positioned successfully, false otherwise.
See Also

Cursor Class
MoveMultiple Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Set the cursor to refer to key, and store that key and as many duplicate data items associated with the given key that can fit in a buffer the size of BufferSize in $\text{CurrentMultiple}$.

**Namespace:**  BerkeleyDB  
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

```csharp
public bool MoveMultiple(
    DatabaseEntry key,
    bool exact,
    int BufferSize
)
```

Visual Basic (Declaration)

```vbnet
Public Function MoveMultiple (_
    key As DatabaseEntry, _
    exact As Boolean, _
    BufferSize As Integer _
) As Boolean
```

Visual C++

```cpp
public:
bool MoveMultiple(
    DatabaseEntry^ key,
    bool exact,
    int BufferSize
)
```

Parameters

key
Type: `BerkeleyDB::DatabaseEntry`
The key at which to position the cursor

exact
Type: `System::Boolean`
If true, require the given key to match the key in the database exactly. If false, position the cursor at the smallest key greater than or equal to the specified key, permitting partial key matches and range searches.
BufferSize
    Type: System::Int32
    The size of a buffer to fill with duplicate data items. Must be at least the page size of the underlying database and be a multiple of 1024.

Return Value

True if the cursor was positioned successfully, false otherwise.
See Also

Cursor Class
MoveMultiple Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Move the cursor to the specified key/data pair of the database, and store that key/data pair and as many duplicate data items associated with the given key that can fit in a buffer the size of one database page in \texttt{CurrentMultiple}. The cursor is positioned to a key/data pair if both the key and data match the values provided on the key and data parameters.

\textbf{Namespace:} \texttt{BerkeleyDB}

\textbf{Assembly:} \texttt{libdb_dotnet48 (in libdb_dotnet48.dll)} Version: 4.8.24.0
C#

```csharp
public bool MoveMultiple(
    KeyValuePair<DatabaseEntry, DatabaseEntry> pair,
    bool exact,
    LockingInfo info
)
```

Visual Basic (Declaration)

```vbnet
Public Function MoveMultiple (_
    pair As KeyValuePair(Of DatabaseEntry, DatabaseEntry), _
    exact As Boolean, _
    info As LockingInfo _
) As Boolean
```

Visual C++

```cpp
public:
    bool MoveMultiple(
        KeyValuePair<DatabaseEntry, DatabaseEntry> pair,
        bool exact,
        LockingInfo info
    )
```

Parameters

pair

Type: `System.Collections.Generic:::KeyValuePair(Of <(DatabaseEntry, DatabaseEntry)>)`

The key/data pair at which to position the cursor.

effect

Type: `System:::Boolean`

If true, require the given key and data to match the key and data in the database exactly. If false, position the cursor at the smallest data value which is greater than or equal to the value provided by pair.Value (as
determined by the comparison function).

info
  Type: BerkeleyDB::LockingInfo
  The locking behavior to use.

**Return Value**

True if the cursor was positioned successfully, false otherwise.
See Also

Cursor Class
MoveMultiple Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Move the cursor to the specified key/data pair of the database, and store that key/data pair and as many duplicate data items associated with the given key that can fit in a buffer the size of BufferSize in CurrentMultiple. The cursor is positioned to a key/data pair if both the key and data match the values provided on the key and data parameters.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool MoveMultiple(
    KeyValuePair<DatabaseEntry, DatabaseEntry> pair,
    bool exact,
    int BufferSize
)

Visual Basic (Declaration)

Public Function MoveMultiple (_
    pair As KeyValuePair(Of DatabaseEntry, DatabaseEntry), _
    exact As Boolean, _
    BufferSize As Integer _) _
) As Boolean

Visual C++

public:
bool MoveMultiple(
    KeyValuePair<DatabaseEntry^, DatabaseEntry^> pair,
    bool exact,
    int BufferSize
)

Parameters

pair
    Type: System.Collections.Generic.KeyValuePair(Of DatabaseEntry, DatabaseEntry)
    The key/data pair at which to position the cursor.

exact
    Type: System.Boolean
    If true, require the given key and data to match the key and data in the
database exactly. If false, position the cursor at the smallest data value
which is greater than or equal to the value provided by pair.Value (as
determined by the comparison function).

BufferSize
Type: System::Int32
The size of a buffer to fill with duplicate data items. Must be at least the page size of the underlying database and be a multiple of 1024.

Return Value

True if the cursor was positioned successfully, false otherwise.
See Also

Cursor Class
MoveMultiple Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Cursor....MoveMultiple Method (DatabaseEntry, Boolean, Int32, LockingInfo)

Set the cursor to refer to key, and store that key and as many duplicate data items associated with the given key that can fit in a buffer the size of BufferSize in CurrentMultiple.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
## Syntax

### C#

```csharp
public bool MoveMultiple(
    DatabaseEntry key,
    bool exact,
    int BufferSize,
    LockingInfo info
)
```

### Visual Basic (Declaration)

```vbnet
Public Function MoveMultiple (  
    key As DatabaseEntry,   
    exact As Boolean,   
    BufferSize As Integer,   
    info As LockingInfo    
) As Boolean
```

### Visual C++

```cpp
public:  
    bool MoveMultiple(
        DatabaseEntry^ key,
        bool exact,
        int BufferSize,
        LockingInfo^ info
    )
```

## Parameters

### key
  - **Type:** BerkeleyDB::DatabaseEntry
  - The key at which to position the cursor

### exact
  - **Type:** System::::Boolean
  - If true, require the given key to match the key in the database exactly. If false, position the cursor at the smallest key greater than or equal to the
specified key, permitting partial key matches and range searches.

BufferSize
Type: System::Int32
The size of a buffer to fill with duplicate data items. Must be at least the page size of the underlying database and be a multiple of 1024.

info
Type: BerkeleyDB::LockingInfo
The locking behavior to use.

Return Value

True if the cursor was positioned successfully, false otherwise.
See Also

Cursor Class
MoveMultiple Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Move the cursor to the specified key/data pair of the database, and store that key/data pair and as many duplicate data items associated with the given key that can fit in a buffer the size of BufferSize in `CurrentMultiple`. The cursor is positioned to a key/data pair if both the key and data match the values provided on the key and data parameters.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

```csharp
public bool MoveMultiple(
    KeyValuePair<DatabaseEntry, DatabaseEntry> pair,
    bool exact,
    int BufferSize,
    LockingInfo info
)
```

Visual Basic (Declaration)

```vbnet
Public Function MoveMultiple (_
    pair As KeyValuePair(Of DatabaseEntry, DatabaseEntry), _
    exact As Boolean, _
    BufferSize As Integer, _
    info As LockingInfo _
) As Boolean
```

Visual C++

```cpp
public:
    bool MoveMultiple(
        KeyValuePair<DatabaseEntry^, DatabaseEntry^> pair,
        bool exact,
        int BufferSize,
        LockingInfo^ info
    )
```

Parameters

pair
Type: System.Collections.Generic.KeyValuePair<DatabaseEntry, DatabaseEntry>
The key/data pair at which to position the cursor.

exact
Type: System.Boolean
If true, require the given key and data to match the key and data in the
database exactly. If false, position the cursor at the smallest data value which is greater than or equal to the value provided by pair.Value (as determined by the comparison function).

BufferSize
Type: System::Int32
The size of a buffer to fill with duplicate data items. Must be at least the page size of the underlying database and be a multiple of 1024.

info
Type: BerkeleyDB::LockingInfo
The locking behavior to use.

**Return Value**

True if the cursor was positioned successfully, false otherwise.
See Also

Cursor Class
MoveMultiple Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation

Cursor Class

See Also
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MoveMultipleKey(DatabaseEntry, Boolean)</td>
<td>Set the cursor to refer to key, and store that key and as many ensuing key/data pairs that can fit in a buffer the size of one database page in CurrentMultipleKey. Move the cursor to the specified key/data pair of the database, and store that key/data pair and as many ensuing key/data pairs that can fit in a buffer the size of one database page in CurrentMultipleKey. The cursor is positioned to a key/data pair if both the key and data match the values provided on the key and data parameters.</td>
</tr>
</tbody>
</table>
| MoveMultipleKey(KeyValuePair<
  Of (DatabaseEntry, DatabaseEntry)>,
  Boolean) |                                                                                                                                                                                                             |
| MoveMultipleKey(DatabaseEntry, Boolean, LockingInfo) | Set the cursor to refer to key, and store that key and as many ensuing key/data pairs that can fit in a buffer the size of one database page in CurrentMultipleKey.                                                                 |
| MoveMultipleKey(DatabaseEntry, Boolean, Int32) | Set the cursor to refer to key, and store that key and as many ensuing key/data pairs that can fit in a buffer the size of BufferSize in CurrentMultipleKey. Move the cursor to the specified key/data pair of the database, and store that key/data pair and |
MoveMultipleKey(KeyValuePair<(Of <(DatabaseEntry, DatabaseEntry>>)>, Boolean, LockingInfo)

as many ensuing key/data pairs that can fit in a buffer the size of one database page in CurrentMultipleKey. The cursor is positioned to a key/data pair if both the key and data match the values provided on the key and data parameters.

Move the cursor to the specified key/data pair of the database, and store that key/data pair and as many ensuing key/data pairs that can fit in a buffer the size of BufferSize in CurrentMultipleKey. The cursor is positioned to a key/data pair if both the key and data match the values provided on the key and data parameters.

Set the cursor to refer to key, and store that key and as many ensuing key/data pairs that can fit in a buffer the size of BufferSize in CurrentMultipleKey.

Move the cursor to the specified key/data pair of the database, and store that key/data pair and as many ensuing key/data pairs that can fit in a buffer the size of BufferSize in CurrentMultipleKey. The cursor is positioned to a key/data pair if both the key and data match the values provided on the key and data parameters.
See Also

Cursor Class
Cursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Set the cursor to refer to key, and store that key and as many ensuing key/data pairs that can fit in a buffer the size of one database page in `CurrentMultipleKey`.

**Namespace:** [BerkeleyDB](https://www berkleydb.com)

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
# Syntax

**C#**

```csharp
public bool MoveMultipleKey(
    DatabaseEntry key,
    bool exact
)
```

**Visual Basic (Declaration)**

```vbnet
Public Function MoveMultipleKey ( _
    key As DatabaseEntry, _
    exact As Boolean _
) As Boolean
```

**Visual C++**

```cpp
public:
    bool MoveMultipleKey(
        DatabaseEntry^ key,
        bool exact
    )
```

## Parameters

- **key**
  - Type: `BerkeleyDB::DatabaseEntry`
  - The key at which to position the cursor

- **exact**
  - Type: `System::Boolean`
  - If true, require the given key to match the key in the database exactly. If false, position the cursor at the smallest key greater than or equal to the specified key, permitting partial key matches and range searches.

## Return Value
True if the cursor was positioned successfully, false otherwise.
See Also

Cursor Class
MoveMultipleKey Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Move the cursor to the specified key/data pair of the database, and store that key/data pair and as many ensuing key/data pairs that can fit in a buffer the size of one database page in CurrentMultipleKey. The cursor is positioned to a key/data pair if both the key and data match the values provided on the key and data parameters.

Namespace: BerkeleyDB
Syntax

C#

public bool MoveMultipleKey(
    KeyValuePair<DatabaseEntry, DatabaseEntry> pair,
    bool exact
)

Visual Basic (Declaration)

Public Function MoveMultipleKey ( _
    pair As KeyValuePair(Of DatabaseEntry, DatabaseEntry), _
    exact As Boolean _
) As Boolean

Visual C++

public:
bool MoveMultipleKey(
    KeyValuePair<DatabaseEntry, DatabaseEntry> pair,
    bool exact
)

Parameters

pair
    Type: System.Collections.Generic.:..:KeyValuePair<Of <(DatabaseEntry, DatabaseEntry)>>
    The key/data pair at which to position the cursor.

exact
    Type: System..::.Boolean
    If true, require the given key and data to match the key and data in the database exactly. If false, position the cursor at the smallest data value which is greater than or equal to the value provided by pair.Value (as determined by the comparison function).
Return Value

True if the cursor was positioned successfully, false otherwise.
See Also

Cursor Class
MoveMultipleKey Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Cursor...::MoveMultipleKey Method (DatabaseEntry, Boolean, LockingInfo)

**Cursor Class**  **See Also**

Set the cursor to refer to key, and store that key and as many ensuing key/data pairs that can fit in a buffer the size of one database page in **CurrentMultipleKey**.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

**C#**

```csharp
public bool MoveMultipleKey(
    DatabaseEntry key,
    bool exact,
    LockingInfo info
)
```

**Visual Basic (Declaration)**

```vbnet
Public Function MoveMultipleKey ( _
    key As DatabaseEntry, _
    exact As Boolean, _
    info As LockingInfo _
) As Boolean
```

**Visual C++**

```c++
public:
bool MoveMultipleKey(
    DatabaseEntry^ key,
    bool exact,
    LockingInfo^ info
)
```

**Parameters**

**key**

Type: `BerkeleyDB::DatabaseEntry`

The key at which to position the cursor

**exact**

Type: `System::Boolean`

If true, require the given key to match the key in the database exactly. If false, position the cursor at the smallest key greater than or equal to the specified key, permitting partial key matches and range searches.
Type: [BerkeleyDB::::LockingInfo](#)
The locking behavior to use.

**Return Value**

True if the cursor was positioned successfully, false otherwise.
See Also

Cursor Class
MoveMultipleKey Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Cursor...::MoveMultipleKey Method (DatabaseEntry, Boolean, Int32)

Set the cursor to refer to key, and store that key and as many ensuing key/data pairs that can fit in a buffer the size of BufferSize in CurrentMultipleKey.

Namespace: BerkeleyDB
Syntax

C#

```csharp
public bool MoveMultipleKey(
    DatabaseEntry key,
    bool exact,
    int BufferSize
)
```

Visual Basic (Declaration)

```vbnet
Public Function MoveMultipleKey (_
    key As DatabaseEntry, _
    exact As Boolean, _
    BufferSize As Integer _
) As Boolean
```

Visual C++

```cpp
public:
    bool MoveMultipleKey(
        DatabaseEntry^ key,
        bool exact,
        int BufferSize
    )
```

Parameters

key
Type: BerkeleyDB::DatabaseEntry
The key at which to position the cursor

exact
Type: System::Boolean
If true, require the given key to match the key in the database exactly. If false, position the cursor at the smallest key greater than or equal to the specified key, permitting partial key matches and range searches.
BufferSize
Type: System::Int32
The size of a buffer to fill with key/data pairs. Must be at least the page size of the underlying database and be a multiple of 1024.

Return Value

True if the cursor was positioned successfully, false otherwise.
See Also

Cursor Class
MoveMultipleKey Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation

Cursor...:.MoveMultipleKey Method (KeyValuePair<Of <(DatabaseEntry, DatabaseEntry)>>, Boolean, LockingInfo)

**Cursor Class**  **See Also**

Move the cursor to the specified key/data pair of the database, and store that key/data pair and as many ensuing key/data pairs that can fit in a buffer the size of one database page in CurrentMultipleKey. The cursor is positioned to a key/data pair if both the key and data match the values provided on the key and data parameters.

**Namespace:** BerkeleyDB
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
**Syntax**

**C#**

```csharp
public bool MoveMultipleKey(
    KeyValuePair<DatabaseEntry, DatabaseEntry> pair,
    bool exact,
    LockingInfo info
)
```

**Visual Basic (Declaration)**

```vbnet
Public Function MoveMultipleKey (_
    pair As KeyValuePair(Of DatabaseEntry, DatabaseEntry), _
    exact As Boolean, _
    info As LockingInfo _
) As Boolean
```

**Visual C++**

```cpp
public: 
    bool MoveMultipleKey(
        KeyValuePair<DatabaseEntry^, DatabaseEntry^> pair, 
        bool exact, 
        LockingInfo^ info 
    )
```

**Parameters**

**pair**
- Type: `System.Collections.Generic:::KeyValuePair(Of (DatabaseEntry, DatabaseEntry))`
- The key/data pair at which to position the cursor.

**exact**
- Type: `System:::Boolean`
- If true, require the given key and data to match the key and data in the database exactly. If false, position the cursor at the smallest data value which is greater than or equal to the value provided by pair.Value (as
determined by the comparison function).

info
Type: BerkeleyDB::LockingInfo
The locking behavior to use.

Return Value

True if the cursor was positioned successfully, false otherwise.
See Also

Cursor Class
MoveMultipleKey Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Cursor Class  See Also

Move the cursor to the specified key/data pair of the database, and store that key/data pair and as many ensuing key/data pairs that can fit in a buffer the size of BufferSize in CurrentMultipleKey. The cursor is positioned to a key/data pair if both the key and data match the values provided on the key and data parameters.

Namespace: BerkeleyDB
## Syntax

### C#

```csharp
public bool MoveMultipleKey(
    KeyValuePair<DatabaseEntry, DatabaseEntry> pair,
    bool exact,
    int BufferSize
)
```

### Visual Basic (Declaration)

```vbnet
Public Function MoveMultipleKey ( _
    pair As KeyValuePair(Of DatabaseEntry, DatabaseEntry), _
    exact As Boolean, _
    BufferSize As Integer _
) As Boolean
```

### Visual C++

```cpp
public:
    bool MoveMultipleKey(
        KeyValuePair<DatabaseEntry, DatabaseEntry> pair,
        bool exact,
        int BufferSize
    )
```

## Parameters

**pair**
- Type: `System.Collections.Generic:::KeyValuePair<DatabaseEntry, DatabaseEntry>`
- The key/data pair at which to position the cursor.

**exact**
- Type: `System:::Boolean`
- If true, require the given key and data to match the key and data in the database exactly. If false, position the cursor at the smallest data value which is greater than or equal to the value provided by pair.Value (as
determined by the comparison function).

BufferSize
Type: System::<Int32>
The size of a buffer to fill with key/data pairs. Must be at least the page size of the underlying database and be a multiple of 1024.

Return Value

True if the cursor was positioned successfully, false otherwise.
See Also

Cursor Class
MoveMultipleKey Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Set the cursor to refer to key, and store that key and as many ensuing key/data pairs that can fit in a buffer the size of BufferSize in `CurrentMultipleKey`.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
## Syntax

### C#

```csharp
public bool MoveMultipleKey(
    DatabaseEntry key,
    bool exact,
    int BufferSize,
    LockingInfo info
)
```

### Visual Basic (Declaration)

```vbnet
Public Function MoveMultipleKey (_
    key As DatabaseEntry, _
    exact As Boolean, _
    BufferSize As Integer, _
    info As LockingInfo _
) As Boolean
```

### Visual C++

```c++
public:
    bool MoveMultipleKey(
        DatabaseEntry^ key,
        bool exact,
        int BufferSize,
        LockingInfo^ info
    )
```

## Parameters

**key**

Type: `BerkeleyDB..::.DatabaseEntry`

The key at which to position the cursor

**exact**

Type: `System..::.Boolean`

If true, require the given key to match the key in the database exactly. If false, position the cursor at the smallest key greater than or equal to the
specified key, permitting partial key matches and range searches.

BufferSize
   Type: System::Int32
   The size of a buffer to fill with key/data pairs. Must be at least the page size of the underlying database and be a multiple of 1024.

info
   Type: BerkeleyDB::LockingInfo
   The locking behavior to use.

Return Value

True if the cursor was positioned successfully, false otherwise.
See Also

Cursor Class
MoveMultipleKey Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Move the cursor to the specified key/data pair of the database, and store that key/data pair and as many ensuing key/data pairs that can fit in a buffer the size of BufferSize in **CurrentMultipleKey**. The cursor is positioned to a key/data pair if both the key and data match the values provided on the key and data parameters.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

```csharp
public bool MoveMultipleKey(
    KeyValuePair<DatabaseEntry, DatabaseEntry> pair,
    bool exact,
    int BufferSize,
    LockingInfo info
)
```

Visual Basic (Declaration)

```vbnet
Public Function MoveMultipleKey ( _
    pair As KeyValuePair(Of DatabaseEntry, DatabaseEntry), _
    exact As Boolean, _
    BufferSize As Integer, _
    info As LockingInfo _
) As Boolean
```

Visual C++

```cpp
public:
    bool MoveMultipleKey(
        KeyValuePair<DatabaseEntry^, DatabaseEntry^> pair,
        bool exact,
        int BufferSize,
        LockingInfo^ info
    )
```

Parameters

pair
Type: `System.Collections.Generic.KeyValuePair<(Of <(DatabaseEntry, DatabaseEntry)>)`
The key/data pair at which to position the cursor.

exact
Type: `System.Boolean`
If true, require the given key and data to match the key and data in the
database exactly. If false, position the cursor at the smallest data value which is greater than or equal to the value provided by pair.Value (as determined by the comparison function).

BufferSize
Type: System::Int32
The size of a buffer to fill with key/data pairs. Must be at least the page size of the underlying database and be a multiple of 1024.

info
Type: BerkeleyDB::LockingInfo
The locking behavior to use.

**Return Value**

True if the cursor was positioned successfully, false otherwise.
See Also

Cursor Class
MoveMultipleKey Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Cursor Class  See Also
Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MoveNext()()</td>
<td>If the cursor is not yet initialized, MoveNext is identical to MoveFirst(). Otherwise, move the cursor to the next key/data pair of the database, and store that pair in Current. In the presence of duplicate key values, the value of Current.Key may not change.</td>
</tr>
<tr>
<td>MoveNext(LockingInfo)</td>
<td>If the cursor is not yet initialized, MoveNext is identical to MoveFirst(LockingInfo). Otherwise, move the cursor to the next key/data pair of the database, and store that pair in Current. In the presence of duplicate key values, the value of Current.Key may not change.</td>
</tr>
</tbody>
</table>
See Also

Cursor Class
Cursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If the cursor is not yet initialized, MoveNext is identical to MoveFirst(). Otherwise, move the cursor to the next key/data pair of the database, and store that pair in Current. In the presence of duplicate key values, the value of Current.Key may not change.

Namespace: BerkeleyDB
Syntax

C#

public bool MoveNext()

Visual Basic (Declaration)

Public Function MoveNext As Boolean

Visual C++

public:
bool MoveNext()

Return Value

True if the cursor was positioned successfully, false otherwise.
Remarks

If positioning the cursor fails, `Current` will contain an empty `KeyValuePair<Of <(TKey, TValue)>>`. 
See Also

Cursor Class
MoveNext Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If the cursor is not yet initialized, MoveNext is identical to MoveFirst(LockingInfo). Otherwise, move the cursor to the next key/data pair of the database, and store that pair in Current. In the presence of duplicate key values, the value of Current.Key may not change.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

```csharp
public bool MoveNext(
    LockingInfo info
)
```

Visual Basic (Declaration)

```vbnet
Public Function MoveNext ( _
    info As LockingInfo _
) As Boolean
```

Visual C++

```cpp
public:
bool MoveNext(
    LockingInfo^ info
)
```

Parameters

info

Type: BerkeleyDB::::LockingInfo
The locking behavior to use.

Return Value

True if the cursor was positioned successfully, false otherwise.
Remarks

If positioning the cursor fails, `Current` will contain an empty `KeyValuePair<(Of <(TKey, TValue)>)>`. 
See Also

Cursor Class
MoveNext Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
C#
Visual Basic
Visual C++
Include Protected Members
Include Inherited Members
Berkeley DB .NET API Documentation
Cursor...:MoveNextDuplicate Method
Cursor Class  See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MoveNextDuplicate()()</td>
<td>If the next key/data pair of the database is a duplicate data record for the current key/data pair, move the cursor to the next key/data pair in the database, and store that pair in Current. MoveNextDuplicate will return false if the next key/data pair of the database is not a duplicate data record for the current key/data pair.</td>
</tr>
<tr>
<td>MoveNextDuplicate(LockingInfo)</td>
<td>If the next key/data pair of the database is a duplicate data record for the current key/data pair, move the cursor to the next key/data pair in the database, and store that pair in Current. MoveNextDuplicate will return false if the next key/data pair of the database is not a duplicate data record for the current key/data pair.</td>
</tr>
</tbody>
</table>
See Also

Cursor Class
Cursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If the next key/data pair of the database is a duplicate data record for the current key/data pair, move the cursor to the next key/data pair in the database, and store that pair in Current. MoveNextDuplicate will return false if the next key/data pair of the database is not a duplicate data record for the current key/data pair.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
### Syntax

**C#**

```csharp
public bool MoveNextDuplicate()
```

**Visual Basic (Declaration)**

```vbnet
Public Function MoveNextDuplicate As Boolean
```

**Visual C++**

```cpp
public:
bool MoveNextDuplicate()
```

### Return Value

True if the cursor was positioned successfully, false otherwise.
Remarks

If positioning the cursor fails, Current will contain an empty KeyValuePair(Of (TKey, TValue)>).
See Also

Cursor Class
MoveNextDuplicate Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If the next key/data pair of the database is a duplicate data record for the current key/data pair, move the cursor to the next key/data pair in the database, and store that pair in `Current`. `MoveNextDuplicate` will return false if the next key/data pair of the database is not a duplicate data record for the current key/data pair.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

```csharp
public bool MoveNextDuplicate(
    LockingInfo info
)
```

Visual Basic (Declaration)

```vbnet
Public Function MoveNextDuplicate ( _
    info As LockingInfo _
) As Boolean
```

Visual C++

```cpp
public:
    bool MoveNextDuplicate( 
        LockingInfo^ info
    )
```

Parameters

info

- Type: `BerkeleyDB::::LockingInfo`
- The locking behavior to use.

Return Value

True if the cursor was positioned successfully, false otherwise.
Remarks

If positioning the cursor fails, `Current` will contain an empty `KeyValuePair<(Of `<(TKey, TValue)>`)`.
See Also

Cursor Class
MoveNextDuplicate Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Cursor..::MoveNextDuplicateMultiple Method

Cursor Class  See Also
### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MoveNextDuplicateMultiple()()()</td>
<td>If the next key/data pair of the database is a duplicate data record for the current key/data pair, move the cursor to the next key/data pair in the database, and store that pair and as many duplicate data items that can fit in a buffer the size of one database page in <code>CurrentMultiple</code>. MoveNextDuplicateMultiple will return false if the next key/data pair of the database is not a duplicate data record for the current key/data pair. If the next key/data pair of the database is a duplicate data record for the current key/data pair, move the cursor to the next key/data pair in the database, and store that pair and as many duplicate data items that can fit in a buffer the size of one database page in <code>CurrentMultiple</code>. MoveNextDuplicateMultiple will return false if the next key/data pair of the database is not a duplicate data record for the current key/data pair.</td>
</tr>
<tr>
<td>MoveNextDuplicateMultiple(LockingInfo)</td>
<td></td>
</tr>
</tbody>
</table>
MoveNextDuplicateMultiple(Int32)

the database is a duplicate data record for the current key/data pair, then move
cursor to the next key/data pair in the database, and store that pair and as many
duplicate data items that can fit in a buffer the size of BufferSize in
CurrentMultiple.
MoveNextDuplicateMultiple
will return false if the next key/data pair of the database
is not a duplicate data record for the current key/data pair.
If the next key/data pair of the database is a duplicate
data record for the current key/data pair, move the
cursor to the next key/data pair in the database, and
store that pair and as many duplicate data items that can
fit in a buffer the size of BufferSize in
CurrentMultiple.
MoveNextDuplicateMultiple
will return false if the next key/data pair of the database
is not a duplicate data record for the current key/data pair.
See Also

Cursor Class
Cursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If the next key/data pair of the database is a duplicate data record for the current key/data pair, move the cursor to the next key/data pair in the database, and store that pair and as many duplicate data items that can fit in a buffer the size of one database page in `CurrentMultiple`. `MoveNextDuplicateMultiple` will return false if the next key/data pair of the database is not a duplicate data record for the current key/data pair.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll)  
**Version:** 4.8.24.0
Syntax

C#

public bool MoveNextDuplicateMultiple()

Visual Basic (Declaration)

Public Function MoveNextDuplicateMultiple As Boolean

Visual C++

public:
bool MoveNextDuplicateMultiple()

Return Value

True if the cursor was positioned successfully, false otherwise.
See Also

Cursor Class
MoveNextDuplicateMultiple Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If the next key/data pair of the database is a duplicate data record for the current key/data pair, move the cursor to the next key/data pair in the database, and store that pair and as many duplicate data items that can fit in a buffer the size of one database page in CurrentMultiple. MoveNextDuplicateMultiple will return false if the next key/data pair of the database is not a duplicate data record for the current key/data pair.

Namespace: BerkeleyDB
Syntax

C#

public bool MoveNextDuplicateMultiple(
    LockingInfo info
)

Visual Basic (Declaration)

Public Function MoveNextDuplicateMultiple ( _
    info As LockingInfo _
) As Boolean

Visual C++

public:
    bool MoveNextDuplicateMultiple(
        LockingInfo^ info
    )

Parameters

info
    Type: BerkeleyDB::LockingInfo
    The locking behavior to use.

Return Value

True if the cursor was positioned successfully, false otherwise.
See Also

Cursor Class
MoveNextDuplicateMultiple Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If the next key/data pair of the database is a duplicate data record for the current key/data pair, then move cursor to the next key/data pair in the database, and store that pair and as many duplicate data items that can fit in a buffer the size of BufferSize in `CurrentMultiple`. `MoveNextDuplicateMultiple` will return false if the next key/data pair of the database is not a duplicate data record for the current key/data pair.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
**Syntax**

**C#**

```csharp
public bool MoveNextDuplicateMultiple(
    int BufferSize
)
```

**Visual Basic (Declaration)**

```vbnet
Public Function MoveNextDuplicateMultiple ( _
    BufferSize As Integer _
) As Boolean
```

**Visual C++**

```cpp
public:
    bool MoveNextDuplicateMultiple(
        int BufferSize
    )
```

**Parameters**

**BufferSize**

Type: `System::::Int32`

The size of a buffer to fill with duplicate data items. Must be at least the page size of the underlying database and be a multiple of 1024.

**Return Value**

True if the cursor was positioned successfully, false otherwise.
See Also

Cursor Class
MoveNextDuplicateMultiple Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If the next key/data pair of the database is a duplicate data record for the current key/data pair, move the cursor to the next key/data pair in the database, and store that pair and as many duplicate data items that can fit in a buffer the size of BufferSize in CurrentMultiple. MoveNextDuplicateMultiple will return false if the next key/data pair of the database is not a duplicate data record for the current key/data pair.

**Namespace**: BerkeleyDB  
**Assembly**: libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool MoveNextDuplicateMultiple(
    int BufferSize,
    LockingInfo info
)

Visual Basic (Declaration)

Public Function MoveNextDuplicateMultiple ( _
    BufferSize As Integer, _
    info As LockingInfo _
) As Boolean

Visual C++

public:
    bool MoveNextDuplicateMultiple(
    int BufferSize,
    LockingInfo info
)

Parameters

BufferSize
Type: System::Int32
The size of a buffer to fill with duplicate data items. Must be at least the page size of the underlying database and be a multiple of 1024.

info
Type: BerkeleyDB::LockingInfo
The locking behavior to use.

Return Value

True if the cursor was positioned successfully, false otherwise.
See Also

Cursor Class
MoveNextDuplicateMultiple Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Cursor.

Cursor::MoveNextDuplicateMultipleKey Method

Cursor Class  See Also
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MoveNextDuplicateMultipleKey()</strong></td>
<td>If the next key/data pair of the database is a duplicate data record for the current key/data pair, move the cursor to the next key/data pair in the database, and store that pair and as many duplicate data items that can fit in a buffer the size of one database page in <code>CurrentMultipleKey</code>. MoveNextDuplicateMultipleKey will return false if the next key/data pair of the database is not a duplicate data record for the current key/data pair. If the next key/data pair of the database is a duplicate data record for the current key/data pair, move the cursor to the next key/data pair in the database, and store that pair and as many duplicate data items that can fit in a buffer the size of one database page in <code>CurrentMultipleKey</code>. MoveNextDuplicateMultipleKey will return false if the next key/data pair of the database is not a duplicate data record for the current key/data pair.</td>
</tr>
<tr>
<td><strong>MoveNextDuplicateMultipleKey(LockingInfo)</strong></td>
<td></td>
</tr>
</tbody>
</table>
MoveNextDuplicateMultipleKey(Int32)

MoveNextDuplicateMultipleKey(Int32, LockingInfo)

pair, move the cursor to the next key/data pair in the database, and store that pair and as many duplicate data items that can fit in a buffer the size of BufferSize in CurrentMultipleKey. MoveNextDuplicateMultipleKey will return false if the next key/data pair of the database is not a duplicate data record for the current key/data pair.

If the next key/data pair of the database is a duplicate data record for the current key/data pair, move the cursor to the next key/data pair in the database, and store that pair and as many duplicate data items that can fit in a buffer the size of BufferSize in CurrentMultipleKey. MoveNextDuplicateMultipleKey will return false if the next key/data pair of the database is not a duplicate data record for the current key/data pair.
See Also

Cursor Class
Cursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If the next key/data pair of the database is a duplicate data record for the current key/data pair, move the cursor to the next key/data pair in the database, and store that pair and as many duplicate data items that can fit in a buffer the size of one database page in CurrentMultipleKey. MoveNextDuplicateMultipleKey will return false if the next key/data pair of the database is not a duplicate data record for the current key/data pair.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

```csharp
public bool MoveNextDuplicateMultipleKey()
```

Visual Basic (Declaration)

```vbnet
Public Function MoveNextDuplicateMultipleKey As Boolean
```

Visual C++

```cpp
public:
bool MoveNextDuplicateMultipleKey()
```

Return Value

True if the cursor was positioned successfully, false otherwise.
See Also

Cursor Class
MoveNextDuplicateMultipleKey Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If the next key/data pair of the database is a duplicate data record for the current key/data pair, move the cursor to the next key/data pair in the database, and store that pair and as many duplicate data items that can fit in a buffer the size of one database page in $\text{CurrentMultipleKey}$. $\text{MoveNextDuplicateMultipleKey}$ will return false if the next key/data pair of the database is not a duplicate data record for the current key/data pair.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
### Syntax

**C#**

```csharp
public bool MoveNextDuplicateMultipleKey(
    LockingInfo info
)
```

**Visual Basic (Declaration)**

```vbnet
Public Function MoveNextDuplicateMultipleKey ( _
    info As LockingInfo _
) As Boolean
```

**Visual C++**

```cpp
public:
    bool MoveNextDuplicateMultipleKey(
        LockingInfo^ info
    )
```

### Parameters

**info**

Type: `BerkeleyDB::LockingInfo`

The locking behavior to use.

### Return Value

True if the cursor was positioned successfully, false otherwise.
See Also

Cursor Class
MoveNextDuplicateMultipleKey Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If the next key/data pair of the database is a duplicate data record for the current key/data pair, move the cursor to the next key/data pair in the database, and store that pair and as many duplicate data items that can fit in a buffer the size of BufferSize in CurrentMultipleKey. MoveNextDuplicateMultipleKey will return false if the next key/data pair of the database is not a duplicate data record for the current key/data pair.

Namespace: BerkeleyDB
### Syntax

**C#**

```csharp
public bool MoveNextDuplicateMultipleKey(
    int BufferSize
)
```

**Visual Basic (Declaration)**

```vbnet
Public Function MoveNextDuplicateMultipleKey ( _
    BufferSize As Integer _
) As Boolean
```

**Visual C++**

```cpp
public:
    bool MoveNextDuplicateMultipleKey(
        int BufferSize
    )
```

### Parameters

**BufferSize**

Type: `System::::Int32`

The size of a buffer to fill with key/data pairs. Must be at least the page size of the underlying database and be a multiple of 1024.

### Return Value

True if the cursor was positioned successfully, false otherwise.
See Also

Cursor Class
MoveNextDuplicateMultipleKey Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If the next key/data pair of the database is a duplicate data record for the current key/data pair, move the cursor to the next key/data pair in the database, and store that pair and as many duplicate data items that can fit in a buffer the size of BufferSize in `CurrentMultipleKey`. `MoveNextDuplicateMultipleKey` will return false if the next key/data pair of the database is not a duplicate data record for the current key/data pair.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
## Syntax

**C#**

```csharp
public bool MoveNextDuplicateMultipleKey(
    int BufferSize,
    LockingInfo info
)
```

**Visual Basic (Declaration)**

```vbnet
Public Function MoveNextDuplicateMultipleKey ( _
    BufferSize As Integer, _
    info As LockingInfo _
) As Boolean
```

**Visual C++**

```cpp
public:
bool MoveNextDuplicateMultipleKey(
    int BufferSize,
    LockingInfo info
)
```

## Parameters

**BufferSize**

Type: `System::::Int32`

The size of a buffer to fill with key/data pairs. Must be at least the page size of the underlying database and be a multiple of 1024.

**info**

Type: `BerkeleyDB::::LockingInfo`

The locking behavior to use.

## Return Value

True if the cursor was positioned successfully, false otherwise.
See Also

Cursor Class
MoveNextDuplicateMultipleKey Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Cursor.

...::

MoveNextMultiple Method

See Also
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MoveNextMultiple()()</td>
<td>If the cursor is not yet initialized, MoveNextMultiple is identical to MoveFirstMultiple(). Otherwise, move the cursor to the next key/data pair of the database, and store that pair and as many duplicate data items that can fit in a buffer the size of one database page in CurrentMultiple. In the presence of duplicate key values, the value of CurrentMultiple.Key may not change.</td>
</tr>
<tr>
<td>MoveNextMultiple(LockingInfo)</td>
<td>If the cursor is not yet initialized, MoveNextMultiple is identical to MoveFirstMultiple(LockingInfo). Otherwise, move the cursor to the next key/data pair of the database, and store that pair and as many duplicate data items that can fit in a buffer the size of one database page in CurrentMultiple. In the presence of duplicate key values, the value of CurrentMultiple.Key may not change.</td>
</tr>
<tr>
<td>MoveNextMultiple(Int32)</td>
<td>If the cursor is not yet initialized, MoveNextMultiple is identical to MoveFirstMultiple(Int32). Otherwise, move the cursor to the next key/data pair of the database, and store that pair and as many duplicate data items that can fit in a buffer the size of BufferSize in CurrentMultiple. In the presence of duplicate key values, the value of CurrentMultiple.Key may not change.</td>
</tr>
</tbody>
</table>
MoveNextMultiple(Int32, LockingInfo)

If the cursor is not yet initialized, MoveNextMultiple is identical to MoveFirstMultiple(Int32, LockingInfo). Otherwise, move the cursor to the next key/data pair of the database, and store that pair and as many duplicate data items that can fit in a buffer the size of BufferSize in CurrentMultiple. In the presence of duplicate key values, the value of CurrentMultiple.Key may not change.
See Also

Cursor Class
Cursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If the cursor is not yet initialized, MoveNextMultiple is identical to MoveFirstMultiple(). Otherwise, move the cursor to the next key/data pair of the database, and store that pair and as many duplicate data items that can fit in a buffer the size of one database page in CurrentMultiple. In the presence of duplicate key values, the value of CurrentMultiple.Key may not change.

Namespace: BerkeleyDB
**Syntax**

**C#**

```csharp
public bool MoveNextMultiple()
```

**Visual Basic (Declaration)**

```vbnet
Public Function MoveNextMultiple As Boolean
```

**Visual C++**

```cpp
public:
bool MoveNextMultiple()
```

**Return Value**

True if the cursor was positioned successfully, false otherwise.
See Also

Cursor Class
MoveNextMultiple Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If the cursor is not yet initialized, MoveNextMultiple is identical to MoveFirstMultiple(LockingInfo). Otherwise, move the cursor to the next key/data pair of the database, and store that pair and as many duplicate data items that can fit in a buffer the size of one database page in CurrentMultiple. In the presence of duplicate key values, the value of CurrentMultiple.Key may not change.

Namespace: BerkeleyDB
### Syntax

**C#**

```csharp
public bool MoveNextMultiple(
    LockingInfo info
)
```

**Visual Basic (Declaration)**

```vbnet
Public Function MoveNextMultiple ( _
    info As LockingInfo _
) As Boolean
```

**Visual C++**

```cpp
public:
bool MoveNextMultiple(
    LockingInfo^ info
)
```

### Parameters

- **info**
  - Type: `BerkeleyDB::LockingInfo`
  - The locking behavior to use.

### Return Value

- True if the cursor was positioned successfully, false otherwise.
See Also

Cursor Class
MoveNextMultiple Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If the cursor is not yet initialized, MoveNextMultiple is identical to MoveFirstMultiple(Int32). Otherwise, move the cursor to the next key/data pair of the database, and store that pair and as many duplicate data items that can fit in a buffer the size of BufferSize in CurrentMultiple. In the presence of duplicate key values, the value of CurrentMultiple.Key may not change.
Syntax

C#

```csharp
public bool MoveNextMultiple(
    int BufferSize
)
```

Visual Basic (Declaration)

```vbnet
Public Function MoveNextMultiple (
    _
    BufferSize As Integer _
) As Boolean
```

Visual C++

```cpp
public:
    bool MoveNextMultiple(
        int BufferSize
    )
```

Parameters

BufferSize
Type: `System::::Int32`  
The size of a buffer to fill with duplicate data items. Must be at least the page size of the underlying database and be a multiple of 1024.

Return Value

True if the cursor was positioned successfully, false otherwise.
See Also

Cursor Class
MoveNextMultiple Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
MoveNextMultiple Method (Int32, LockingInfo)

If the cursor is not yet initialized, MoveNextMultiple is identical to MoveFirstMultiple(Int32, LockingInfo). Otherwise, move the cursor to the next key/data pair of the database, and store that pair and as many duplicate data items that can fit in a buffer the size of BufferSize in CurrentMultiple. In the presence of duplicate key values, the value of CurrentMultiple.Key may not change.

Namespace: BerkeleyDB
**Syntax**

**C#**

```csharp
public bool MoveNextMultiple(
    int BufferSize,
    LockingInfo info
)
```

**Visual Basic (Declaration)**

```vbnet
Public Function MoveNextMultiple ( _
    BufferSize As Integer, _
    info As LockingInfo _
) As Boolean
```

**Visual C++**

```c++
public:
bool MoveNextMultiple(
    int BufferSize,
    LockingInfo info
)
```

**Parameters**

**BufferSize**

Type: `System::Int32`

The size of a buffer to fill with duplicate data items. Must be at least the page size of the underlying database and be a multiple of 1024.

**info**

Type: `BerkeleyDB::LockingInfo`

The locking behavior to use.

**Return Value**

True if the cursor was positioned successfully, false otherwise.
See Also

Cursor Class
MoveNextMultiple Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Cursor...::MoveNextMultipleKey Method

Cursor Class  See Also
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MoveNextMultipleKey()()</strong></td>
<td>If the cursor is not yet initialized, MoveNextMultipleKey is identical to MoveFirstMultipleKey(). Otherwise, move the cursor to the next key/data pair of the database, and store that pair and as many ensuing key/data pairs that can fit in a buffer the size of one database page in CurrentMultipleKey. In the presence of duplicate key values, the keys of CurrentMultipleKey may not change.</td>
</tr>
<tr>
<td><strong>MoveNextMultipleKey(LockingInfo)</strong></td>
<td>If the cursor is not yet initialized, MoveNextMultipleKey is identical to MoveFirstMultipleKey(LockingInfo). Otherwise, move the cursor to the next key/data pair of the database, and store that pair and as many ensuing key/data pairs that can fit in a buffer the size of one database page in CurrentMultipleKey. In the presence of duplicate key values, the keys of CurrentMultipleKey may not change.</td>
</tr>
<tr>
<td><strong>MoveNextMultipleKey(Int32)</strong></td>
<td>If the cursor is not yet initialized, MoveNextMultipleKey is identical to MoveFirstMultipleKey(Int32). Otherwise, move the cursor to the next key/data pair of the database, and store that pair and as many ensuing key/data pairs that can fit in a buffer the size of BufferSize in CurrentMultipleKey. In the presence</td>
</tr>
</tbody>
</table>
of duplicate key values, the keys of \texttt{CurrentMultipleKey} may not change. If the cursor is not yet initialized, \texttt{MoveNextMultipleKey} is identical to \texttt{MoveFirstMultipleKey(Int32, LockingInfo)}. Otherwise, move the cursor to the next key/data pair of the database, and store that pair and as many ensuing key/data pairs that can fit in a buffer the size of BufferSize in \texttt{CurrentMultipleKey}. In the presence of duplicate key values, the keys of \texttt{CurrentMultipleKey} may not change.
See Also

Cursor Class
Cursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Cursor Class  See Also

If the cursor is not yet initialized, MoveNextMultipleKey is identical to MoveFirstMultipleKey(). Otherwise, move the cursor to the next key/data pair of the database, and store that pair and as many ensuing key/data pairs that can fit in a buffer the size of one database page in CurrentMultipleKey. In the presence of duplicate key values, the keys of CurrentMultipleKey may not change.

Namespace: BerkeleyDB
Syntax

C#

public bool MoveNextMultipleKey()

Visual Basic (Declaration)

Public Function MoveNextMultipleKey As Boolean

Visual C++

public:
bool MoveNextMultipleKey()

Return Value

True if the cursor was positioned successfully, false otherwise.
See Also

Cursor Class
MoveNextMultipleKey Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If the cursor is not yet initialized, MoveNextMultipleKey is identical to MoveFirstMultipleKey(LockingInfo). Otherwise, move the cursor to the next key/data pair of the database, and store that pair and as many ensuing key/data pairs that can fit in a buffer the size of one database page in CurrentMultipleKey. In the presence of duplicate key values, the keys of CurrentMultipleKey may not change.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

```csharp
public bool MoveNextMultipleKey(
    LockingInfo info
)
```

Visual Basic (Declaration)

```vbnet
Public Function MoveNextMultipleKey ( _
    info As LockingInfo _
) As Boolean
```

Visual C++

```cpp
public:
bool MoveNextMultipleKey(
    LockingInfo^ info
)
```

Parameters

info
  Type: BerkeleyDB::::LockingInfo
  The locking behavior to use.

Return Value

True if the cursor was positioned successfully, false otherwise.
See Also

Cursor Class
MoveNextMultipleKey Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
MoveNextMultipleKey Method (Int32)

If the cursor is not yet initialized, MoveNextMultipleKey is identical to MoveFirstMultipleKey(Int32). Otherwise, move the cursor to the next key/data pair of the database, and store that pair and as many ensuing key/data pairs that can fit in a buffer the size of BufferSize in CurrentMultipleKey. In the presence of duplicate key values, the keys of CurrentMultipleKey may not change.

Namespace: BerkeleyDB
Syntax

C#

```csharp
public bool MoveNextMultipleKey(
    int BufferSize
)
```

Visual Basic (Declaration)

```vbnet
Public Function MoveNextMultipleKey (_
    BufferSize As Integer _
) As Boolean
```

Visual C++

```cpp
public:
    bool MoveNextMultipleKey(
        int BufferSize
    )
```

Parameters

BufferSize

Type: System::::Int32

The size of a buffer to fill with key/data pairs. Must be at least the page size of the underlying database and be a multiple of 1024.

Return Value

True if the cursor was positioned successfully, false otherwise.
See Also

Cursor Class
MoveNextMultipleKey Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If the cursor is not yet initialized, MoveNextMultipleKey is identical to MoveFirstMultipleKey(Int32, LockingInfo). Otherwise, move the cursor to the next key/data pair of the database, and store that pair and as many ensuing key/data pairs that can fit in a buffer the size of BufferSize in CurrentMultipleKey. In the presence of duplicate key values, the keys of CurrentMultipleKey may not change.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

```csharp
public bool MoveNextMultipleKey(
    int BufferSize,
    LockingInfo info
)
```

Visual Basic (Declaration)

```vbnet
Public Function MoveNextMultipleKey ( _
    BufferSize As Integer, _
    info As LockingInfo _
) As Boolean
```

Visual C++

```cpp
public:
bool MoveNextMultipleKey(
    int BufferSize,
    LockingInfo^ info
)
```

Parameters

BufferSize
Type: `System::::Int32`
The size of a buffer to fill with key/data pairs. Must be at least the page size of the underlying database and be a multiple of 1024.

info
Type: `BerkeleyDB::::LockingInfo`
The locking behavior to use.

Return Value

True if the cursor was positioned successfully, false otherwise.
See Also

Cursor Class
MoveNextMultipleKey Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
C#  Visual Basic
Visual C++
Include Protected Members
Include Inherited Members
Berkeley DB .NET API Documentation
Cursor::MoveNextUnique Method
Cursor Class  See Also
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MoveNextUnique()()</td>
<td>If the cursor is not yet initialized, MoveNextUnique is identical to MoveFirst(). Otherwise, move the cursor to the next non-duplicate key in the database, and store that key and associated datum in Current. MoveNextUnique will return false if no non-duplicate key/data pairs exist after the cursor position in the database.</td>
</tr>
<tr>
<td>MoveNextUnique(LockingInfo)</td>
<td>If the cursor is not yet initialized, MoveNextUnique is identical to MoveFirst(LockingInfo). Otherwise, move the cursor to the next non-duplicate key in the database, and store that key and associated datum in Current. MoveNextUnique will return false if no non-duplicate key/data pairs exist after the cursor position in the database.</td>
</tr>
</tbody>
</table>
See Also

Cursor Class
Cursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If the cursor is not yet initialized, MoveNextUnique is identical to MoveFirst(). Otherwise, move the cursor to the next non-duplicate key in the database, and store that key and associated datum in Current. MoveNextUnique will return false if no non-duplicate key/data pairs exist after the cursor position in the database.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool MoveNextUnique()

Visual Basic (Declaration)

Public Function MoveNextUnique As Boolean

Visual C++

public:
bool MoveNextUnique()

Return Value

True if the cursor was positioned successfully, false otherwise.
Remarks

If positioning the cursor fails, Current will contain an empty KeyValuePair(Of (TKey, TValue)>).
See Also

Cursor Class
MoveNextUnique Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
MoveNextUnique Method (LockingInfo)

If the cursor is not yet initialized, MoveNextUnique is identical to MoveFirst(LockingInfo). Otherwise, move the cursor to the next non-duplicate key in the database, and store that key and associated datum in Current. MoveNextUnique will return false if no non-duplicate key/data pairs exist after the cursor position in the database.

Namespace: BerkeleyDB
Syntax

C#

public bool MoveNextUnique(
    LockingInfo info
)

Visual Basic (Declaration)

Public Function MoveNextUnique ( _
    info As LockingInfo _
) As Boolean

Visual C++

public:
    bool MoveNextUnique(  
        LockingInfo^ info
    )

Parameters

info
   Type: BerkeleyDB::LockingInfo
   The locking behavior to use.

Return Value

True if the cursor was positioned successfully, false otherwise.
Remarks

If the database is a Queue or Recno database, MoveNextUnique will ignore any keys that exist but were never explicitly created by the application, or those that were created and later deleted.

If positioning the cursor fails, Current will contain an empty KeyValuePair(Of (TKey, TValue)>).
See Also

Cursor Class
MoveNextUnique Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Cursor Class  See Also
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MoveNextUniqueMultiple()</strong></td>
<td>If the cursor is not yet initialized, <code>MoveNextUniqueMultiple</code> is identical to <code>MoveFirstMultiple()</code>. Otherwise, move the cursor to the next non-duplicate key in the database, and store that key and associated datum and as many duplicate data items that can fit in a buffer the size of one database page in <code>CurrentMultiple</code>. <code>MoveNextUniqueMultiple</code> will return false if no non-duplicate key/data pairs exist after the cursor position in the database. If the cursor is not yet initialized, <code>MoveNextUniqueMultiple</code> is identical to <code>MoveFirstMultiple(LockingInfo)</code>. Otherwise, move the cursor to the next non-duplicate key in the database, and store that key and associated datum and as many duplicate data items that can fit in a buffer the size of one database page in <code>CurrentMultiple</code>. <code>MoveNextUniqueMultiple</code> will return false if no non-duplicate key/data pairs exist after the cursor position in the database.</td>
</tr>
<tr>
<td><strong>MoveNextUniqueMultiple(LockingInfo)</strong></td>
<td>If the cursor is not yet initialized, <code>MoveNextUniqueMultiple</code> is identical to <code>MoveFirstMultiple(LockingInfo)</code>. Otherwise, move the cursor to the next non-duplicate key in the database, and store that key and associated datum and as many duplicate data items that can fit in a buffer the size of one database page in <code>CurrentMultiple</code>. <code>MoveNextUniqueMultiple</code> will return false if no non-duplicate key/data pairs exist after the cursor position in the database. If the cursor is not yet initialized, <code>MoveNextUniqueMultiple</code> is identical to</td>
</tr>
</tbody>
</table>
MoveNextUniqueMultiple(Int32)

MoveNextUniqueMultiple(Int32)

Otherwise, move the cursor to the next non-duplicate key in the database, and store that key and associated datum and as many duplicate data items that can fit in a buffer the size of BufferSize in CurrentMultiple.

MoveNextUniqueMultiple will return false if no non-duplicate key/data pairs exist after the cursor position in the database.

If the cursor is not yet initialized, MoveNextUniqueMultiple is identical to MoveFirstMultiple(Int32, LockingInfo). Otherwise, move the cursor to the next non-duplicate key in the database, and store that key and associated datum and as many duplicate data items that can fit in a buffer the size of BufferSize in CurrentMultiple.

MoveNextUniqueMultiple will return false if no non-duplicate key/data pairs exist after the cursor position in the database.
See Also

Cursor Class
Cursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If the cursor is not yet initialized, MoveNextUniqueMultiple is identical to MoveFirstMultiple(). Otherwise, move the cursor to the next non-duplicate key in the database, and store that key and associated datum and as many duplicate data items that can fit in a buffer the size of one database page in CurrentMultiple. MoveNextUniqueMultiple will return false if no non-duplicate key/data pairs exist after the cursor position in the database.

Namespace: BerkeleyDB
Syntax

C#

public bool MoveNextUniqueMultiple()

Visual Basic (Declaration)

Public Function MoveNextUniqueMultiple As Boolean

Visual C++

public:
bool MoveNextUniqueMultiple()

Return Value

True if the cursor was positioned successfully, false otherwise.
See Also

Cursor Class
MoveNextUniqueMultiple Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If the cursor is not yet initialized, `MoveNextUniqueMultiple` is identical to `MoveFirstMultiple(LockingInfo)`. Otherwise, move the cursor to the next non-duplicate key in the database, and store that key and associated datum and as many duplicate data items that can fit in a buffer the size of one database page in `CurrentMultiple`. `MoveNextUniqueMultiple` will return false if no non-duplicate key/data pairs exist after the cursor position in the database.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

**C#**

```csharp
public bool MoveNextUniqueMultiple(
   LockingInfo info
)
```

**Visual Basic (Declaration)**

```vbnet
Public Function MoveNextUniqueMultiple (_
   info As LockingInfo _
) As Boolean
```

**Visual C++**

```cpp
public:
   bool MoveNextUniqueMultiple(
      LockingInfo^ info
   )
```

**Parameters**

info

Type: `BerkeleyDB::::LockingInfo`

The locking behavior to use.

**Return Value**

True if the cursor was positioned successfully, false otherwise.
See Also

Cursor Class
MoveNextUniqueMultiple Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Cursor...::MoveNextUniqueMultiple Method (Int32)

Cursor Class  See Also

If the cursor is not yet initialized, MoveNextUniqueMultiple is identical to MoveFirstMultiple(Int32). Otherwise, move the cursor to the next non-duplicate key in the database, and store that key and associated datum and as many duplicate data items that can fit in a buffer the size of BufferSize in CurrentMultiple. MoveNextUniqueMultiple will return false if no non-duplicate key/data pairs exist after the cursor position in the database.

Namespace:  BerkeleyDB
Syntax

C#

public bool MoveNextUniqueMultiple(
    int BufferSize
)

Visual Basic (Declaration)

Public Function MoveNextUniqueMultiple ( _
    BufferSize As Integer _
) As Boolean

Visual C++

public:
bool MoveNextUniqueMultiple(
    int BufferSize
)

Parameters

BufferSize
    Type: System::::Int32
    The size of a buffer to fill with duplicate data items. Must be at least the page size of the underlying database and be a multiple of 1024.

Return Value

True if the cursor was positioned successfully, false otherwise.
See Also

Cursor Class
MoveNextUniqueMultiple Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If the cursor is not yet initialized, MoveNextUniqueMultiple is identical to MoveFirstUniqueMultiple(Int32, LockingInfo). Otherwise, move the cursor to the next non-duplicate key in the database, and store that key and associated datum and as many duplicate data items that can fit in a buffer the size of BufferSize in CurrentMultiple. MoveNextUniqueMultiple will return false if no non-duplicate key/data pairs exist after the cursor position in the database.
Syntax

**C#**

```csharp
public bool MoveNextUniqueMultiple(
    int BufferSize,
    LockingInfo info
)
```

**Visual Basic (Declaration)**

```vbnet
Public Function MoveNextUniqueMultiple ( _
    BufferSize As Integer, _
    info As LockingInfo _
) As Boolean
```

**Visual C++**

```cpp
public:
    bool MoveNextUniqueMultiple( 
    int BufferSize, 
    LockingInfo^ info
)
```

**Parameters**

BufferSize
Type: `System::::Int32`
The size of a buffer to fill with duplicate data items. Must be at least the page size of the underlying database and be a multiple of 1024.

info
Type: `BerkeleyDB::::LockingInfo`
The locking behavior to use.

**Return Value**

True if the cursor was positioned successfully, false otherwise.
See Also

Cursor Class
MoveNextUniqueMultiple Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Cursor...MoveNextUniqueMultipleKey Method

Cursor Class  See Also
### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MoveNextUniqueMultipleKey()</strong></td>
<td>If the cursor is not yet initialized, <code>MoveNextUniqueMultipleKey</code> is identical to <code>MoveFirstMultipleKey()</code>. Otherwise, move the cursor to the next non-duplicate key in the database, and store that key and associated datum and as many ensuing key/data pairs that can fit in a buffer the size of one database page in <code>CurrentMultipleKey</code>. <code>MoveNextUniqueMultipleKey</code> will return false if no non-duplicate key/data pairs exist after the cursor position in the database. If the cursor is not yet initialized, <code>MoveNextUniqueMultipleKey</code> is identical to <code>MoveFirstMultipleKey(LockingInfo)</code>. Otherwise, move the cursor to the next non-duplicate key in the database, and store that key and associated datum and as many ensuing key/data pairs that can fit in a buffer the size of one database page in <code>CurrentMultipleKey</code>. <code>MoveNextUniqueMultipleKey</code> will return false if no non-duplicate key/data pairs exist after the cursor position in the database.</td>
</tr>
<tr>
<td><strong>MoveNextUniqueMultipleKey(LockingInfo)</strong></td>
<td></td>
</tr>
</tbody>
</table>
MoveNextUniqueMultipleKey(Int32)

Otherwise, move the cursor to the next non-duplicate key in the database, and store that key and associated datum and as many ensuing key/data pairs that can fit in a buffer the size of BufferSize in CurrentMultipleKey.

MoveNextUniqueMultipleKey() will return false if no non-duplicate key/data pairs exist after the cursor position in the database.

If the cursor is not yet initialized, MoveNextUniqueMultipleKey is identical to MoveFirstMultipleKey(Int32, LockingInfo). Otherwise, move the cursor to the next non-duplicate key in the database, and store that key and associated datum and as many ensuing key/data pairs that can fit in a buffer the size of BufferSize in CurrentMultipleKey.

MoveNextUniqueMultipleKey() will return false if no non-duplicate key/data pairs exist after the cursor position in the database.
See Also

Cursor Class
Cursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If the cursor is not yet initialized, `MoveNextUniqueMultipleKey` is identical to `MoveFirstMultipleKey()`. Otherwise, move the cursor to the next non-duplicate key in the database, and store that key and associated datum and as many ensuing key/data pairs that can fit in a buffer the size of one database page in `CurrentMultipleKey`. `MoveNextUniqueMultipleKey` will return false if no non-duplicate key/data pairs exist after the cursor position in the database.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool MoveNextUniqueMultipleKey()

Visual Basic (Declaration)

Public Function MoveNextUniqueMultipleKey As Boolean

Visual C++

public:
bool MoveNextUniqueMultipleKey()

Return Value

True if the cursor was positioned successfully, false otherwise.
See Also

Cursor Class
MoveNextUniqueMultipleKey Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If the cursor is not yet initialized, MoveNextUniqueMultipleKey is identical to MoveFirstMultipleKey(LockingInfo). Otherwise, move the cursor to the next non-duplicate key in the database, and store that key and associated datum and as many ensuing key/data pairs that can fit in a buffer the size of one database page in CurrentMultipleKey. MoveNextUniqueMultipleKey will return false if no non-duplicate key/data pairs exist after the cursor position in the database.
Syntax

C#

```csharp
public bool MoveNextUniqueMultipleKey(
   LockingInfo info
)
```

Visual Basic (Declaration)

```vbnet
Public Function MoveNextUniqueMultipleKey ( _
   info As LockingInfo _
) As Boolean
```

Visual C++

```cpp
public:
   bool MoveNextUniqueMultipleKey(
       LockingInfo^ info
   )
```

Parameters

info
Type: `BerkeleyDB::::LockingInfo`
The locking behavior to use.

Return Value

True if the cursor was positioned successfully, false otherwise.
See Also

Cursor Class
MoveNextUniqueMultipleKey Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If the cursor is not yet initialized, MoveNextUniqueMultipleKey is identical to
MoveFirstMultipleKey(Int32). Otherwise, move the cursor to the next non-
duplicate key in the database, and store that key and associated datum and as
many ensuing key/data pairs that can fit in a buffer the size of BufferSize in
CurrentMultipleKey. MoveNextUniqueMultipleKey will return false if no non-
duplicate key/data pairs exist after the cursor position in the database.
Syntax

C#

```csharp
public bool MoveNextUniqueMultipleKey(
    int BufferSize
)
```

Visual Basic (Declaration)

```vbnet
Public Function MoveNextUniqueMultipleKey ( _
    BufferSize As Integer _
) As Boolean
```

Visual C++

```cpp
public:
bool MoveNextUniqueMultipleKey(
    int BufferSize
)
```

Parameters

BufferSize

Type: `System::::Int32`

The size of a buffer to fill with key/data pairs. Must be at least the page size of the underlying database and be a multiple of 1024.

Return Value

True if the cursor was positioned successfully, false otherwise.
See Also

Cursor Class
MoveNextUniqueMultipleKey Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If the cursor is not yet initialized, MoveNextUniqueMultipleKey is identical to MoveFirstMultipleKey(Int32, LockingInfo). Otherwise, move the cursor to the next non-duplicate key in the database, and store that key and associated datum and as many ensuing key/data pairs that can fit in a buffer the size of BufferSize in CurrentMultipleKey. MoveNextUniqueMultipleKey will return false if no non-duplicate key/data pairs exist after the cursor position in the database.
Syntax

**C#**

```csharp
public bool MoveNextUniqueMultipleKey(
    int BufferSize,
    LockingInfo info
)
```

**Visual Basic (Declaration)**

```vbnet
Public Function MoveNextUniqueMultipleKey ( _
    BufferSize As Integer, _
    info As LockingInfo _
) As Boolean
```

**Visual C++**

```cpp
public:
    bool MoveNextUniqueMultipleKey(
        int BufferSize,
        LockingInfo^ info
    )
```

**Parameters**

**BufferSize**
Type: `System::::Int32`
The size of a buffer to fill with key/data pairs. Must be at least the page size of the underlying database and be a multiple of 1024.

**info**
Type: `BerkeleyDB::::LockingInfo`
The locking behavior to use.

**Return Value**

True if the cursor was positioned successfully, false otherwise.
See Also

Cursor Class
MoveNextUniqueMultipleKey Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MovePrev()()</td>
<td>If the cursor is not yet initialized, MovePrev is identical to MoveLast(). Otherwise, move the cursor to the previous key/data pair of the database, and store that pair in Current. In the presence of duplicate key values, the value of Current.Key may not change.</td>
</tr>
<tr>
<td>MovePrev(LockingInfo)</td>
<td>If the cursor is not yet initialized, MovePrev is identical to MoveLast(LockingInfo). Otherwise, move the cursor to the previous key/data pair of the database, and store that pair in Current. In the presence of duplicate key values, the value of Current.Key may not change.</td>
</tr>
</tbody>
</table>
See Also

Cursor Class
Cursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If the cursor is not yet initialized, MovePrev is identical to MoveLast(). Otherwise, move the cursor to the previous key/data pair of the database, and store that pair in Current. In the presence of duplicate key values, the value of Current.Key may not change.

Namespace: BerkeleyDB
Syntax

**C#**

```csharp
public bool MovePrev()
```

**Visual Basic (Declaration)**

```vbnet
Public Function MovePrev As Boolean
```

**Visual C++**

```cpp
public:
bool MovePrev()
```

**Return Value**

True if the cursor was positioned successfully, false otherwise.
Remarks

If positioning the cursor fails, Current will contain an empty KeyValuePair(Of (TKey, TValue)>).
See Also

Cursor Class
MovePrev Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If the cursor is not yet initialized, MovePrev is identical to MoveLast(LockingInfo). Otherwise, move the cursor to the previous key/data pair of the database, and store that pair in Current. In the presence of duplicate key values, the value of Current.Key may not change.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

```csharp
public bool MovePrev(
    LockingInfo info
)
```

Visual Basic (Declaration)

```vbnet
Public Function MovePrev (_
    info As LockingInfo _
) As Boolean
```

Visual C++

```cpp
public:
bool MovePrev(
    LockingInfo^ info
)
```

Parameters

info

Type: BerkeleyDB::::LockingInfo
The locking behavior to use.

Return Value

True if the cursor was positioned successfully, false otherwise.
Remarks

If positioning the cursor fails, `Current` will contain an empty `KeyValuePair<(TKey, TValue)>`. 
See Also

Cursor Class
MovePrev Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation

Cursor...:MovePrevDuplicate Method

Cursor Class  See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MovePrevDuplicate()</td>
<td>If the previous key/data pair of the database is a duplicate data record for the current key/data pair, the cursor is moved to the previous key/data pair of the database, and that pair is stored in Current. MovePrevDuplicate will return false if the previous key/data pair of the database is not a duplicate data record for the current key/data pair.</td>
</tr>
<tr>
<td>MovePrevDuplicate(LockingInfo)</td>
<td>If the previous key/data pair of the database is a duplicate data record for the current key/data pair, the cursor is moved to the previous key/data pair of the database, and that pair is stored in Current. MovePrevDuplicate will return false if the previous key/data pair of the database is not a duplicate data record for the current key/data pair.</td>
</tr>
</tbody>
</table>
See Also

Cursor Class
Cursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Cursor Class  See Also

If the previous key/data pair of the database is a duplicate data record for the current key/data pair, the cursor is moved to the previous key/data pair of the database, and that pair is stored in Current. MovePrevDuplicate will return false if the previous key/data pair of the database is not a duplicate data record for the current key/data pair.

Namespace: BerkeleyDB
Syntax

**C#**

```csharp
public bool MovePrevDuplicate()
```

**Visual Basic (Declaration)**

```vbnet
Public Function MovePrevDuplicate As Boolean
```

**Visual C++**

```cpp
public:
bool MovePrevDuplicate()
```

**Return Value**

True if the cursor was positioned successfully, false otherwise.
Remarks

If positioning the cursor fails, `Current` will contain an empty `KeyValuePair<Of <(TKey, TValue)>)>.`
See Also

Cursor Class
MovePrevDuplicate Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Cursor Class  See Also

If the previous key/data pair of the database is a duplicate data record for the current key/data pair, the cursor is moved to the previous key/data pair of the database, and that pair is stored in Current. MovePrevDuplicate will return false if the previous key/data pair of the database is not a duplicate data record for the current key/data pair.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

```csharp
public bool MovePrevDuplicate(
    LockingInfo info
)
```

Visual Basic (Declaration)

```vbscript
Public Function MovePrevDuplicate ( _
    info As LockingInfo _
) As Boolean
```

Visual C++

```cpp
public:
bool MovePrevDuplicate( 
    LockingInfo^ info
)
```

Parameters

info

Type: BerkeleyDB::LockingInfo

The locking behavior to use.

Return Value

True if the cursor was positioned successfully, false otherwise.
Remarks

If positioning the cursor fails, `Current` will contain an empty `KeyValuePair<Of `<(TKey, TValue)>`).
See Also

Cursor Class
MovePrevDuplicate Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Cursor.

Cursor Class  See Also
# Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MovePrevUnique</td>
<td>If the cursor is not yet initialized, MovePrevUnique is identical to MoveLast(). Otherwise, move the cursor to the previous non-duplicate key in the database, and store that key and associated datum in Current. MovePrevUnique will return false if no non-duplicate key/data pairs exist after the cursor position in the database.</td>
</tr>
<tr>
<td>MovePrevUnique(LockingInfo)</td>
<td>If the cursor is not yet initialized, MovePrevUnique is identical to MoveLast(LockingInfo). Otherwise, move the cursor to the previous non-duplicate key in the database, and store that key and associated datum in Current. MovePrevUnique will return false if no non-duplicate key/data pairs exist after the cursor position in the database.</td>
</tr>
</tbody>
</table>
See Also

Cursor Class
Cursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Cursor Class

If the cursor is not yet initialized, MovePrevUnique is identical to `MoveLast()`(). Otherwise, move the cursor to the previous non-duplicate key in the database, and store that key and associated datum in `Current`. MovePrevUnique will return false if no non-duplicate key/data pairs exist after the cursor position in the database.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
**Syntax**

**C#**

public bool MovePrevUnique()

**Visual Basic (Declaration)**

Public Function MovePrevUnique As Boolean

**Visual C++**

public:

bool MovePrevUnique()

**Return Value**

True if the cursor was positioned successfully, false otherwise.
Remarks

If positioning the cursor fails, Current will contain an empty KeyValuePair<(Of <(TKey, TValue)>).
See Also

Cursor Class
MovePrevUnique Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If the cursor is not yet initialized, MovePrevUnique is identical to MoveLast(LockingInfo). Otherwise, move the cursor to the previous non-duplicate key in the database, and store that key and associated datum in Current. MovePrevUnique will return false if no non-duplicate key/data pairs exist after the cursor position in the database.

Namespace: BerkeleyDB
Syntax

C#  

public bool MovePrevUnique(  
        LockingInfo info
    )

Visual Basic (Declaration)

Public Function MovePrevUnique ( _  
        info As LockingInfo _
    ) As Boolean

Visual C++

public:  
        bool MovePrevUnique(  
        LockingInfo^ info
    )

Parameters

info  
        Type: BerkeleyDB::LockingInfo  
        The locking behavior to use.

Return Value

True if the cursor was positioned successfully, false otherwise.
Remarks

If positioning the cursor fails, Current will contain an empty KeyValuePair<(Of <(TKey, TValue)>)).
See Also

Cursor Class
MovePrevUnique Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Overwrite the data of the key/data pair to which the cursor refers with the specified data item.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public void Overwrite(
        DatabaseEntry data
)

Visual Basic (Declaration)

Public Sub Overwrite ( _
        data As DatabaseEntry _
)

Visual C++

public:
void Overwrite(
        DatabaseEntry^ data
)

Parameters

data
    Type: BerkeleyDB...;DatabaseEntry
See Also

Cursor Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
C#  Visual Basic  Visual C++  Include Protected Members  Include Inherited Members
Berkeley DB .NET API Documentation
Cursor...::Refresh Method
Cursor Class  See Also
# Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Refresh()()</td>
<td>Store the key/data pair to which the cursor refers in <strong>Current</strong>.</td>
</tr>
<tr>
<td>Refresh(LockingInfo)</td>
<td>Store the key/data pair to which the cursor refers in <strong>Current</strong>.</td>
</tr>
</tbody>
</table>
See Also

Cursor Class
Cursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Store the key/data pair to which the cursor refers in **Current**.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool Refresh()

Visual Basic (Declaration)

Public Function Refresh As Boolean

Visual C++

public:
bool Refresh()

Return Value

True if the cursor was positioned successfully, false otherwise.
**Remarks**

If positioning the cursor fails, `Current` will contain an empty `KeyValuePair<(Of <(TKey, TValue)>)>`. 
See Also

Cursor Class
Refresh Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Store the key/data pair to which the cursor refers in `Current`.

**Namespace:**  [BerkeleyDB](https://www.berkeleydb.com/)

**Assembly:**  `libdb_dotnet48` (in `libdb_dotnet48.dll`) Version: 4.8.24.0
Syntax

C#

```csharp
public bool Refresh(
    LockingInfo info
)
```

Visual Basic (Declaration)

```vbnet
Public Function Refresh (_
    info As LockingInfo _
) As Boolean
```

Visual C++

```cpp
public:
bool Refresh(
    LockingInfo^ info
)
```

Parameters

info

Type: BerkeleyDB::LockingInfo
The locking behavior to use.

Return Value

True if the cursor was positioned successfully, false otherwise.
Remarks

If positioning the cursor fails, Current will contain an empty KeyValuePair(Of TKey, TValue>).
See Also

Cursor Class
Refresh Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Cursor Class
See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RefreshMultiple()()()</td>
<td>Store the key/data pair to which the cursor refers and as many duplicate data items that can fit in a buffer the size of one database page in <strong>CurrentMultiple</strong>.</td>
</tr>
<tr>
<td>RefreshMultiple(LockingInfo)</td>
<td>Store the key/data pair to which the cursor refers and as many duplicate data items that can fit in a buffer the size of <strong>CurrentMultiple</strong>.</td>
</tr>
<tr>
<td>RefreshMultiple(Int32)</td>
<td>Store the key/data pair to which the cursor refers and as many duplicate data items that can fit in a buffer the size of <strong>BufferSize in CurrentMultiple</strong>.</td>
</tr>
<tr>
<td>RefreshMultiple(Int32, LockingInfo)</td>
<td>Store the key/data pair to which the cursor refers and as many duplicate data items that can fit in a buffer the size of <strong>BufferSize in CurrentMultiple</strong>.</td>
</tr>
</tbody>
</table>
See Also

Cursor Class
Cursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Cursor Class  See Also

Store the key/data pair to which the cursor refers and as many duplicate data items that can fit in a buffer the size of one database page in CurrentMultiple.

Namespace: BerkeleyDB
Syntax

**C#**

```csharp
public bool RefreshMultiple()
```

**Visual Basic (Declaration)**

```vbnet
Public Function RefreshMultiple As Boolean
```

**Visual C++**

```cpp
public:
bool RefreshMultiple()
```

**Return Value**

True if the cursor was positioned successfully, false otherwise.
See Also

Cursor Class
RefreshMultiple Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Store the key/data pair to which the cursor refers and as many duplicate data items that can fit in a buffer the size of one database page in `CurrentMultiple`.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
### Syntax

**C#**

```csharp
public bool RefreshMultiple(
    LockingInfo info
)
```

**Visual Basic (Declaration)**

```vbnet
Public Function RefreshMultiple ( _
    info As LockingInfo _
) As Boolean
```

**Visual C++**

```cpp
public:
    bool RefreshMultiple(
        LockingInfo^ info
    )
```

### Parameters

**info**

Type: `BerkeleyDB::LockingInfo`

The locking behavior to use.

### Return Value

True if the cursor was positioned successfully, false otherwise.
See Also

Cursor Class
RefreshMultiple Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
See Also:

Store the key/data pair to which the cursor refers and as many duplicate data items that can fit in a buffer the size of BufferSize in `CurrentMultiple`.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool RefreshMultiple(
    int BufferSize
)

Visual Basic (Declaration)

Public Function RefreshMultiple ( _
    BufferSize As Integer _
) As Boolean

Visual C++

public:
    bool RefreshMultiple(
        int BufferSize
    )

Parameters

BufferSize
    Type: System::::Int32
    The size of a buffer to fill with duplicate data items. Must be at least the page size of the underlying database and be a multiple of 1024.

Return Value

True if the cursor was positioned successfully, false otherwise.
See Also

Cursor Class
RefreshMultiple Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Cursor...::RefreshMultiple Method (Int32, LockingInfo)

See Also

Store the key/data pair to which the cursor refers and as many duplicate data items that can fit in a buffer the size of BufferSize in CurrentMultiple.

Namespace: BerkeleyDB

Syntax

C#

```csharp
public bool RefreshMultiple(
    int BufferSize,
    LockingInfo info
)
```

Visual Basic (Declaration)

```vbnet
Public Function RefreshMultiple ( _
    BufferSize As Integer, _
    info As LockingInfo _
) As Boolean
```

Visual C++

```cpp
public:
    bool RefreshMultiple(
        int BufferSize,
        LockingInfo^ info
    )
```

Parameters

BufferSize
Type: `System::Int32`
The size of a buffer to fill with duplicate data items. Must be at least the page size of the underlying database and be a multiple of 1024.

info
Type: `BerkeleyDB::LockingInfo` The locking behavior to use.

Return Value

True if the cursor was positioned successfully, false otherwise.
See Also

Cursor Class
RefreshMultiple Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Cursor.

::

RefreshMultipleKey Method

See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>RefreshMultipleKey()</strong></td>
<td>Store the key/data pair to which the cursor refers and as many ensuing key/data pairs that can fit in a buffer the size of one database page in CurrentMultipleKey.</td>
</tr>
<tr>
<td><strong>RefreshMultipleKey(LockingInfo)</strong></td>
<td>Store the key/data pair to which the cursor refers and as many ensuing key/data pairs that can fit in a buffer the size of one database page in CurrentMultipleKey.</td>
</tr>
<tr>
<td><strong>RefreshMultipleKey(Int32)</strong></td>
<td>Store the key/data pair to which the cursor refers and as many ensuing key/data pairs that can fit in a buffer the size of BufferSize in CurrentMultipleKey.</td>
</tr>
<tr>
<td><strong>RefreshMultipleKey(Int32, LockingInfo)</strong></td>
<td>Store the key/data pair to which the cursor refers and as many ensuing key/data pairs that can fit in a buffer the size of BufferSize in CurrentMultipleKey.</td>
</tr>
</tbody>
</table>
See Also

Cursor Class
Cursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Cursor Class  See Also

Store the key/data pair to which the cursor refers and as many ensuing key/data pairs that can fit in a buffer the size of one database page in CurrentMultipleKey.

Namespace: BerkeleyDB
Syntax

C#  
public bool RefreshMultipleKey()

Visual Basic (Declaration)  
Public Function RefreshMultipleKey As Boolean

Visual C++  
public:
bool RefreshMultipleKey()

Return Value

True if the cursor was positioned successfully, false otherwise.
See Also

Cursor Class
RefreshMultipleKey Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
RefreshMultipleKey Method (LockingInfo)

Cursor Class  See Also

Store the key/data pair to which the cursor refers and as many ensuing key/data pairs that can fit in a buffer the size of one database page in CurrentMultipleKey.

Namespace:  BerkeleyDB
Syntax

C#

```csharp
public bool RefreshMultipleKey(
    LockingInfo info
)
```

Visual Basic (Declaration)

```vbnet
Public Function RefreshMultipleKey ( _
    info As LockingInfo _
) As Boolean
```

Visual C++

```cpp
public:
bool RefreshMultipleKey(
    LockingInfo^ info
)
```

Parameters

info

Type: BerkeleyDB:::::LockingInfo
The locking behavior to use.

Return Value

True if the cursor was positioned successfully, false otherwise.
See Also

Cursor Class
RefreshMultipleKey Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Store the key/data pair to which the cursor refers and as many ensuing key/data pairs that can fit in a buffer the size of BufferSize in `CurrentMultipleKey`.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

```csharp
public bool RefreshMultipleKey(
    int BufferSize
)
```

Visual Basic (Declaration)

```vbnet
Public Function RefreshMultipleKey ( _
    BufferSize As Integer _
) As Boolean
```

Visual C++

```cpp
public:
    bool RefreshMultipleKey(
        int BufferSize
    )
```

Parameters

BufferSize
Type: `System::Int32`
The size of a buffer to fill with key/data pairs. Must be at least the page size of the underlying database and be a multiple of 1024.

Return Value

True if the cursor was positioned successfully, false otherwise.
See Also

Cursor Class
RefreshMultipleKey Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Cursor RefreshMultipleKey Method (Int32, LockingInfo)

See Also

Store the key/data pair to which the cursor refers and as many ensuing key/data pairs that can fit in a buffer the size of BufferSize in CurrentMultipleKey.

Namespace: BerkeleyDB
Syntax

C#

```csharp
public bool RefreshMultipleKey(
    int BufferSize,
    LockingInfo info
)
```

Visual Basic (Declaration)

```vbnet
Public Function RefreshMultipleKey ( _
    BufferSize As Integer, _
    info As LockingInfo _
) As Boolean
```

Visual C++

```cpp
public:
bool RefreshMultipleKey(
    int BufferSize,
    LockingInfo^ info
)
```

Parameters

BufferSize
Type: `System::::Int32`
The size of a buffer to fill with key/data pairs. Must be at least the page size of the underlying database and be a multiple of 1024.

info
Type: `BerkeleyDB::::LockingInfo`
The locking behavior to use.

Return Value

True if the cursor was positioned successfully, false otherwise.
See Also

Cursor Class
RefreshMultipleKey Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `Cursor` type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current</td>
<td>The key/data pair at which the cursor currently points.</td>
</tr>
<tr>
<td>CurrentMultiple</td>
<td>The key and multiple data items at which the cursor currently points.</td>
</tr>
<tr>
<td>CurrentMultipleKey</td>
<td>The multiple key and data items at which the cursor currently points.</td>
</tr>
<tr>
<td>Priority</td>
<td>The cache priority for pages referenced by the cursor.</td>
</tr>
</tbody>
</table>
See Also

Cursor Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The key/data pair at which the cursor currently points.

**Namespace:**  [BerkeleyDB](https://www.berkeleydb.com)

**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public KeyValuePair<DatabaseEntry, DatabaseEntry> Current { get; private; }

Visual Basic (Declaration)

Public Property Current As KeyValuePair(Of DatabaseEntry, DatabaseEntry)

Visual C++

public:
property KeyValuePair<DatabaseEntry^, DatabaseEntry^> Current {
    KeyValuePair<DatabaseEntry^, DatabaseEntry^> get ();
    void set (KeyValuePair<DatabaseEntry^, DatabaseEntry^> value);}
Remarks

Only one of Current, CurrentMultiple and CurrentMultipleKey will ever be non-empty.
See Also

Cursor Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The key and multiple data items at which the cursor currently points.

Namespace: BerkeleyDB
Syntax

**C#**

```csharp
public KeyValuePair<DatabaseEntry, MultipleDatabaseEntry> CurrentMultiple;
```

**Visual Basic (Declaration)**

```vbnet
Public Property CurrentMultiple As KeyValuePair(Of DatabaseEntry, MultipleDatabaseEntry)
```

**Visual C++**

```cpp
public:
    property KeyValuePair<DatabaseEntry^, MultipleDatabaseEntry^> CurrentMultiple;
    void set (KeyValuePair<DatabaseEntry^, MultipleDatabaseEntry^> value);
```
Remarks

Only one of Current, CurrentMultiple and CurrentMultipleKey will ever be non-empty.
See Also

Cursor Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The multiple key and data items at which the cursor currently points.

**Namespace:** [BerkeleyDB](https://docs.oracle.com/cd/E14382_01/ncv.910/ncv910_index.html)

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public MultipleKeyDatabaseEntry CurrentMultipleKey { get; private set; }

Visual Basic (Declaration)

Public Property CurrentMultipleKey As MultipleKeyDatabaseEntry

Visual C++

public:
property MultipleKeyDatabaseEntry^ CurrentMultipleKey {
MultipleKeyDatabaseEntry^ get ();
void set (MultipleKeyDatabaseEntry^ value);
}


Remarks

Only one of Current, CurrentMultiple and CurrentMultipleKey will ever be non-empty.
See Also

- Cursor Class
- BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The cache priority for pages referenced by the cursor.

Namespace: BerkeleyDB
Syntax

C#

```csharp
public CachePriority Priority { get; set; }
```

Visual Basic (Declaration)

```vbscript
Public Property Priority As CachePriority
```

Visual C++

```c++
public:
property CachePriority^ Priority {
    CachePriority^ get ();
    void set (CachePriority^ value);
}
```
Remarks

The priority of a page biases the replacement algorithm to be more or less likely to discard a page when space is needed in the buffer pool. The bias is temporary, and pages will eventually be discarded if they are not referenced again. The setting is only advisory, and does not guarantee pages will be treated in a specific way.
See Also

Cursor Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Specifies where to place duplicate data elements of the key to which the cursor refers.

Namespace: BerkeleyDB
Syntax

C#

public enum InsertLocation

Visual Basic (Declaration)

Public Enumeration InsertLocation

Visual C++

public enum class InsertLocation
<table>
<thead>
<tr>
<th>Member name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AFTER</td>
<td>The new element appears immediately after the current cursor position.</td>
</tr>
<tr>
<td>BEFORE</td>
<td>The new element appears immediately before the current cursor position.</td>
</tr>
<tr>
<td>FIRST</td>
<td>The new element appears as the first of the data items for the given key</td>
</tr>
<tr>
<td>LAST</td>
<td>The new element appears as the last of the data items for the given key</td>
</tr>
</tbody>
</table>
See Also

BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
A class representing configuration parameters for Cursor

Namespace: BerkeleyDB
Syntax

C#
public class CursorConfig

Visual Basic (Declaration)
Public Class CursorConfig

Visual C++
public ref class CursorConfig
Inheritance Hierarchy

System..:::Object
BerkeleyDB..:::CursorConfig
See Also

CursorConfig Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `CursorConfig` type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CursorConfig</td>
<td>Instantiate a new CursorConfig object</td>
</tr>
</tbody>
</table>
### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <code>Object</code> is equal to the current <code>Object</code>.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <code>Type</code> of the current instance.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a <code>String</code> that represents the current <code>Object</code>.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
</tbody>
</table>
## Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>IsolationDegree</strong></td>
<td>The isolation degree the cursor should use.</td>
</tr>
<tr>
<td><strong>Priority</strong></td>
<td>The cache priority for pages referenced by the cursor. Configure a transactional cursor to operate with read-only snapshot isolation. For databases with <strong>UseMVCC</strong> set, data values will be read as they are when the cursor is opened, without taking read locks.</td>
</tr>
<tr>
<td><strong>SnapshotIsolation</strong></td>
<td>This setting implicitly begins a transaction that is committed when the cursor is closed. This setting is silently ignored if <strong>UseMVCC</strong> is not set on the underlying database or if a transaction is supplied to <strong>Cursor</strong>()()</td>
</tr>
<tr>
<td><strong>WriteCursor</strong></td>
<td>If true, specify that the cursor will be used to update the database. The underlying database environment must have been opened with <strong>UseCDB</strong> set.</td>
</tr>
</tbody>
</table>
See Also

CursorConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Instantiate a new CursorConfig object

**Namespace:**  BerkeleyDB  
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

**C#**

```csharp
public CursorConfig()
```

**Visual Basic (Declaration)**

```vbnet
Public Sub New
```

**Visual C++**

```cpp
public:
CursorConfig()
```
See Also

CursorConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `CursorConfig` type exposes the following members.
### Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IsolationDegree</td>
<td>The isolation degree the cursor should use.</td>
</tr>
<tr>
<td>Priority</td>
<td>The cache priority for pages referenced by the cursor.</td>
</tr>
<tr>
<td></td>
<td>Configure a transactional cursor to operate with read-only snapshot isolation. For databases with <a href="#">UseMVCC</a> set, data values will be read as they are when the cursor is opened, without taking read locks.</td>
</tr>
<tr>
<td>SnapshotIsolation</td>
<td>This setting implicitly begins a transaction that is committed when the cursor is closed.</td>
</tr>
<tr>
<td></td>
<td>This setting is silently ignored if <a href="#">UseMVCC</a> is not set on the underlying database or if a transaction is supplied to <a href="#">CursorQQ</a>.</td>
</tr>
<tr>
<td>WriteCursor</td>
<td>If true, specify that the cursor will be used to update the database. The underlying database environment must have been opened with <a href="#">UseCDB</a> set.</td>
</tr>
</tbody>
</table>
See Also

CursorConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The isolation degree the cursor should use.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public Isolation IsolationDegree

Visual Basic (Declaration)

Public IsolationDegree As Isolation

Visual C++

public:
Isolation IsolationDegree
Remarks

**DEGREE TWO** ensures the stability of the current data item read by this cursor but permits data read by this cursor to be modified or deleted prior to the commit of the transaction for this cursor.

**DEGREE ONE** allows read operations performed by the cursor to return modified but not yet committed data. Silently ignored if the `ReadUncommitted` was not specified when the underlying database was opened.
See Also

CursorConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The cache priority for pages referenced by the cursor.

**Namespace:** BerkeleyDB
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public CachePriority Priority

Visual Basic (Declaration)

Public Priority As CachePriority

Visual C++

public:

CachePriority^ Priority
Remarks

The priority of a page biases the replacement algorithm to be more or less likely to discard a page when space is needed in the buffer pool. The bias is temporary, and pages will eventually be discarded if they are not referenced again. The setting is only advisory, and does not guarantee pages will be treated in a specific way.
See Also

CursorConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Configure a transactional cursor to operate with read-only snapshot isolation. For databases with UseMVCC set, data values will be read as they are when the cursor is opened, without taking read locks.

This setting implicitly begins a transaction that is committed when the cursor is closed.

This setting is silently ignored if UseMVCC is not set on the underlying database or if a transaction is supplied to Cursor().

Namespace: BerkeleyDB
Syntax

**C#**

```csharp
public bool SnapshotIsolation
```

**Visual Basic (Declaration)**

```vbnet
Public SnapshotIsolation As Boolean
```

**Visual C++**

```cpp
public:
    bool SnapshotIsolation
```
See Also

CursorConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, specify that the cursor will be used to update the database. The underlying database environment must have been opened with UseCDB set.

Namespace: BerkeleyDB
Syntax

C#

public bool WriteCursor

Visual Basic (Declaration)

Public WriteCursor As Boolean

Visual C++

public:
bool WriteCursor
See Also

CursorConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `CursorConfig` type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a <a href="#">String</a> that represents the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
</tbody>
</table>
See Also

CursorConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
A class representing a Berkeley DB database, a base class for access method specific classes.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public class Database : BaseDatabase, IDisposable

Visual Basic (Declaration)

Public Class Database
    Inherits BaseDatabase
    Implements IDisposable

Visual C++

public ref class Database : public BaseDatabase, IDisposable
Inheritance Hierarchy

System:::Object
BerkeleyDB:::BaseDatabase
BerkeleyDB:::Database
   BerkeleyDB:::BTreeDatabase
   BerkeleyDB:::HashDatabase
   BerkeleyDB:::QueueDatabase
   BerkeleyDB:::RecnoDatabase
See Also

Database Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The Database type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Close</td>
<td>Overloaded.</td>
</tr>
</tbody>
</table>

Although closing a database will close any open cursors, it is recommended that applications explicitly close all their Cursor objects before closing the database. The reason why is that when the cursor is explicitly closed, the memory allocated for it is reclaimed; however, this will not happen if you close a database while cursors are still opened.

The same rule, for the same reasons, hold true for Transaction objects. Simply make sure you resolve all your transaction objects before closing your database handle.

Because key/data pairs are cached in memory, applications should make a point to always either close database handles or sync their data to disk (using `Sync()` before exiting, to ensure that any data cached in main memory are reflected in the underlying file system.

When called on a database that is the primary database for a secondary index, the primary database should be closed only after all secondary indices referencing it have been closed.

When multiple threads are using the object concurrently, only a single thread may call the Close method.

The object may not be accessed again after Close is called, regardless of its outcome.
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Cursor</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>Delete</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>Dispose</strong></td>
<td>Release the resources held by this object, and close the database if it's still open. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified Object is equal to the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>Exists</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>Get</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>GetBoth</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>GetBothMultiple</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>GetMultiple</strong></td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>Join</strong></td>
<td>Create a specialized join cursor for use in performing equality or natural joins on secondary indices.</td>
</tr>
<tr>
<td><strong>Open</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>PrintFastStats</strong></td>
<td>The statistical information is described by the BTreeStats, HashStats, QueueStats, and RecnoStats classes.</td>
</tr>
<tr>
<td><strong>PrintStats</strong></td>
<td>The statistical information is described by the BTreeStats, HashStats, QueueStats, and RecnoStats classes.</td>
</tr>
<tr>
<td><strong>Put</strong></td>
<td>If the database supports duplicates, add the new data value at the end of the duplicate set. If the database supports sorted duplicates, the new data value is inserted at the correct sorted location.</td>
</tr>
<tr>
<td><strong>PutNoOverwrite</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td><strong>Salvarge</strong></td>
<td>Overloaded. Flushed any cached information to disk. (Inherited from <code>BaseDatabase</code>).</td>
</tr>
<tr>
<td><strong>Sync</strong></td>
<td>Returns a <code>String</code> that represents the current <code>Object</code>. (Inherited from <code>Object</code>).</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Overloaded. When called on a database configured with secondary indices, <code>Truncate</code> will truncate the primary database and all secondary indices. A count of the records discarded from the primary database is returned.</td>
</tr>
<tr>
<td><strong>Truncate</strong></td>
<td>Overloaded. Database upgrades are done in place and are destructive. For example, if pages need to be allocated and no disk space is available, the database may be left corrupted. Backups should be made before databases are upgraded. See Upgrading databases in the Programmer's Reference Guide for more information.</td>
</tr>
<tr>
<td><strong>Upgrade</strong></td>
<td>Overloaded. Verify does not perform any locking, even in Berkeley DB environments that are configured with a locking subsystem. As such, it should only be used on files that are not being modified by another thread of control.</td>
</tr>
<tr>
<td><strong>Verify</strong></td>
<td>Overloaded.</td>
</tr>
</tbody>
</table>
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AutoCommit</strong></td>
<td>If true, all database modification operations based on this object will be transactionally protected. (Inherited from <a href="#">BaseDatabase</a>)</td>
</tr>
<tr>
<td><strong>CacheSize</strong></td>
<td>The size of the shared memory buffer pool -- that is, the cache. (Inherited from <a href="#">BaseDatabase</a>)</td>
</tr>
<tr>
<td><strong>Creation</strong></td>
<td>The CreatePolicy with which this database was opened. (Inherited from <a href="#">BaseDatabase</a>)</td>
</tr>
<tr>
<td><strong>DatabaseName</strong></td>
<td>The name of this database, if it has one. (Inherited from <a href="#">BaseDatabase</a>)</td>
</tr>
<tr>
<td><strong>DoChecksum</strong></td>
<td>If true, do checksum verification of pages read into the cache from the backing filestore. (Inherited from <a href="#">BaseDatabase</a>)</td>
</tr>
<tr>
<td><strong>EncryptAlgorithm</strong></td>
<td>The algorithm used by the Berkeley DB library to perform encryption and decryption. (Inherited from <a href="#">BaseDatabase</a>)</td>
</tr>
<tr>
<td><strong>Encrypted</strong></td>
<td>If true, encrypt all data stored in the database. (Inherited from <a href="#">BaseDatabase</a>)</td>
</tr>
<tr>
<td><strong>Endianness</strong></td>
<td>The database byte order. (Inherited from <a href="#">BaseDatabase</a>)</td>
</tr>
<tr>
<td><strong>ErrorFeedback</strong></td>
<td>The mechanism for reporting detailed error messages to the application. (Inherited from <a href="#">BaseDatabase</a>)</td>
</tr>
<tr>
<td><strong>ErrorPrefix</strong></td>
<td>The prefix string that appears before error messages issued by Berkeley DB. (Inherited from <a href="#">BaseDatabase</a>)</td>
</tr>
<tr>
<td><strong>FileName</strong></td>
<td>The filename of this database, if it has one. (Inherited from <a href="#">BaseDatabase</a>)</td>
</tr>
<tr>
<td>Feature</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>FreeThreaded</strong></td>
<td>If true, the object is free-threaded; that is, concurrently usable by multiple threads in the address space. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>HasMultiple</strong></td>
<td>If true, the object references a physical file supporting multiple databases. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td></td>
<td>If true, the underlying database files were created on an architecture of the same byte order as the current one. This information may be used to determine whether application data needs to be adjusted for this architecture or not. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>InHostOrder</strong></td>
<td>If true, this database is not mapped into process memory.</td>
</tr>
<tr>
<td><strong>NoMMap</strong></td>
<td>See MMapSize for further information. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>NonDurableTxns</strong></td>
<td>If true, Berkeley DB will not write log records for this database. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>Pagesize</strong></td>
<td>The database's current page size. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>Priority</strong></td>
<td>The cache priority for pages referenced by this object. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>ReadOnly</strong></td>
<td>If true, this database has been opened for reading only. Any attempt to modify items in the database will fail, regardless of the actual permissions of any underlying files. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>ReadUncommitted</strong></td>
<td>If true, this database supports transactional read operations with degree 1 isolation. Read operations on the database may request the return of modified but not yet committed data. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>Transactional</strong></td>
<td>If true, this database has been opened in a transactional mode. (Inherited from <a href="https://example.com">BaseDatabase</a>.)</td>
</tr>
<tr>
<td><strong>Truncated</strong></td>
<td>If true, the underlying file was physically truncated upon open, discarding all previous databases it might have held. (Inherited from <a href="https://example.com">BaseDatabase</a>.)</td>
</tr>
<tr>
<td><strong>Type</strong></td>
<td>The type of the underlying access method (and file format). This value may be used to determine the type of the database after an <a href="https://example.com">Open(String, DatabaseConfig)</a>. (Inherited from <a href="https://example.com">BaseDatabase</a>.)</td>
</tr>
<tr>
<td><strong>UseMVCC</strong></td>
<td>If true, the database was opened with support for multiversion concurrency control. (Inherited from <a href="https://example.com">BaseDatabase</a>.)</td>
</tr>
</tbody>
</table>
See Also

Database Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `Database` type exposes the following members.
### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Close</td>
<td>Overloaded.</td>
</tr>
</tbody>
</table>

Although closing a database will close any open cursors, it is recommended that applications explicitly close all their Cursor objects before closing the database. The reason why is that when the cursor is explicitly closed, the memory allocated for it is reclaimed; however, this will not happen if you close a database while cursors are still opened.

The same rule, for the same reasons, hold true for `Transaction` objects. Simply make sure you resolve all your transaction objects before closing your database handle.

Because key/data pairs are cached in memory, applications should make a point to always either close database handles or sync their data to disk (using `Sync()`) before exiting, to ensure that any data cached in main memory are reflected in the underlying file system.

When called on a database that is the primary database for a secondary index, the primary database should be closed only after all secondary indices referencing it have been closed.

When multiple threads are using the object concurrently, only a single thread may call the Close method.

The object may not be accessed again after Close is called, regardless of its outcome.
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cursor</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>Delete</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>Dispose</td>
<td>Release the resources held by this object, and close the database if it's still open. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified Object is equal to the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Exists</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>Get</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>GetBoth</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>GetBothMultiple</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as a hash function for a particular type. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetMultiple</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>Join</td>
<td>Create a specialized join cursor for use in performing equality or natural joins on secondary indices.</td>
</tr>
<tr>
<td>Open</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>PrintFastStats</td>
<td>The statistical information is described by the BTreeStats, HashStats, QueueStats, and RecnoStats classes. Overloaded.</td>
</tr>
<tr>
<td>PrintStats</td>
<td>The statistical information is described by the BTreeStats, HashStats, QueueStats, and RecnoStats classes. Overloaded.</td>
</tr>
<tr>
<td>Put</td>
<td>If the database supports duplicates, add the new data value at the end of the duplicate set. If the database supports sorted duplicates, the new data value is inserted at the correct sorted location.</td>
</tr>
<tr>
<td>PutNoOverwrite</td>
<td>Overloaded.</td>
</tr>
</tbody>
</table>
- **Salvage**
  Overloaded.
  Flush any cached information to disk.
  (Inherited from BaseDatabase.)

- **Sync**
  Returns a String that represents the current Object.
  (Inherited from Object.)

- **ToString**
  Overloaded.
  When called on a database configured with secondary indices, Truncate will truncate the primary database and all secondary indices. A count of the records discarded from the primary database is returned.

- **Truncate**
  Overloaded.
  Database upgrades are done in place and are destructive. For example, if pages need to be allocated and no disk space is available, the database may be left corrupted. Backups should be made before databases are upgraded. See Upgrading databases in the Programmer's Reference Guide for more information.

- **Upgrade**
  Overloaded.
  Verify does not perform any locking, even in Berkeley DB environments that are configured with a locking subsystem. As such, it should only be used on files that are not being modified by another thread of control.
See Also

- Database Class
- BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Although closing a database will close any open cursors, it is recommended that applications explicitly close all their Cursor objects before closing the database. The reason why is that when the cursor is explicitly closed, the memory allocated for it is reclaimed; however, this will not happen if you close a database while cursors are still opened.

The same rule, for the same reasons, hold true for Transaction objects. Simply make sure you resolve all your transaction objects before closing your database handle.

Because key/data pairs are cached in memory, applications should make a point to always either close database handles or sync their data to disk (using Sync() before exiting, to ensure that any data cached in main memory are reflected in the underlying file system.

When called on a database that is the primary database for a secondary index, the primary database should be closed only after all secondary indices referencing it have been closed.

When multiple threads are using the object concurrently, only a single thread may call the Close method.

The object may not be accessed again after Close is called, regardless of its outcome.
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Close()()</td>
<td>Flush any cached database information to disk, close any open Cursor() objects, free any allocated resources, and close any underlying files. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>Close(Boolean)</td>
<td>Optionally flush any cached database information to disk, close any open Cursor() objects, free any allocated resources, and close any underlying files. (Inherited from BaseDatabase.)</td>
</tr>
</tbody>
</table>
See Also

Database Class
Database Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Database..::.Cursor Method

Database Class  See Also
# Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| **Cursor()** | Create a database cursor.  
(Inherited from [BaseDatabase](#).) |
| **Cursor(CursorConfig)** | Create a database cursor with the given configuration.  
(Inherited from [BaseDatabase](#).) |
| **Cursor(Transaction)** | Create a transactionally protected database cursor.  
(Inherited from [BaseDatabase](#).) |
| **Cursor(CursorConfig, Transaction)** | Create a transactionally protected database cursor with the given configuration.  
(Inherited from [BaseDatabase](#).) |
See Also

Database Class
Database Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Database::Delete Method

Database Class  See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delete(DatabaseEntry)</td>
<td>Remove key/data pairs from the database. The key/data pair associated with key is discarded from the database. In the presence of duplicate key values, all records associated with the designated key will be discarded. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>Delete(DatabaseEntry, Transaction)</td>
<td>Remove key/data pairs from the database. The key/data pair associated with key is discarded from the database. In the presence of duplicate key values, all records associated with the designated key will be discarded. (Inherited from BaseDatabase.)</td>
</tr>
</tbody>
</table>
See Also

Database Class
Database Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation

Database::Exists Method

Database Class  See Also
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exists(DatabaseEntry)</td>
<td>Check whether key appears in the database. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>Exists(DatabaseEntry, Transaction)</td>
<td>Check whether key appears in the database. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>Exists(DatabaseEntry, Transaction, LockingInfo)</td>
<td>Check whether key appears in the database. (Inherited from BaseDatabase.)</td>
</tr>
</tbody>
</table>
See Also

Database Class
Database Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Database::Get Method

See Also

Database Class
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Get(DatabaseEntry)</td>
<td>Retrieve a key/data pair from the database. In the presence of duplicate key values, Get will return the first data item for key. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>Get(DatabaseEntry, Transaction)</td>
<td>Retrieve a key/data pair from the database. In the presence of duplicate key values, Get will return the first data item for key. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>Get(DatabaseEntry, Transaction, LockingInfo)</td>
<td>Retrieve a key/data pair from the database. In the presence of duplicate key values, Get will return the first data item for key. (Inherited from BaseDatabase.)</td>
</tr>
</tbody>
</table>
See Also

Database Class
Database Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation
Database...::GetBoth Method

(Database Class) See Also
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetBoth(DatabaseEntry, DatabaseEntry)</td>
<td>Retrieve a key/data pair from the database which matches key and data. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>GetBoth(DatabaseEntry, DatabaseEntry, Transaction)</td>
<td>Retrieve a key/data pair from the database which matches key and data. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>GetBoth(DatabaseEntry, DatabaseEntry, Transaction, LockingInfo)</td>
<td>Retrieve a key/data pair from the database which matches key and data. (Inherited from BaseDatabase.)</td>
</tr>
</tbody>
</table>
See Also

Database Class
Database Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation

Database...:..GetBothMultiple Method

Database Class  See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetBothMultiple(DatabaseEntry, DatabaseEntry)</td>
<td>If a key/data pair in the database matches key and data, return the key and all duplicate data items.</td>
</tr>
<tr>
<td>GetBothMultiple(DatabaseEntry, DatabaseEntry, Int32)</td>
<td>If a key/data pair in the database matches key and data, return the key and all duplicate data items.</td>
</tr>
<tr>
<td>GetBothMultiple(DatabaseEntry, DatabaseEntry, Int32, Transaction)</td>
<td>If a key/data pair in the database matches key and data, return the key and all duplicate data items.</td>
</tr>
<tr>
<td>GetBothMultiple(DatabaseEntry, DatabaseEntry, Int32, Transaction, LockingInfo)</td>
<td>If a key/data pair in the database matches key and data, return the key and all duplicate data items.</td>
</tr>
</tbody>
</table>
See Also

Database Class
Database Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If a key/data pair in the database matches key and data, return the key and all duplicate data items.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public KeyValuePair<DatabaseEntry, MultipleDatabaseEntry> GetBothMultiple(DatabaseEntry key, DatabaseEntry data)

Visual Basic (Declaration)

Public Function GetBothMultiple (_
    key As DatabaseEntry, _
    data As DatabaseEntry _
) As KeyValuePair(Of DatabaseEntry, MultipleDatabaseEntry)

Visual C++

public:
    KeyValuePair<DatabaseEntry, MultipleDatabaseEntry> GetBothMultiple(DatabaseEntry key, DatabaseEntry data)

Parameters

key
    Type: BerkeleyDB::<DatabaseEntry
    The key to search for

data
    Type: BerkeleyDB::<DatabaseEntry
    The data to search for

Return Value

A KeyValuePair<(Of <(TKey, TValue)>)> whose Key parameter is key and whose Value parameter is the retrieved data items.
## Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>BerkeleyDB::::NotFoundException</td>
<td>A NotFoundException is thrown if key and data are not in the database.</td>
</tr>
<tr>
<td>BerkeleyDB::::KeyEmptyException</td>
<td>A KeyEmptyException is thrown if the database is a QueueDatabase or RecnoDatabase database and key exists, but was never explicitly created by the application or was later deleted.</td>
</tr>
</tbody>
</table>
See Also

Database Class
GetBothMultiple Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If a key/data pair in the database matches key and data, return the key and all duplicate data items.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public KeyValuePair<DatabaseEntry, MultipleDatabaseEntry> GetBothMultiple(DatabaseEntry key, DatabaseEntry data, int BufferSize)

Visual Basic (Declaration)

Public Function GetBothMultiple(key As DatabaseEntry, data As DatabaseEntry, BufferSize As Integer) As KeyValuePair(Of DatabaseEntry, MultipleDatabaseEntry)

Visual C++

public:
KeyValuePair<DatabaseEntry^, MultipleDatabaseEntry^> GetBothMultiple(DatabaseEntry^ key, DatabaseEntry^ data, int BufferSize)

Parameters

key
  Type: BerkeleyDB::DatabaseEntry
  The key to search for

data
  Type: BerkeleyDB::DatabaseEntry
  The data to search for

BufferSize
  Type: System::Int32
The initial size of the buffer to fill with duplicate data items. If the buffer is not large enough, it will be automatically resized.

**Return Value**

A `KeyValuePair(Of (TKey, TValue)>)` whose Key parameter is key and whose Value parameter is the retrieved data items.
## Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>BerkeleyDB::NotFoundException</td>
<td>A <code>NotFoundException</code> is thrown if key and data are not in the database.</td>
</tr>
<tr>
<td>BerkeleyDB::KeyEmptyException</td>
<td>A <code>KeyEmptyException</code> is thrown if the database is a <a href="#">QueueDatabase</a> or <a href="#">RecnoDatabase</a> database and key exists, but was never explicitly created by the application or was later deleted.</td>
</tr>
</tbody>
</table>
See Also

Database Class
GetBothMultiple Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If a key/data pair in the database matches key and data, return the key and all duplicate data items.

**Namespace:**  BerkeleyDB  
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
**Syntax**

### C#

```csharp
public KeyValuePair<DatabaseEntry, MultipleDatabaseEntry> GetBothMultiple(DatabaseEntry key,
    DatabaseEntry data,
    int BufferSize,
    Transaction txn
)
```

### Visual Basic (Declaration)

```vbnet
Public Function GetBothMultiple (_
    key As DatabaseEntry, _
    data As DatabaseEntry, _
    BufferSize As Integer, _
    txn As Transaction _
) As KeyValuePair(Of DatabaseEntry, MultipleDatabaseEntry)
```

### Visual C++

```cpp
public:
KeyValuePair<DatabaseEntry^, MultipleDatabaseEntry^> GetBothMultiple(
    DatabaseEntry^ key,
    DatabaseEntry^ data,
    int BufferSize,
    Transaction^ txn
)
```

### Parameters

**key**

Type: **BerkeleyDB::DatabaseEntry**
The key to search for

**data**

Type: **BerkeleyDB::DatabaseEntry**
The data to search for
BufferSize
  Type: `System..::..Int32`
  The initial size of the buffer to fill with duplicate data items. If the buffer is not large enough, it will be automatically resized.

txn
  Type: `BerkeleyDB..::..Transaction`
  txn is a Transaction object returned from `BeginTransaction()();` if the operation is part of a Berkeley DB Concurrent Data Store group, txn is a handle returned from `BeginCDSGroup()();` otherwise null.

**Return Value**

A `KeyValuePair<((TKey, TValue))>`) whose Key parameter is key and whose Value parameter is the retrieved data items.
## Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>BerkeleyDB::::NotFoundException</code></td>
<td>A <code>NotFoundException</code> is thrown if key and data are not in the database.</td>
</tr>
<tr>
<td></td>
<td>A <code>KeyEmptyException</code> is thrown if the database is a <code>QueueDatabase</code> or</td>
</tr>
<tr>
<td><code>BerkeleyDB::::KeyEmptyException</code></td>
<td>RecnoDatabase database and key exists, but was never explicitly created by the</td>
</tr>
<tr>
<td></td>
<td>application or was later deleted.</td>
</tr>
</tbody>
</table>
See Also

Database Class
GetBothMultiple Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If a key/data pair in the database matches key and data, return the key and all duplicate data items.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

```csharp
public KeyValuePair<DatabaseEntry, MultipleDatabaseEntry> GetBothMultiple(DatabaseEntry key, DatabaseEntry data, int BufferSize, Transaction txn, LockingInfo info)
```

Visual Basic (Declaration)

```vbnet
Public Function GetBothMultiple (_
    key As DatabaseEntry, _
    data As DatabaseEntry, _
    BufferSize As Integer, _
    txn As Transaction, _
    info As LockingInfo _
) As KeyValuePair(Of DatabaseEntry, MultipleDatabaseEntry)
```

Visual C++

```cpp
public: 
    KeyValuePair<DatabaseEntry^, MultipleDatabaseEntry^> GetBothMultiple(
        DatabaseEntry^ key,
        DatabaseEntry^ data,
        int BufferSize,
        Transaction^ txn,
        LockingInfo^ info
    )
```

Parameters

key
   Type: BerkeleyDB::DatabaseEntry
   The key to search for

data
Type: `BerkeleyDB::DatabaseEntry`
The data to search for

**BufferSize**
Type: `System::Int32`
The initial size of the buffer to fill with duplicate data items. If the buffer is not large enough, it will be automatically resized.

**txn**
Type: `BerkeleyDB::Transaction`
txn is a Transaction object returned from `BeginTransaction()`; if the operation is part of a Berkeley DB Concurrent Data Store group, txn is a handle returned from `BeginCDSGroup()`; otherwise null.

**info**
Type: `BerkeleyDB::LockingInfo`
The locking behavior to use.

**Return Value**

A `KeyValuePair<Of <(TKey, TValue)>)>` whose Key parameter is key and whose Value parameter is the retrieved data items.
## Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>BerkeleyDB::::NotFoundException</td>
<td>A <code>NotFoundException</code> is thrown if key and data are not in the database.</td>
</tr>
<tr>
<td></td>
<td>A <code>KeyEmptyException</code> is thrown if the database is a <a href="#">QueueDatabase</a> or <a href="#">RecnoDatabase</a> database and key exists, but was never explicitly created by the application or was later deleted.</td>
</tr>
</tbody>
</table>
See Also

Database Class
GetBothMultiple Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation
Database...: GetMultiple Method

Database Class  See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>GetMultiple(DatabaseEntry)</code></td>
<td>Retrieve a key and all duplicate data items from the database.</td>
</tr>
<tr>
<td><code>GetMultiple(DatabaseEntry, Int32)</code></td>
<td>Retrieve a key and all duplicate data items from the database.</td>
</tr>
<tr>
<td><code>GetMultiple(DatabaseEntry, Int32, Transaction)</code></td>
<td>Retrieve a key and all duplicate data items from the database.</td>
</tr>
<tr>
<td><code>GetMultiple(DatabaseEntry, Int32, Transaction, LockingInfo)</code></td>
<td>Retrieve a key and all duplicate data items from the database.</td>
</tr>
</tbody>
</table>
See Also

Database Class
Database Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Retrieve a key and all duplicate data items from the database.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

```csharp
public KeyValuePair<DatabaseEntry, MultipleDatabaseEntry> GetMultiple(DatabaseEntry key)
```

Visual Basic (Declaration)

```vbnet
Public Function GetMultiple( _
key As DatabaseEntry _
) As KeyValuePair(Of DatabaseEntry, MultipleDatabaseEntry)
```

Visual C++

```cpp
public:
KeyValuePair<DatabaseEntry^, MultipleDatabaseEntry^> GetMultiple( _
    DatabaseEntry^ key
```

Parameters

key

Type: BerkeleyDB::DatabaseEntry
The key to search for

Return Value

A KeyValuePair(Of<<(TKey, TValue>>) whose Key parameter is key and whose Value parameter is the retrieved data items.
## Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>BerkeleyDB:::NotFoundException</td>
<td>A <code>NotFoundException</code> is thrown if key is not in the database.</td>
</tr>
<tr>
<td>BerkeleyDB:::KeyEmptyException</td>
<td>A <code>KeyEmptyException</code> is thrown if the database is a <a href="#">QueueDatabase</a> or <a href="#">RecnoDatabase</a> database and key exists, but was never explicitly created by the application or was later deleted.</td>
</tr>
</tbody>
</table>
See Also

Database Class
GetMultiple Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Retrieve a key and all duplicate data items from the database.

**Namespace:** [BerkeleyDB](https://www.oracle.com/technetwork/database/database-technologies/berkeleydb/)

**Assembly:** `libdb_dotnet48` (in `libdb_dotnet48.dll`) Version: 4.8.24.0
Syntax

C#

```csharp
public KeyValuePair<DatabaseEntry, MultipleDatabaseEntry> GetMultiple(DatabaseEntry key, int BufferSize)
```

Visual Basic (Declaration)

```vbnet
Public Function GetMultiple (_
    key As DatabaseEntry, _
    BufferSize As Integer _
) As KeyValuePair(Of DatabaseEntry, MultipleDatabaseEntry)
```

Visual C++

```cpp
public:
    KeyValuePair<DatabaseEntry^, MultipleDatabaseEntry^> GetMultiple(
        DatabaseEntry^ key,
        int BufferSize
    )
```

Parameters

key
Type: BerkeleyDB::::DatabaseEntry
The key to search for

BufferSize
Type: System::::Int32
The initial size of the buffer to fill with duplicate data items. If the buffer is not large enough, it will be automatically resized.

Return Value

A KeyValuePair(Of (TKey, TValue)>) whose Key parameter is key and
whose Value parameter is the retrieved data items.
### Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>BerkeleyDB::::NotFoundException</code></td>
<td>A <code>NotFoundException</code> is thrown if key is not in the database. A <code>KeyEmptyException</code> is thrown if the database is a <code>QueueDatabase</code> or <code>RecnoDatabase</code> database and key exists, but was never explicitly created by the application or was later deleted.</td>
</tr>
</tbody>
</table>
See Also

Database Class
GetMultiple Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Retrieve a key and all duplicate data items from the database.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
**Syntax**

**C#**

```csharp
public KeyValuePair<DatabaseEntry, MultipleDatabaseEntry> GetMultiple(DatabaseEntry key,
int BufferSize,
Transaction txn
)
```

**Visual Basic (Declaration)**

```vbnet
Public Function GetMultiple( _
    key As DatabaseEntry, _
    BufferSize As Integer, _
    txn As Transaction _
) As KeyValuePair(Of DatabaseEntry, MultipleDatabaseEntry)
```

**Visual C++**

```cpp
public:
KeyValuePair<DatabaseEntry^, MultipleDatabaseEntry^> GetMultiple( _
    DatabaseEntry^ key,
    int BufferSize,
    Transaction^ txn
)
```

**Parameters**

**key**

Type: BerkeleyDB::DatabaseEntry

The key to search for

**BufferSize**

Type: System::Int32

The initial size of the buffer to fill with duplicate data items. If the buffer is not large enough, it will be automatically resized.

**txn**
Type: BerkeleyDB::Transaction
txn is a Transaction object returned from BeginTransaction(); if the operation is part of a Berkeley DB Concurrent Data Store group, txn is a handle returned from BeginCDSGroup(); otherwise null.

Return Value

A KeyValuePair<(Of <(TKey, TValue)>>) whose Key parameter is key and whose Value parameter is the retrieved data items.
See Also

Database Class
GetMultiple Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
GetMultiple Method (DatabaseEntry, Int32, Transaction, LockingInfo)

Database Class  See Also

Retrieve a key and all duplicate data items from the database.

Namespace:  BerkeleyDB
Syntax

C#

```csharp
public KeyValuePair<DatabaseEntry, MultipleDatabaseEntry> GetMultiple(
    DatabaseEntry key,
    int BufferSize,
    Transaction txn,
    LockingInfo info
)
```

Visual Basic (Declaration)

```vbnet
Public Function GetMultiple ( _
    key As DatabaseEntry, _
    BufferSize As Integer, _
    txn As Transaction, _
    info As LockingInfo _
) As KeyValuePair(Of DatabaseEntry, MultipleDatabaseEntry)
```

Visual C++

```cpp
public: 
    KeyValuePair<DatabaseEntry^, MultipleDatabaseEntry^> GetMultiple(
        DatabaseEntry^ key,
        int BufferSize,
        Transaction^ txn,
        LockingInfo^ info
    )
```

Parameters

key
  Type: BerkeleyDB::DatabaseEntry
  The key to search for

BufferSize
  Type: System::Int32
  The initial size of the buffer to fill with duplicate data items. If the buffer is not large enough, it will be automatically resized.
txn
Type: BerkeleyDB::Transaction
txn is a Transaction object returned from BeginTransaction(); if the operation is part of a Berkeley DB Concurrent Data Store group, txn is a handle returned from BeginCDSGroup(); otherwise null.

info
Type: BerkeleyDB::LockingInfo
The locking behavior to use.

Return Value

A KeyValuePair(Of (TKey, TValue)> whose Key parameter is key and whose Value parameter is the retrieved data items.
See Also

Database Class
GetMultiple Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Create a specialized join cursor for use in performing equality or natural joins on secondary indices.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
### Syntax

#### C#

public `JoinCursor` Join(
    `SecondaryCursor[]` lst,
    `bool` sortCursors
)

#### Visual Basic (Declaration)

Public Function Join (
    lst As `SecondaryCursor`(), _
    sortCursors As `Boolean` _
) As `JoinCursor`

#### Visual C++

public: `JoinCursor`^ Join(
    array<`SecondaryCursor`^>^` lst,
    `bool` sortCursors
)

### Parameters

**lst**

Type: array<`BerkeleyDB::::SecondaryCursor` >[][[]]

An array of `SecondaryCursors`. Each cursor must have been initialized to refer to the key on which the underlying database should be joined.

**sortCursors**

Type: `System::::Boolean`

If true, sort the cursors from the one that refers to the least number of data items to the one that refers to the most. If the data are structured so that cursors with many data items also share many common elements, higher performance will result from listing those cursors before cursors with fewer data items; that is, a sort order other than the default. A setting of false
permits applications to perform join optimization prior to calling Join.

**Return Value**

A specialized join cursor for use in performing equality or natural joins on secondary indices.
Remarks

Once the cursors have been passed as part of lst, they should not be accessed or modified until the newly created JoinCursor has been closed, or else inconsistent results may be returned.

Joined values are retrieved by doing a sequential iteration over the first cursor in lst, and a nested iteration over each secondary cursor in the order they are specified in the curslist parameter. This requires database traversals to search for the current datum in all the cursors after the first. For this reason, the best join performance normally results from sorting the cursors from the one that refers to the least number of data items to the one that refers to the most.
See Also

- Database Class
- BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation

Database::Open Method

Database Class  See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open(String, DatabaseConfig)</td>
<td>Instantiate a new Database object and open the database represented by Filename. The file specified by Filename must exist.</td>
</tr>
<tr>
<td>Open(String, DatabaseConfig, Transaction)</td>
<td>Instantiate a new Database object and open the database represented by Filename. The file specified by Filename must exist.</td>
</tr>
<tr>
<td>Open(String, String, DatabaseConfig)</td>
<td>Instantiate a new Database object and open the database represented by Filename and DatabaseName. The file specified by Filename must exist.</td>
</tr>
<tr>
<td>Open(String, String, DatabaseConfig, Transaction)</td>
<td>Instantiate a new Database object and open the database represented by Filename and DatabaseName. The file specified by Filename must exist.</td>
</tr>
</tbody>
</table>
See Also

Database Class
Database Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Instantiate a new Database object and open the database represented by Filename. The file specified by Filename must exist.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public static Database Open(
        string Filename,
        DatabaseConfig cfg
    )

Visual Basic (Declaration)

Public Shared Function Open ( _
        Filename As String, _
        cfg As DatabaseConfig _
    ) As Database

Visual C++

public:
    static Database^ Open(
        String^ Filename,
        DatabaseConfig^ cfg
    )

Parameters

Filename
    Type: System::::String
    The name of an underlying file that will be used to back the database.

cfg
    Type: BerkeleyDB::::DatabaseConfig
    The database's configuration

Return Value

A new, open database object
Remarks

If [AutoCommit](#) is set, the operation will be implicitly transaction protected. Note that transactionally protected operations on a database object requires the object itself be transactionally protected during its open.
See Also

Database Class
Open Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation

Database:::Open Method (String, DatabaseConfig, Transaction)

Database Class  See Also

Instantiate a new Database object and open the database represented by Filename. The file specified by Filename must exist.

Namespace: BerkeleyDB
### Syntax

**C#**

```csharp
public static Database Open(
    string Filename,
    DatabaseConfig cfg,
    Transaction txn
)
```

**Visual Basic (Declaration)**

```vbnet
Public Shared Function Open (_
    Filename As String, _
    cfg As DatabaseConfig, _
    txn As Transaction _
) As Database
```

**Visual C++**

```cpp
public:
static Database^ Open(
    String^ Filename, 
    DatabaseConfig^ cfg, 
    Transaction^ txn
)
```

### Parameters

Filename

Type: `System::String`

The name of an underlying file that will be used to back the database. In-memory databases never intended to be preserved on disk may be created by setting this parameter to null.

cfg

Type: `BerkeleyDB::DatabaseConfig`

The database’s configuration
txn

Type: BerkeleyDB::::Transaction
If the operation is part of an application-specified transaction, txn is a Transaction object returned from BeginTransaction(); if the operation is part of a Berkeley DB Concurrent Data Store group, txn is a handle returned from BeginCDSGroup(); otherwise null.

Return Value

A new, open database object
Remarks

If Filename is null, the database is strictly temporary and cannot be opened by any other thread of control, thus the database can only be accessed by sharing the single database object that created it, in circumstances where doing so is safe.

If txn is null, but AutoCommit is set, the operation will be implicitly transaction protected. Note that transactionally protected operations on a database object requires the object itself be transactionally protected during its open. Also note that the transaction must be committed before the object is closed.
See Also

Database Class
Open Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation

Database...Open Method (String, String, DatabaseConfig)

**Database Class**  **See Also**

Instantiate a new Database object and open the database represented by Filename and DatabaseName. The file specified by Filename must exist.

**Namespace:**  BerkeleyDB

**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
## Syntax

### C#

```csharp
public static Database Open(
    string Filename,
    string DatabaseName,
    DatabaseConfig cfg
)
```

### Visual Basic (Declaration)

```vbnet
Public Shared Function Open (
    Filename As String,
    DatabaseName As String,
    cfg As DatabaseConfig
) As Database
```

### Visual C++

```cpp
public:
static Database^ Open(
    String^ Filename,
    String^ DatabaseName,
    DatabaseConfig^ cfg
)
```

## Parameters

### Filename

**Type:** `System::::String`

The name of an underlying file that will be used to back the database. In-memory databases never intended to be preserved on disk may be created by setting this parameter to null.

### DatabaseName

**Type:** `System::::String`

This parameter allows applications to have multiple databases in a single file. Although no DatabaseName needs to be specified, it is an error to
attempt to open a second database in a file that was not initially created using a database name.

cfg
  Type: BerkeleyDB...::DatabaseConfig
  The database's configuration

**Return Value**

A new, open database object
Remarks

If Filename is null and DatabaseName is non-null, the database can be opened by other threads of control and will be replicated to client sites in any replication group.

If AutoCommit is set, the operation will be implicitly transaction protected. Note that transactionally protected operations on a database object requires the object itself be transactionally protected during its open.
See Also

Database Class
Open Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Database...:::Open Method (String, String, DatabaseConfig, Transaction)

Database Class  See Also

Instantiate a new Database object and open the database represented by Filename and DatabaseName. The file specified by Filename must exist.

Namespace: BerkeleyDB
Syntax

C#

```csharp
public static Database Open(
    string Filename,
    string DatabaseName,
    DatabaseConfig cfg,
    Transaction txn
)
```

Visual Basic (Declaration)

```vbnet
Public Shared Function Open()
    Filename As String,
    DatabaseName As String,
    cfg As DatabaseConfig,
    txn As Transaction
) As Database
```

Visual C++

```cpp
public:
static Database^ Open(
    String^ Filename,
    String^ DatabaseName,
    DatabaseConfig^ cfg,
    Transaction^ txn
)
```

Parameters

Filename

Type: System::String

The name of an underlying file that will be used to back the database. In-memory databases never intended to be preserved on disk may be created by setting this parameter to null.

DatabaseName

Type: System::String
This parameter allows applications to have multiple databases in a single file. Although no DatabaseName needs to be specified, it is an error to attempt to open a second database in a file that was not initially created using a database name.

**cfg**
Type: **BerkeleyDB::DatabaseConfig**
The database's configuration

**txn**
Type: **BerkeleyDB::Transaction**
If the operation is part of an application-specified transaction, txn is a Transaction object returned from `BeginTransaction()`; if the operation is part of a Berkeley DB Concurrent Data Store group, txn is a handle returned from `BeginCDSGroup()`; otherwise null.

**Return Value**

A new, open database object
Remarks

If both Filename and DatabaseName are null, the database is strictly temporary and cannot be opened by any other thread of control, thus the database can only be accessed by sharing the single database object that created it, in circumstances where doing so is safe. If Filename is null and DatabaseName is non-null, the database can be opened by other threads of control and will be replicated to client sites in any replication group.

If txn is null, but AutoCommit is set, the operation will be implicitly transaction protected. Note that transactionally protected operations on a database object requires the object itself be transactionally protected during its open. Also note that the transaction must be committed before the object is closed.
See Also

Database Class
Open Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The statistical information is described by the `BTreeStats`, `HashStats`, `QueueStats`, and `RecnoStats` classes.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PrintFastStats()()()</td>
<td>Display the database statistical information which does not require traversal of the database. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>PrintFastStats(Boolean)</td>
<td>Display the database statistical information which does not require traversal of the database. (Inherited from BaseDatabase.)</td>
</tr>
</tbody>
</table>
See Also

Database Class
Database Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The statistical information is described by the **BTreeStats**, **HashStats**, **QueueStats**, and **RecnoStats** classes.
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PrintStats()()</td>
<td>Display the database statistical information. (Inherited from BaseDatabase)</td>
</tr>
<tr>
<td>PrintStats(Boolean)</td>
<td>Display the database statistical information. (Inherited from BaseDatabase)</td>
</tr>
</tbody>
</table>
See Also

Database Class
Database Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If the database supports duplicates, add the new data value at the end of the duplicate set. If the database supports sorted duplicates, the new data value is inserted at the correct sorted location.
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Put(DatabaseEntry, DatabaseEntry)</td>
<td>Store the key/data pair in the database, replacing any previously existing key if duplicates are disallowed, or adding a duplicate data item if duplicates are allowed.</td>
</tr>
<tr>
<td>Put(DatabaseEntry, DatabaseEntry, Transaction)</td>
<td>Store the key/data pair in the database, replacing any previously existing key if duplicates are disallowed, or adding a duplicate data item if duplicates are allowed.</td>
</tr>
</tbody>
</table>
See Also

Database Class
Database Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Store the key/data pair in the database, replacing any previously existing key if duplicates are disallowed, or adding a duplicate data item if duplicates are allowed.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public void Put(
    DatabaseEntry key,
    DatabaseEntry data
)

Visual Basic (Declaration)

Public Sub Put (
    key As DatabaseEntry, 
    data As DatabaseEntry
)

Visual C++

public:
void Put(
    DatabaseEntry^ key,
    DatabaseEntry^ data
)

Parameters

key
Type: BerkeleyDB:::DatabaseEntry
The key to store in the database

data
Type: BerkeleyDB:::DatabaseEntry
The data item to store in the database
See Also

Database Class
Put Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Store the key/data pair in the database, replacing any previously existing key if duplicates are disallowed, or adding a duplicate data item if duplicates are allowed.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

```csharp
public void Put(
    DatabaseEntry key,
    DatabaseEntry data,
    Transaction txn
)
```

Visual Basic (Declaration)

```vbnet
Public Sub Put ( _
    key As DatabaseEntry, _
    data As DatabaseEntry, _
    txn As Transaction _
)
```

Visual C++

```cpp
public:
void Put(
    DatabaseEntry^ key,
    DatabaseEntry^ data,
    Transaction^ txn
)
```

Parameters

key
Type: `BerkeleyDB::DatabaseEntry`
The key to store in the database

data
Type: `BerkeleyDB::DatabaseEntry`
The data item to store in the database

txn
Type: `BerkeleyDB::Transaction`
If the operation is part of an application-specified transaction, \texttt{txn} is a Transaction object returned from \texttt{BeginTransaction()}; if the operation is part of a Berkeley DB Concurrent Data Store group, \texttt{txn} is a handle returned from \texttt{BeginCDSGroup()}; otherwise \texttt{null}. 
See Also

Database Class
Put Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation

Database PutNoOverwrite Method

Database Class See Also
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PutNoOverwrite(DatabaseEntry, DatabaseEntry)</td>
<td>Store the key/data pair in the database, only if the key does not already appear in the database.</td>
</tr>
<tr>
<td>PutNoOverwrite(DatabaseEntry, DatabaseEntry, Transaction)</td>
<td>Store the key/data pair in the database, only if the key does not already appear in the database.</td>
</tr>
</tbody>
</table>
See Also

Database Class
Database Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation

Database...: PutNoOverwrite Method (DatabaseEntry, DatabaseEntry)

Database Class  See Also

Store the key/data pair in the database, only if the key does not already appear in the database.

Namespace: BerkeleyDB
Syntax

C#

public void PutNoOverwrite(
       DatabaseEntry key,
       DatabaseEntry data
)

Visual Basic (Declaration)

Public Sub PutNoOverwrite ( _
       key As DatabaseEntry, _
       data As DatabaseEntry _
)

Visual C++

public:
void PutNoOverwrite(
       DatabaseEntry^ key,
       DatabaseEntry^ data
)

Parameters

key
   Type: BerkeleyDB::::DatabaseEntry
   The key to store in the database

data
   Type: BerkeleyDB::::DatabaseEntry
   The data item to store in the database
Remarks

This enforcement of uniqueness of keys applies only to the primary key, the behavior of insertions into secondary databases is not affected. In particular, the insertion of a record that would result in the creation of a duplicate key in a secondary database that allows duplicates would not be prevented by the use of this flag.
See Also

Database Class
PutNoOverwrite Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
PutNoOverwrite Method (DatabaseEntry, DatabaseEntry, Transaction)

Store the key/data pair in the database, only if the key does not already appear in the database.

Namespace: BerkeleyDB
Syntax

C#

```csharp
public void PutNoOverwrite(
    DatabaseEntry key,
    DatabaseEntry data,
    Transaction txn
)
```

Visual Basic (Declaration)

```vbnet
Public Sub PutNoOverwrite (_
    key As DatabaseEntry, _
    data As DatabaseEntry, _
    txn As Transaction _
)
```

Visual C++

```cpp
public:
void PutNoOverwrite(
    DatabaseEntry^ key,
    DatabaseEntry^ data,
    Transaction^ txn
)
```

Parameters

key
Type: BerkeleyDB::DatabaseEntry
The key to store in the database

data
Type: BerkeleyDB::DatabaseEntry
The data item to store in the database

txn
Type: BerkeleyDB::Transaction
If the operation is part of an application-specified transaction, txn is a Transaction object returned from `BeginTransaction()`; if the operation is part of a Berkeley DB Concurrent Data Store group, txn is a handle returned from `BeginCDSGroup()`; otherwise null.
Remarks

This enforcement of uniqueness of keys applies only to the primary key, the behavior of insertions into secondary databases is not affected. In particular, the insertion of a record that would result in the creation of a duplicate key in a secondary database that allows duplicates would not be prevented by the use of this flag.
See Also

Database Class
PutNoOverwrite Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation

Database:::

Salvage Method

Database Class  See Also
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Salvage(String, DatabaseConfig)</td>
<td>Write the key/data pairs from all databases in the file to <strong>Out</strong>. Key values are written for Btree, Hash and Queue databases, but not for Recno databases.</td>
</tr>
<tr>
<td>Salvage(String, DatabaseConfig, Boolean)</td>
<td>Write the key/data pairs from all databases in the file to <strong>Out</strong>. Key values are written for Btree, Hash and Queue databases, but not for Recno databases.</td>
</tr>
<tr>
<td>Salvage(String, DatabaseConfig, TextWriter)</td>
<td>Write the key/data pairs from all databases in the file to <strong>OutputStream</strong>. Key values are written for Btree, Hash and Queue databases, but not for Recno databases.</td>
</tr>
<tr>
<td>Salvage(String, DatabaseConfig, Boolean, Boolean)</td>
<td>Write the key/data pairs from all databases in the file to <strong>Out</strong>. Key values are written for Btree, Hash and Queue databases, but not for Recno databases.</td>
</tr>
<tr>
<td>Salvage(String, DatabaseConfig, Boolean, TextWriter)</td>
<td>Write the key/data pairs from all databases in the file to <strong>OutputStream</strong>. Key values are written for Btree, Hash and Queue databases, but not for Recno databases.</td>
</tr>
</tbody>
</table>
See Also

Database Class
Database Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Salvage Method (String, DatabaseConfig)

Database Class  See Also

Write the key/data pairs from all databases in the file to Out. Key values are written for Btree, Hash and Queue databases, but not for Recno databases.

Namespace: BerkeleyDB
Syntax

C#

```csharp
public static void Salvage(
    string file,
    DatabaseConfig cfg
)
```

Visual Basic (Declaration)

```vbnet
Public Shared Sub Salvage (_
    file As String, _
    cfg As DatabaseConfig _
)
```

Visual C++

```c++
public:
static void Salvage(
    String^ file,
    DatabaseConfig^ cfg
)
```

Parameters

file
Type: `System::String`
The physical file in which the databases to be salvaged are found.

cfg
Type: `BerkeleyDB::DatabaseConfig`
Configuration parameters for the databases to be salvaged.
See Also

Database Class
Salvage Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Write the key/data pairs from all databases in the file to **Out**. Key values are written for Btree, Hash and Queue databases, but not for Recno databases.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public static void Salvage(
    string file,
    DatabaseConfig cfg,
    bool Printable
)

Visual Basic (Declaration)

Public Shared Sub Salvage ( _
    file As String, _
    cfg As DatabaseConfig, _
    Printable As Boolean _
)

Visual C++

public:
    static void Salvage(
        String^ file,
        DatabaseConfig^ cfg,
        bool Printable
    )

Parameters

file
    Type: System::String
    The physical file in which the databases to be salvaged are found.

cfg
    Type: BerkeleyDB::DatabaseConfig
    Configuration parameters for the databases to be salvaged.

Printable
    Type: System::Boolean
If true and characters in either the key or data items are printing characters (as defined by isprint(3)), use printing characters to represent them. This setting permits users to use standard text editors and tools to modify the contents of databases or selectively remove data from salvager output.
See Also

Database Class
Salvage Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Salvage Method (String, DatabaseConfig, TextWriter)

Write the key/data pairs from all databases in the file to OutputStream. Key values are written for Btree, Hash and Queue databases, but not for Recno databases.

Namespace: BerkeleyDB
### Syntax

**C#**

```csharp
public static void Salvage(
    string file,
    DatabaseConfig cfg,
    TextWriter OutputStream
)
```

**Visual Basic (Declaration)**

```vbnet
Public Shared Sub Salvage (_
    file As String, _
    cfg As DatabaseConfig, _
    OutputStream As TextWriter _
)
```

**Visual C++**

```cpp
public:
static void Salvage(
    String^ file,
    DatabaseConfig^ cfg,
    TextWriter^ OutputStream
)
```

### Parameters

**file**

Type: `System::::String`

The physical file in which the databases to be salvaged are found.

**cfg**

Type: `BerkeleyDB::::DatabaseConfig`

Configuration parameters for the databases to be salvaged.

**OutputStream**

Type: `System.IO::::TextWriter`
The TextWriter to which the databases' key/data pairs are written. If null, **Out** will be used.
See Also

Database Class
Salvage Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Salvage Method (String, DatabaseConfig, Boolean, Boolean)

Write the key/data pairs from all databases in the file to Out. Key values are written for Btree, Hash and Queue databases, but not for Recno databases.

Namespace: BerkeleyDB
Syntax

C#

```csharp
public static void Salvage(
    string file,
    DatabaseConfig cfg,
    bool Printable,
    bool Aggressive
)
```

Visual Basic (Declaration)

```vbnet
Public Shared Sub Salvage (_
    file As String, _
    cfg As DatabaseConfig, _
    Printable As Boolean, _
    Aggressive As Boolean _
)
```

Visual C++

```cpp
public:
    static void Salvage(
        String^ file,
        DatabaseConfig^ cfg,
        bool Printable,
        bool Aggressive
    )
```

Parameters

file
Type: `System::String`
The physical file in which the databases to be salvaged are found.

cfg
Type: `BerkeleyDB::DatabaseConfig`
Configuration parameters for the databases to be salvaged.
Printable
  Type: System::Boolean
  If true and characters in either the key or data items are printing characters (as defined by isprint(3)), use printing characters to represent them. This setting permits users to use standard text editors and tools to modify the contents of databases or selectively remove data from salvager output.

Aggressive
  Type: System::Boolean
  If true, output all the key/data pairs in the file that can be found. Corruption will be assumed and key/data pairs that are corrupted or have been deleted may appear in the output (even if the file being salvaged is in no way corrupt), and the output will almost certainly require editing before being loaded into a database.
See Also

Database Class
Salvage Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Database...:::

Salvage Method (String, DatabaseConfig, Boolean, TextWriter)

Database Class  See Also

Write the key/data pairs from all databases in the file to OutputStream. Key values are written for Btree, Hash and Queue databases, but not for Recno databases.

Namespace:  BerkeleyDB
Syntax

C#

```csharp
public static void Salvage(
    string file,
    DatabaseConfig cfg,
    bool Printable,
    TextWriter OutputStream
)
```

Visual Basic (Declaration)

```vbnet
Public Shared Sub Salvage ( _
    file As String, _
    cfg As DatabaseConfig, _
    Printable As Boolean, _
    OutputStream As TextWriter _
)
```

Visual C++

```cpp
public:
    static void Salvage(
        String^ file,
        DatabaseConfig^ cfg,
        bool Printable,
        TextWriter^ OutputStream
    )
```

Parameters

file
Type: `System::::String`
The physical file in which the databases to be salvaged are found.

cfg
Type: `BerkeleyDB::::DatabaseConfig`
Configuration parameters for the databases to be salvaged.
Printable
Type: System..:::Boolean
If true and characters in either the key or data items are printing characters (as defined by isprint(3)), use printing characters to represent them. This setting permits users to use standard text editors and tools to modify the contents of databases or selectively remove data from salvager output.

OutputStream
Type: System.IO..:::TextWriter
The TextWriter to which the databases' key/data pairs are written. If null, Out will be used.
See Also

- Database Class
- Salvage Overload
- BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Salvage Method (String, DatabaseConfig, Boolean, Boolean, TextWriter)

Write the key/data pairs from all databases in the file to OutputStream. Key values are written for Btree, Hash and Queue databases, but not for Recno databases.

Namespace: BerkeleyDB
Syntax

C#

public static void Salvage(?string file,
    DatabaseConfig cfg,
    bool Printable,
    bool Aggressive,
    TextWriter OutputStream)

Visual Basic (Declaration)

Public Shared Sub Salvage ( _
    file As String, _
    cfg As DatabaseConfig, _
    Printable As Boolean, _
    Aggressive As Boolean, _
    OutputStream As TextWriter _
)

Visual C++

public:
static void Salvage(?String ^ file,
    DatabaseConfig ^ cfg,
    bool Printable,
    bool Aggressive,
    TextWriter ^ OutputStream)

Parameters

file
Type: System:::String
The physical file in which the databases to be salvaged are found.

cfg
Type: **BerkeleyDB::DatabaseConfig**

Configuration parameters for the databases to be salvaged.

**Printable**

Type: **System::Boolean**

If true and characters in either the key or data items are printing characters (as defined by isprint(3)), use printing characters to represent them. This setting permits users to use standard text editors and tools to modify the contents of databases or selectively remove data from salvager output.

**Aggressive**

Type: **System::Boolean**

If true, output all the key/data pairs in the file that can be found. Corruption will be assumed and key/data pairs that are corrupted or have been deleted may appear in the output (even if the file being salvaged is in no way corrupt), and the output will almost certainly require editing before being loaded into a database.

**OutputStream**

Type: **System.IO::TextWriter**

The TextWriter to which the databases' key/data pairs are written. If null, `Out` will be used.
See Also

Database Class
Salvage Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
When called on a database configured with secondary indices, Truncate will truncate the primary database and all secondary indices. A count of the records discarded from the primary database is returned.
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Truncate()()</td>
<td>Empty the database, discarding all records it contains. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>Truncate(Transaction)</td>
<td>Empty the database, discarding all records it contains. (Inherited from BaseDatabase.)</td>
</tr>
</tbody>
</table>
See Also

Database Class
Database Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Database upgrades are done in place and are destructive. For example, if pages need to be allocated and no disk space is available, the database may be left corrupted. Backups should be made before databases are upgraded. See Upgrading databases in the Programmer's Reference Guide for more information.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Upgrade(String, DatabaseConfig)</td>
<td>Upgrade all of the databases included in the file, if necessary. If no upgrade is necessary, Upgrade always returns successfully.</td>
</tr>
<tr>
<td>Upgrade(String, DatabaseConfig, Boolean)</td>
<td>Upgrade all of the databases included in the file, if necessary. If no upgrade is necessary, Upgrade always returns successfully.</td>
</tr>
</tbody>
</table>
See Also

Database Class
Database Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Upgrade all of the databases included in the file file, if necessary. If no upgrade is necessary, Upgrade always returns successfully.

**Namespace:**  BerkeleyDB
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public static void Upgrade(
    string file,
    DatabaseConfig cfg
)

Visual Basic (Declaration)

Public Shared Sub Upgrade ( _
    file As String, _
    cfg As DatabaseConfig _
)

Visual C++

public:
static void Upgrade(
    String^ file,
    DatabaseConfig^ cfg
)

Parameters

file
Type: System::String
The physical file containing the databases to be upgraded.

cfg
Type: BerkeleyDB::DatabaseConfig
Configuration parameters for the databases to be upgraded.
See Also

Database Class
Upgrade Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Upgrade all of the databases included in the file file, if necessary. If no upgrade is necessary, Upgrade always returns successfully.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public static void Upgrade(
    string file,
    DatabaseConfig cfg,
    bool dupSortUpgraded
)

Visual Basic (Declaration)

Public Shared Sub Upgrade ( _
    file As String, _
    cfg As DatabaseConfig, _
    dupSortUpgraded As Boolean _
)

Visual C++

public:
static void Upgrade(
    String^ file,
    DatabaseConfig^ cfg,
    bool dupSortUpgraded
)

Parameters

file
    Type: System::String
    The physical file containing the databases to be upgraded.

cfg
    Type: BerkeleyDB::DatabaseConfig
    Configuration parameters for the databases to be upgraded.

dupSortUpgraded
    Type: System::Boolean
If true, the duplicates in the upgraded database are sorted; otherwise they are assumed to be unsorted. This setting is only meaningful when upgrading databases from releases before the Berkeley DB 3.1 release.
Remarks

As part of the upgrade from the Berkeley DB 3.0 release to the 3.1 release, the on-disk format of duplicate data items changed. To correctly upgrade the format requires applications to specify whether duplicate data items in the database are sorted or not. Specifying dupSortUpgraded informs Upgrade that the duplicates are sorted; otherwise they are assumed to be unsorted. Incorrectly specifying the value of this flag may lead to database corruption.

Further, because this method upgrades a physical file (including all the databases it contains), it is not possible to use Upgrade to upgrade files in which some of the databases it includes have sorted duplicate data items, and some of the databases it includes have unsorted duplicate data items. If the file does not have more than a single database, if the databases do not support duplicate data items, or if all of the databases that support duplicate data items support the same style of duplicates (either sorted or unsorted), Upgrade will work correctly as long as dupSortUpgraded is correctly specified. Otherwise, the file cannot be upgraded using Upgrade it must be upgraded manually by dumping and reloading the databases.
See Also

Database Class
Upgrade Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Verify does not perform any locking, even in Berkeley DB environments that are configured with a locking subsystem. As such, it should only be used on files that are not being modified by another thread of control.
# Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Verify(String, DatabaseConfig)</td>
<td>Verify the integrity of all databases in the file specified by file.</td>
</tr>
<tr>
<td>Verify(String, DatabaseConfig, Database::VerifyOperation)</td>
<td>Verify the integrity of all databases in the file specified by file.</td>
</tr>
<tr>
<td>Verify(String, String, DatabaseConfig, Database::VerifyOperation)</td>
<td>Verify the integrity of the database specified by file and database.</td>
</tr>
</tbody>
</table>
See Also

Database Class
Database Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Verify the integrity of all databases in the file specified by file.

**Namespace:**  [BerkeleyDB](#)

**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
## Syntax

### C#

```csharp
public static void Verify(
    string file,
    DatabaseConfig cfg
)
```

### Visual Basic (Declaration)

```vbnet
Public Shared Sub Verify ( _
    file As String, _
    cfg As DatabaseConfig _
)
```

### Visual C++

```cpp
public:
static void Verify(
    String^ file,
    DatabaseConfig^ cfg
)
```

## Parameters

- **file**
  - Type: `System::String`
  - The physical file in which the databases to be verified are found.

- **cfg**
  - Type: `BerkeleyDB::DatabaseConfig`
  - Configuration parameters for the databases to be verified.
See Also

Database Class
Verify Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Verify Method (String, DatabaseConfig, Database.VerifyOperation)

Database Class  See Also

Verify the integrity of all databases in the file specified by file.

Namespace: BerkeleyDB
Syntax

C#

public static void Verify(
    string file,
    DatabaseConfig cfg,
    Database:::VerifyOperation op
)

Visual Basic (Declaration)

Public Shared Sub Verify (_
    file As String, _
    cfg As DatabaseConfig, _
    op As Database:::VerifyOperation _
)

Visual C++

public:
static void Verify(
    String^ file,
    DatabaseConfig^ cfg,
    Database:::VerifyOperation op
)

Parameters

file
    Type: System:::String
    The physical file in which the databases to be verified are found.

cfg
    Type: BerkeleyDB:::DatabaseConfig
    Configuration parameters for the databases to be verified.

op
    Type: BerkeleyDB:::Database:::VerifyOperation
The extent of verification
Remarks

Berkeley DB normally verifies that btree keys and duplicate items are correctly sorted, and hash keys are correctly hashed. If the file being verified contains multiple databases using differing sorting or hashing algorithms, some of them must necessarily fail database verification because only one sort order or hash function can be specified in cfg. To verify files with multiple databases having differing sorting orders or hashing functions, first perform verification of the file as a whole by using NO_ORDER_CHECK, and then individually verify the sort order and hashing function for each database in the file using ORDER_CHECK_ONLY.
See Also

Database Class
Verify Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Verify the integrity of the database specified by file and database.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
### Syntax

**C#**

```csharp
public static void Verify(
    string file,
    string database,
    DatabaseConfig cfg,
    Database:::VerifyOperation op
)
```

**Visual Basic (Declaration)**

```vbnet
Public Shared Sub Verify (_
    file As String, _
    database As String, _
    cfg As DatabaseConfig, _
    op As Database:::VerifyOperation _
)
```

**Visual C++**

```cpp
public:
static void Verify(
    String^ file,
    String^ database,
    DatabaseConfig^ cfg,
    Database:::VerifyOperation op
)
```

### Parameters

**file**

Type: `System:::String`

The physical file in which the databases to be verified are found.

**database**

Type: `System:::String`

The database in file on which the database checks for btree and duplicate sort order and for hashing are to be performed. A non-null value for
database is only allowed with ORDER_CHECKONLY.

cfg
Type: BerkeleyDB::DatabaseConfig
Configuration parameters for the databases to be verified.

op
Type: BerkeleyDB::Database::VerifyOperation
The extent of verification
Remarks

Berkeley DB normally verifies that btree keys and duplicate items are correctly sorted, and hash keys are correctly hashed. If the file being verified contains multiple databases using differing sorting or hashing algorithms, some of them must necessarily fail database verification because only one sort order or hash function can be specified in cfg. To verify files with multiple databases having differing sorting orders or hashing functions, first perform verification of the file as a whole by using NO_ORDER_CHECK, and then individually verify the sort order and hashing function for each database in the file using ORDER_CHECK_ONLY.
See Also

Database Class
Verify Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The **Database** type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AutoCommit</strong></td>
<td>If true, all database modification operations based on this object will be transactionally protected. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>CacheSize</strong></td>
<td>The size of the shared memory buffer pool -- that is, the cache. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>Creation</strong></td>
<td>The CreatePolicy with which this database was opened. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>DatabaseName</strong></td>
<td>The name of this database, if it has one. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>DoChecksum</strong></td>
<td>If true, do checksum verification of pages read into the cache from the backing filestore. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>EncryptAlgorithm</strong></td>
<td>The algorithm used by the Berkeley DB library to perform encryption and decryption. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>Encrypted</strong></td>
<td>If true, encrypt all data stored in the database. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>Endianness</strong></td>
<td>The database byte order. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>ErrorFeedback</strong></td>
<td>The mechanism for reporting detailed error messages to the application. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>ErrorPrefix</strong></td>
<td>The prefix string that appears before error messages issued by Berkeley DB. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>Feedback</strong></td>
<td>Monitor progress within long running operations. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>FileName</strong></td>
<td>The filename of this database, if it has one. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>FreeThreaded</td>
<td>If true, the object is free-threaded; that is, concurrently usable by multiple threads in the address space. (Inherited from <code>BaseDatabase</code>.)</td>
</tr>
<tr>
<td>HasMultiple</td>
<td>If true, the object references a physical file supporting multiple databases. (Inherited from <code>BaseDatabase</code>.)</td>
</tr>
<tr>
<td>InHostOrder</td>
<td>If true, the underlying database files were created on an architecture of the same byte order as the current one. This information may be used to determine whether application data needs to be adjusted for this architecture or not. (Inherited from <code>BaseDatabase</code>.)</td>
</tr>
<tr>
<td>NoMMap</td>
<td>If true, this database is not mapped into process memory. See <code>MMapSize</code> for further information. (Inherited from <code>BaseDatabase</code>.)</td>
</tr>
<tr>
<td>NonDurableTxns</td>
<td>If true, Berkeley DB will not write log records for this database. (Inherited from <code>BaseDatabase</code>.)</td>
</tr>
<tr>
<td>Pagesize</td>
<td>The database's current page size. (Inherited from <code>BaseDatabase</code>.)</td>
</tr>
<tr>
<td>Priority</td>
<td>The cache priority for pages referenced by this object. (Inherited from <code>BaseDatabase</code>.)</td>
</tr>
<tr>
<td>ReadOnly</td>
<td>If true, this database has been opened for reading only. Any attempt to modify items in the database will fail, regardless of the actual permissions of any underlying files. (Inherited from <code>BaseDatabase</code>.)</td>
</tr>
<tr>
<td>ReadUncommitted</td>
<td>If true, this database supports transactional read operations with degree 1 isolation. Read operations on the database may request the return of modified but not yet committed data. (Inherited from <code>BaseDatabase</code>.)</td>
</tr>
</tbody>
</table>
**Transactional**
If true, this database has been opened in a transactional mode.
(Inherited from [BaseDatabase](#)).

**Truncated**
If true, the underlying file was physically truncated upon open, discarding all previous databases it might have held.
(Inherited from [BaseDatabase](#)).

**Type**
The type of the underlying access method (and file format). This value may be used to determine the type of the database after an Open(String, DatabaseConfig).
(Inherited from [BaseDatabase](#)).

**UseMVCC**
If true, the database was opened with support for multiversion concurrency control.
(Inherited from [BaseDatabase](#)).
See Also

Database Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Database Class See Also

Specifies the type of verification to perform

Namespace: BerkeleyDB
Syntax

C#

public enum VerifyOperation

Visual Basic (Declaration)

Public Enumeration VerifyOperation

Visual C++

public enum class VerifyOperation
<table>
<thead>
<tr>
<th>Member name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEFAULT</td>
<td>Perform database checks and check sort order</td>
</tr>
<tr>
<td>ORDER_CHECK_ONLY</td>
<td>Perform the database checks for btree and duplicate sort order and for hashing</td>
</tr>
<tr>
<td>NO_ORDER_CHECK</td>
<td>Skip the database checks for btree and duplicate sort order and for hashing.</td>
</tr>
</tbody>
</table>
See Also

BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
A class representing configuration parameters for Database

Namespace: BerkeleyDB
**Syntax**

**C#**

```csharp
public class DatabaseConfig
```

**Visual Basic (Declaration)**

```vbnet
Public Class DatabaseConfig
```

**Visual C++**

```cpp
public ref class DatabaseConfig
```
Inheritance Hierarchy

System::Object
BerkeleyDB::DatabaseConfig
  BerkeleyDB::BTreeDatabaseConfig
  BerkeleyDB::HashDatabaseConfig
  BerkeleyDB::QueueDatabaseConfig
  BerkeleyDB::RecnoDatabaseConfig
  BerkeleyDB::SecondaryDatabaseConfig
See Also

DatabaseConfig Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `DatabaseConfig` type exposes the following members.
### Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DatabaseConfig</td>
<td>Instantiate a new DatabaseConfig object</td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>SetEncryption</strong></td>
<td>Set the password and algorithm used by the Berkeley DB library to perform encryption and decryption.</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a <a href="#">String</a> that represents the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
</tbody>
</table>
# Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AutoCommit</strong></td>
<td>Enclose the open call within a transaction. If the call succeeds, the open operation will be recoverable and all subsequent database modification operations based on this handle will be transactionally protected. If the call fails, no database will have been created.</td>
</tr>
<tr>
<td><strong>ByteOrder</strong></td>
<td>The byte order for integers in the stored database metadata. The host byte order of the machine where the Berkeley DB library was compiled is the default value.</td>
</tr>
<tr>
<td><strong>CacheSize</strong></td>
<td>The size of the shared memory buffer pool -- that is, the cache.</td>
</tr>
<tr>
<td><strong>DoChecksum</strong></td>
<td>If true, do checksum verification of pages read into the cache from the backing filestore.</td>
</tr>
<tr>
<td><strong>Env</strong></td>
<td>The Berkeley DB environment within which to create a database. If null, the database will be created stand-alone; that is, it is not part of any Berkeley DB environment.</td>
</tr>
<tr>
<td><strong>ErrorFeedback</strong></td>
<td>The mechanism for reporting error messages to the application.</td>
</tr>
<tr>
<td><strong>ErrorPrefix</strong></td>
<td>The prefix string that appears before error messages issued by Berkeley DB.</td>
</tr>
<tr>
<td><strong>Feedback</strong></td>
<td></td>
</tr>
<tr>
<td><strong>FreeThreaded</strong></td>
<td>Cause the database object to be free-threaded; that is, concurrently usable by multiple threads in the address space.</td>
</tr>
<tr>
<td><strong>NoMMap</strong></td>
<td>Do not map this database into process memory.</td>
</tr>
<tr>
<td><strong>NonDurableTxns</strong></td>
<td>If true, Berkeley DB will not write log records for this database.</td>
</tr>
<tr>
<td><strong>Priority</strong></td>
<td>The cache priority for pages referenced by the database.</td>
</tr>
</tbody>
</table>
- **ReadOnly**
  Open the database for reading only. Any attempt to modify items in the database will fail, regardless of the actual permissions of any underlying files.

- **ReadUncommitted**
  Support transactional read operations with degree 1 isolation.

- **Truncate**
  Physically truncate the underlying file, discarding all previous databases it might have held.

- **UseMVCC**
  Open the database with support for multiversion concurrency control.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EncryptAlgorithm</td>
<td>The algorithm used to perform encryption and decryption.</td>
</tr>
<tr>
<td>EncryptionPassword</td>
<td>The password used to perform encryption and decryption.</td>
</tr>
<tr>
<td>PageSize</td>
<td>The size of the pages used to hold items in the database, in bytes.</td>
</tr>
</tbody>
</table>
See Also

DatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Instantiate a new DatabaseConfig object

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public DatabaseConfig()

Visual Basic (Declaration)

Public Sub New

Visual C++

public:
DatabaseConfig()
See Also

DatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `DatabaseConfig` type exposes the following members.
## Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AutoCommit</strong></td>
<td>Enclose the open call within a transaction. If the call succeeds, the open operation will be recoverable and all subsequent database modification operations based on this handle will be transactionally protected. If the call fails, no database will have been created.</td>
</tr>
<tr>
<td><strong>ByteOrder</strong></td>
<td>The byte order for integers in the stored database metadata. The host byte order of the machine where the Berkeley DB library was compiled is the default value.</td>
</tr>
<tr>
<td><strong>CacheSize</strong></td>
<td>The size of the shared memory buffer pool -- that is, the cache.</td>
</tr>
<tr>
<td><strong>DoChecksum</strong></td>
<td>If true, do checksum verification of pages read into the cache from the backing filestore.</td>
</tr>
<tr>
<td><strong>Env</strong></td>
<td>The Berkeley DB environment within which to create a database. If null, the database will be created stand-alone; that is, it is not part of any Berkeley DB environment.</td>
</tr>
<tr>
<td><strong>ErrorFeedback</strong></td>
<td>The mechanism for reporting error messages to the application.</td>
</tr>
<tr>
<td><strong>ErrorPrefix</strong></td>
<td>The prefix string that appears before error messages issued by Berkeley DB.</td>
</tr>
<tr>
<td><strong>Feedback</strong></td>
<td></td>
</tr>
<tr>
<td><strong>FreeThreaded</strong></td>
<td>Cause the database object to be free-threaded; that is, concurrently usable by multiple threads in the address space.</td>
</tr>
<tr>
<td><strong>NoMMap</strong></td>
<td>Do not map this database into process memory.</td>
</tr>
<tr>
<td><strong>NonDurableTxns</strong></td>
<td>If true, Berkeley DB will not write log records for this database.</td>
</tr>
<tr>
<td><strong>Priority</strong></td>
<td>The cache priority for pages referenced by the database.</td>
</tr>
</tbody>
</table>
- **ReadOnly**
  Open the database for reading only. Any attempt to modify items in the database will fail, regardless of the actual permissions of any underlying files.

- **ReadUncommitted**
  Support transactional read operations with degree 1 isolation.

- **Truncate**
  Physically truncate the underlying file, discarding all previous databases it might have held.

- **UseMVCC**
  Open the database with support for multiversion concurrency control.
See Also

DatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Enclose the open call within a transaction. If the call succeeds, the open operation will be recoverable and all subsequent database modification operations based on this handle will be transactionally protected. If the call fails, no database will have been created.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool AutoCommit

Visual Basic (Declaration)

Public AutoCommit As Boolean

Visual C++

public:
bool AutoCommit
See Also

DatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation
DatabaseConfig::ByteOrder Field

**DatabaseConfig Class**  **See Also**

The byte order for integers in the stored database metadata. The host byte order of the machine where the Berkeley DB library was compiled is the default value.

**Namespace:**  [BerkeleyDB](#)

**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public ByteOrder ByteOrder

Visual Basic (Declaration)

Public ByteOrder As ByteOrder

Visual C++

public:
ByteOrder^ ByteOrder
Remarks

The access methods provide no guarantees about the byte ordering of the application data stored in the database, and applications are responsible for maintaining any necessary ordering.

If creating additional databases in a single physical file, this parameter will be ignored and the byte order of the existing databases will be used.
See Also

DatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The size of the shared memory buffer pool -- that is, the cache.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public CacheInfo CacheSize

Visual Basic (Declaration)

Public CacheSize As CacheInfo

Visual C++

public:
CacheInfo^ CacheSize
Remarks

The cache should be the size of the normal working data set of the application, with some small amount of additional memory for unusual situations. (Note: the working set is not the same as the number of pages accessed simultaneously, and is usually much larger.)

The default cache size is 256KB, and may not be specified as less than 20KB. Any cache size less than 500MB is automatically increased by 25% to account for buffer pool overhead; cache sizes larger than 500MB are used as specified. The maximum size of a single cache is 4GB on 32-bit systems and 10TB on 64-bit systems. (All sizes are in powers-of-two, that is, 256KB is $2^{18}$ not 256,000.) For information on tuning the Berkeley DB cache size, see Selecting a cache size in the Programmer's Reference Guide.
See Also

DatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, do checksum verification of pages read into the cache from the backing file store.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool DoChecksum

Visual Basic (Declaration)

Public DoChecksum As Boolean

Visual C++

public:
    bool DoChecksum
Remarks

Berkeley DB uses the SHA1 Secure Hash Algorithm if encryption is configured and a general hash algorithm if it is not.

If the database already exists, this setting will be ignored.
See Also

DatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The Berkeley DB environment within which to create a database. If null, the database will be created stand-alone; that is, it is not part of any Berkeley DB environment.

**Namespace:**  BerkeleyDB  
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll)  Version: 4.8.24.0
Syntax

C#

public DatabaseEnvironment Env

Visual Basic (Declaration)

Public Env As DatabaseEnvironment

Visual C++

public: DatabaseEnvironment^ Env
Remarks

The database access methods automatically make calls to the other subsystems in Berkeley DB, based on the enclosing environment. For example, if the environment has been configured to use locking, the access methods will automatically acquire the correct locks when reading and writing pages of the database.
See Also

DatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The mechanism for reporting error messages to the application.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public ErrorFeedbackDelegate ErrorFeedback

Visual Basic (Declaration)

Public ErrorFeedback As ErrorFeedbackDelegate

Visual C++

public:
ErrorFeedbackDelegate^ ErrorFeedback
Remarks

In some cases, when an error occurs, Berkeley DB will call ErrorFeedback with additional error information. It is up to the delegate function to display the error message in an appropriate manner.

This error-logging enhancement does not slow performance or significantly increase application size, and may be run during normal operation as well as during application debugging.

For databases opened inside of Berkeley DB environments, setting ErrorFeedback affects the entire environment and is equivalent to setting ErrorFeedback.
See Also

DatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation

DatabaseConfig.ErrorPrefix Field

DatabaseConfig Class  See Also

The prefix string that appears before error messages issued by Berkeley DB.

Namespace: BerkeleyDB
Syntax

C#

public string ErrorPrefix

Visual Basic (Declaration)

Public ErrorPrefix As String

Visual C++

public:
String^ ErrorPrefix
See Also

DatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation
DatabaseConfig Feedback Field

Namespace: BerkeleyDB
Syntax

C#

public DatabaseFeedbackDelegate Feedback

Visual Basic (Declaration)

Public Feedback As DatabaseFeedbackDelegate

Visual C++

public: DatabaseFeedbackDelegate^ Feedback
See Also

DatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Cause the database object to be free-threaded; that is, concurrently usable by multiple threads in the address space.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

**C#**

```csharp
public bool FreeThreaded
```

**Visual Basic (Declaration)**

```vbnet
Public FreeThreaded As Boolean
```

**Visual C++**

```cpp
public:
  bool FreeThreaded
```
See Also

DatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Do not map this database into process memory.

Namespace: BerkeleyDB
Syntax

**C#**

public `bool` NoMMap

**Visual Basic (Declaration)**

Public NoMMap As `Boolean`

**Visual C++**

public:

`bool` NoMMap
See Also

DatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, Berkeley DB will not write log records for this database.

**Namespace:** [BerkeleyDB](https://www.bereleydb.com)

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool NonDurableTxns

Visual Basic (Declaration)

Public NonDurableTxns As Boolean

Visual C++

public:
bool NonDurableTxns
Remarks

If Berkeley DB does not write log records, updates of this database will exhibit the ACI (atomicity, consistency, and isolation) properties, but not D (durability); that is, database integrity will be maintained, but if the application or system fails, integrity will not persist. The database file must be verified and/or restored from backup after a failure. In order to ensure integrity after application shut down, the database must be synced when closed, or all database changes must be flushed from the database environment cache using either Checkpoint() or SyncMemPool(). All database objects for a single physical file must set NonDurableTxns, including database objects for different databases in a physical file.
See Also

DatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The cache priority for pages referenced by the database.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public CachePriority Priority

Visual Basic (Declaration)

Public Priority As CachePriority

Visual C++

public:
    CachePriority^ Priority
Remarks

The priority of a page biases the replacement algorithm to be more or less likely to discard a page when space is needed in the buffer pool. The bias is temporary, and pages will eventually be discarded if they are not referenced again. This priority is only advisory, and does not guarantee pages will be treated in a specific way.
See Also

[DatabaseConfig Class](#)
[BerkeleyDB Namespace](#)

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Open the database for reading only. Any attempt to modify items in the database will fail, regardless of the actual permissions of any underlying files.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool ReadOnly

Visual Basic (Declaration)

Public ReadOnly As Boolean

Visual C++

public:

bool ReadOnly
See Also

DatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Support transactional read operations with degree 1 isolation.

**Namespace:**  [BerkeleyDB](https://www.bdb.com)

**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool ReadUncommitted

Visual Basic (Declaration)

Public ReadUncommitted As Boolean

Visual C++

public:
bool ReadUncommitted
Remarks

Read operations on the database may request the return of modified but not yet committed data. This flag must be specified on all database objects used to perform dirty reads or database updates, otherwise requests for dirty reads may not be honored and the read may block.
See Also

DatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Physically truncate the underlying file, discarding all previous databases it might have held.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll)  
**Version:** 4.8.24.0
Syntax

C#
public bool Truncate

Visual Basic (Declaration)
Public Truncate As Boolean

Visual C++
public:
bool Truncate
Remarks

Underlying filesystem primitives are used to implement this flag. For this reason, it is applicable only to the file and cannot be used to discard databases within a file.

This setting cannot be lock or transaction-protected, and it is an error to specify it in a locking or transaction-protected environment.
See Also

DatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Open the database with support for multiversion concurrency control.

Namespace: BerkeleyDB
Syntax

C#

public bool UseMVCC

Visual Basic (Declaration)

Public UseMVCC As Boolean

Visual C++

public:
    bool UseMVCC
Remarks

This will cause updates to the database to follow a copy-on-write protocol, which is required to support snapshot isolation. This setting requires that the database be transactionally protected during its open and is not supported by the queue format.
See Also

- DatabaseConfig Class
- BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `DatabaseConfig` type exposes the following members.
# Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="https://docs.oracle.com/javase/8/docs/api/java/lang/Object.html">Object</a> is equal to the current <a href="https://docs.oracle.com/javase/8/docs/api/java/lang/Object.html">Object</a>. (Inherited from <a href="https://docs.oracle.com/javase/8/docs/api/java/lang/Object.html">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="https://docs.oracle.com/javase/8/docs/api/java/lang/Object.html">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <a href="https://docs.oracle.com/javase/8/docs/api/java/lang/Type.html">Type</a> of the current instance. (Inherited from <a href="https://docs.oracle.com/javase/8/docs/api/java/lang/Object.html">Object</a>.)</td>
</tr>
<tr>
<td><strong>SetEncryption</strong></td>
<td>Set the password and algorithm used by the Berkeley DB library to perform encryption and decryption.</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a <a href="https://docs.oracle.com/javase/8/docs/api/java/lang/String.html">String</a> that represents the current <a href="https://docs.oracle.com/javase/8/docs/api/java/lang/Object.html">Object</a>. (Inherited from <a href="https://docs.oracle.com/javase/8/docs/api/java/lang/Object.html">Object</a>.)</td>
</tr>
</tbody>
</table>
See Also

DatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Set the password and algorithm used by the Berkeley DB library to perform encryption and decryption.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public void SetEncryption(
    string password,
    EncryptionAlgorithm alg
)

Visual Basic (Declaration)

Public Sub SetEncryption (_
    password As String, _
    alg As EncryptionAlgorithm _
)

Visual C++

public:
void SetEncryption(
    String^ password,
    EncryptionAlgorithm alg
)

Parameters

password
    Type: System::String
    The password used to perform encryption and decryption.

alg
    Type: BerkeleyDB::EncryptionAlgorithm
    The algorithm used to perform encryption and decryption.
See Also

DatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `DatabaseConfig` type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EncryptAlgorithm</td>
<td>The algorithm used to perform encryption and decryption.</td>
</tr>
<tr>
<td>EncryptionPassword</td>
<td>The password used to perform encryption and decryption.</td>
</tr>
<tr>
<td>PageSize</td>
<td>The size of the pages used to hold items in the database, in bytes.</td>
</tr>
</tbody>
</table>
See Also

DatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The algorithm used to perform encryption and decryption.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
**Syntax**

**C#**

```csharp
public EncryptionAlgorithm EncryptAlgorithm { get; }
```

**Visual Basic (Declaration)**

Public ReadOnly Property EncryptAlgorithm As EncryptionAlgorithm

**Visual C++**

```c++
public:
    property EncryptionAlgorithm EncryptAlgorithm {
        EncryptionAlgorithm get ();
    }
```
See Also

DatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The password used to perform encryption and decryption.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public string EncryptionPassword { get; }

Visual Basic (Declaration)

Public ReadOnly Property EncryptionPassword As String

Visual C++

public:
property String^ EncryptionPassword {
    String^ get ();
}

See Also

DatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The size of the pages used to hold items in the database, in bytes.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public uint PageSize { get; set; }

Visual Basic (Declaration)

Public Property PageSize AsUInteger

Visual C++

public:
property unsigned int PageSize {
    unsigned int get ();
    void set (unsigned int value);
}
Remarks

The minimum page size is 512 bytes, the maximum page size is 64K bytes, and the page size must be a power-of-two. If the page size is not explicitly set, one is selected based on the underlying filesystem I/O block size. The automatically selected size has a lower limit of 512 bytes and an upper limit of 16K bytes.

For information on tuning the Berkeley DB page size, see Selecting a page size in the Programmer's Reference Guide.

If creating additional databases in a single physical file, this parameter will be ignored and the page size of the existing databases will be used.
See Also

DatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
A class representing a key or data item in a Berkeley DB database

Namespace: BerkeleyDB
Syntax

C#

public class DatabaseEntry : IDisposable

Visual Basic (Declaration)

Public Class DatabaseEntry
    Implements IDisposable

Visual C++

public ref class DatabaseEntry : IDisposable
Inheritance Hierarchy

System:::Object
BerkeleyDB:::DatabaseEntry
See Also

DatabaseEntry Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `DatabaseEntry` type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DatabaseEntry</td>
<td>Overloaded.</td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dispose</td>
<td>Release the resources held by the underlying C library.</td>
</tr>
<tr>
<td></td>
<td>Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a>.</td>
</tr>
<tr>
<td>Equals</td>
<td>Serves as a hash function for a particular type.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Gets the <a href="#">Type</a> of the current instance.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Returns a <a href="#">String</a> that represents the current <a href="#">Object</a>.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Object</a>.)</td>
</tr>
</tbody>
</table>
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>The byte string stored in or retrieved from a database</td>
</tr>
</tbody>
</table>
See Also

DatabaseEntry Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation
DatabaseEntry Constructor

See Also
## Overload List

<table>
<thead>
<tr>
<th><strong>Name</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>DatabaseEntry()()</td>
<td>Create a new, empty DatabaseEntry object.</td>
</tr>
<tr>
<td>DatabaseEntry(array&lt;Byte&gt;[[]])</td>
<td>Create a new DatabaseEntry object, with the specified data</td>
</tr>
</tbody>
</table>
See Also

DatabaseEntry Class
DatabaseEntry Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Create a new, empty DatabaseEntry object.

**Namespace:**  BerkeleyDB  
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll)  Version: 4.8.24.0
Syntax

C#

public DatabaseEntry(

Visual Basic (Declaration)

Public Sub New

Visual C++

public:
DatabaseEntry()
See Also

DatabaseEntry Class
DatabaseEntry Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Create a new DatabaseEntry object, with the specified data

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
### Syntax

**C#**

```csharp
public DatabaseEntry(
    byte[] data
)
```

**Visual Basic (Declaration)**

```vbnet
Public Sub New (_
    data As Byte() _
)
```

**Visual C++**

```cpp
public:
DatabaseEntry(
    array<unsigned char>^ data
)
```

### Parameters

```csharp
data
```

Type: array<[System::.::Byte >[]][[]

The new object's **Data**
See Also

DatabaseEntry Class
DatabaseEntry Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `DatabaseEntry` type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Dispose</strong></td>
<td>Release the resources held by the underlying C library.</td>
</tr>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <em>Object</em> is equal to the current <em>Object</em>.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <em>Object</em>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <em>Object</em>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <em>Type</em> of the current instance.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <em>Object</em>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a <em>String</em> that represents the current <em>Object</em>.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <em>Object</em>.)</td>
</tr>
</tbody>
</table>
See Also

DatabaseEntry Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Release the resources held by the underlying C library.

**Namespace:**  BerkeleyDB  
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public virtual void Dispose()

Visual Basic (Declaration)

Public Overridable Sub Dispose

Visual C++

public:
virtual void Dispose()

Implements

IDisposable:::Dispose()()
See Also

DatabaseEntry Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `DatabaseEntry` type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>The byte string stored in or retrieved from a database</td>
</tr>
</tbody>
</table>
See Also

DatabaseEntry Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The byte string stored in or retrieved from a database

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public byte[] Data { get; set; }

Visual Basic (Declaration)

Public Property Data As Byte()

Visual C++

public:
property array<unsigned char>^ Data {
    array<unsigned char>^ get ();
    void set (array<unsigned char>^ value);
}
See Also

DatabaseEntry Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
A class representing a Berkeley DB database environment - a collection including support for some or all of caching, locking, logging and transaction subsystems, as well as databases and log files.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public class DatabaseEnvironment

Visual Basic (Declaration)

Public Class DatabaseEnvironment

Visual C++

public ref class DatabaseEnvironment
Inheritance Hierarchy

System::Object
BerkeleyDB::DatabaseEnvironment
See Also

DatabaseEnvironment Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `DatabaseEnvironment` type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArchivableDatabaseFiles</td>
<td>The database files that need to be archived in order to recover the database from catastrophic failure. If any of the database files have not been accessed during the lifetime of the current log files, they will not be included in this list. It is also possible that some of the files referred to by the log have since been deleted from the system.</td>
</tr>
<tr>
<td>ArchivableLogFiles</td>
<td>The names of all of the log files that are no longer in use (for example, that are no longer involved in active transactions), and that may safely be archived, used for catastrophic recovery and then removed from the system.</td>
</tr>
<tr>
<td>BeginCDSGroup</td>
<td>Allocate a locker ID in an environment configured for Berkeley DB Concurrent Data Store applications.</td>
</tr>
<tr>
<td>BeginTransaction</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>Checkpoint</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>Close</td>
<td>Close the Berkeley DB environment, freeing any allocated resources and closing any underlying subsystems.</td>
</tr>
<tr>
<td>DetectDeadlocks</td>
<td>Run one iteration of the deadlock detector. The deadlock detector traverses the lock table and marks one of the participating lock requesters for rejection in each deadlock it finds.</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified Object is equal to the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>FailCheck</td>
<td>Check for threads of control (either a true thread or a process) that have exited while manipulating DB library data structures, while holding a logical database lock, or with an unresolved transaction. A transaction that was never aborted or committed.</td>
</tr>
</tbody>
</table>
- **GetHashCode**
  Serves as a hash function for a particular type (Inherited from `Object`.)

- **GetType**
  Gets the `Type` of the current instance. (Inherited from `Object`.)

- **LockingSystemStats**
  Overloaded.

- **LogFile**
  Map an LSN object to a log filename

- **LogFiles**
  The names of all of the log files

- **LogFlush**
  Overloaded.

- **LoggingSystemStats**
  Overloaded.

- **LogWrite**
  Append a record to the log

- **MPoolSystemStats**
  Overloaded.

- **MutexSystemStats**
  Overloaded.

- **Open**
  Instantiate a new DatabaseEnvironment object and open the Berkeley DB environment represented by `home`.
  Set the panic state for the database environment. (Database environments in a panic state normally refuse all attempts to call Berkeley DB functions, throwing `RunRecoveryException`.)

- **Panic**

- **PrintLockingSystemStats**
  Overloaded.

- **PrintLoggingSystemStats**
  Overloaded.

- **PrintMPoolSystemStats**
  Overloaded.

- **PrintMutexSystemStats**
  Overloaded.

- **PrintReplicationSystemStats**
  Overloaded.

- **PrintRepMgrSystemStats**
  Overloaded.

- **PrintStats**
  Overloaded.

- **PrintSubsystemStats**
  Overloaded.

- **PrintTransactionSystemStats**
  Overloaded.

- **Recover**
  Restore transactions that were prepared, but not resolved at the time of the system shut down to their state prior to the shut down or crash, including any locks previously held.
  Overloaded.

  The environment regions, including any back
are removed. Any log or database files and the environment directory are not removed.

If there are processes that have called \texttt{Open(String, DatabaseEnvironmentConfig)} without calling \texttt{Close} (that is, there are processes currently using the environment), Remove will fail without further action.

Calling Remove should not be necessary for most applications because the Berkeley DB environment is cleaned up as part of normal database recovery procedures. However, applications may want to call Remove as part of application shut down to free system resources. For example, if \texttt{SystemMemory} specified to \texttt{Open(String, DatabaseEnvironmentConfig)}, it may be useful to call Remove in order to release system shared memory segments that have been allocated. Or, on architectures in which mutexes require allocation of underlying system resources, it may be useful to call Remove in order to release those resources. Alternatively, if recovery is not required because no database state is maintained across failures, and no system resources need to be released, it is possible to clean up an environment by simply removing all the Berkeley DB files in the database environment's directories.

In multithreaded applications, only a single thread may call Remove.

- **RemoveDB**
  Overloaded.
  Remove log files that are no longer needed. Automatic log file removal is likely to make catastrophic recovery impossible.

- **RemoveUnusedLogFiles**
  Overloaded.

- **RenameDB**
  Overloaded.

RepHoldElection is not called by most replication applications. It should only be called by applications.
implementing their own network transport layer explicitly holding replication group elections handling replication messages outside of the replication manager framework.

If the election is successful, Berkeley DB will notify the application of the results of the election by raising either the `REP_ELECTED` or `REP_NEWMASTER` events (see `EventNotify` for more information). The application is responsible for adjusting its relationship to the other database environments in the replication group, including directing all database updates to the newly selected master, in accordance with the results of the election.

The thread of control that calls `RepHoldElection` must not be the thread of control that processes incoming messages; processing the incoming messages is necessary to successfully complete an election.

Before calling this method, the `RepTransport` delegate must already have been configured to send replication messages.

There are two ways to build Berkeley DB replication applications: the most common approach is to use the Berkeley DB library "replication manager" support, where the Berkeley DB library manages the replication group, including network transport, all replication message processing and acknowledgment, and elections. Applications using the replication manager support generally make the following calls:

1. Configure the local site in the replication group using `RepMgrLocalSite`. 
2. Call `RepMgrAddRemoteSite(ReplicationHostAddress)` to configure the remote site(s) in the replication group.

3. Configure the message acknowledgment policy (`RepMgrAckPolicy`) which provides the replication group's transactional needs.

4. Configure the local site's election priority (`RepPriority`).

5. Call `RepMgrStartClient(Int32)` or `RepMgrStartMaster(Int32)` to start the replication application.

For more information on building replication applications, please see the Replication Getting Started Guide included in the Berkeley DB documentation.

Applications with special needs (for example, applications using network protocols not supported by the Berkeley DB replication manager), must perform additional configuration and call other Berkeley DB replication methods. For more information on building advanced replication applications, please see the Replication API section in the Berkeley DB Programmer’s Reference Guide for more information.

Starting the replication manager consists of opening the TCP/IP listening socket to accept incoming connections, and starting all necessary background threads. When multiple processes share a database environment, only one process can open the listening socket; `RepMgrStartClient(Int32)` (and `RepMgrStartMaster(Int32)`) automatically opens the socket in the first process to call it, and skips it in the later calls from other processes.

Start the replication manager as a master site, not call for an election.

Overloaded.
- **RepProcessMessage**
  Process an incoming replication message sent by a member of the replication group to the local database environment.

- **RepSetClockskew**
  Set the clock skew ratio among replication group members based on the fastest and slowest measurements among the group for use with master leases.
  Set a threshold for the minimum and maximum that a client waits before requesting retransmission of a missing message.

- **RepSetRetransmissionRequest**
  Set a byte-count limit on the amount of data to be transmitted from a site in response to a single message processed by `RepProcessMessage(DatabaseEntry, DatabaseEntry, Int32)`. The limit is not a hard limit, and the record that exceeds the limit is the last record to be sent.

- **RepSetTransmitLimit**
  Initialize the communication infrastructure for a database environment participating in a replicated application.

- **RepSetTransport**
  Overloaded.

  RepStartClient is not called by most replication applications. It should only be called by applications implementing their own network transport layer, explicitly holding replication group elections, and handling replication messages outside of the replication manager framework.

  Replication master environments are the only environments where replicated databases may be modified. Replication client environments are read-only as long as they are clients. Replication client environments may be upgraded to be replication master environments in the case that the current master fails or there is no master present. If master leases are in use, this method cannot be used to appoint a new master and should only be used to configure a database environment as a master as the result of an election.
Before calling this method, the `RepTransport` must already have been configured to send replication messages.

**RepStartMaster**

RepStartMaster is not called by most replication applications. It should only be called by applications implementing their own network transport layer explicitly holding replication group elections handling replication messages outside of the replication manager framework.

Replication master environments are the only environments where replicated databases may be modified. Replication client environments are read-only as long as they are clients. Replication client environments may be upgraded to be replication master environments in the case that the current master fails or there is no master present. If master leases are in use, this method cannot be used to appoint a master and should only be used to configure a database environment as a master as the result of an election.

Before calling this method, the `RepTransport` must already have been configured to send replication messages.

**RepSync**

Force master synchronization to begin for this client.

**ResetFileID**

Allow database files to be copied, and then used in the same database environment as the original.

**ResetLSN**

Allow database files to be moved from one transactional database environment to another.

**SetMaxSequentialWrites**

Limit the number of sequential write operations scheduled by the library when flushing dirty pages from the cache.

**SyncMemPool**

Overloaded.

Returns a `String` that represents the current Object.
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ToString</strong></td>
<td>(Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>TransactionSystemStats</strong></td>
<td>Overloaded. Ensure that a specified percent of the pages in the cache are clean, by writing dirty pages to their backing files.</td>
</tr>
<tr>
<td><strong>TrickleCleanMemPool</strong></td>
<td>Overloaded. WriteToLog allows applications to include information in the database environment log files, for later review using the db_printlog utility. This method is intended for debugging and performance tuning.</td>
</tr>
<tr>
<td><strong>WriteToLog</strong></td>
<td>WriteToLog allows applications to include information in the database environment log files, for later review using the db_printlog utility. This method is intended for debugging and performance tuning.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>AutoCommit</td>
<td>If true, database operations for which no explicit transaction handle was specified, and which modify databases in the database environment, will be automatically enclosed within a transaction.</td>
</tr>
<tr>
<td>CacheSize</td>
<td>The size of the shared memory buffer pool -- that is, the cache.</td>
</tr>
<tr>
<td>CDB_ALLDB</td>
<td>If true, Berkeley DB Concurrent Data Store applications will perform locking on an environment-wide basis rather than on a per-database basis.</td>
</tr>
<tr>
<td>Create</td>
<td>If true, Berkeley DB subsystems will create any underlying files, as necessary.</td>
</tr>
<tr>
<td>DataDirs</td>
<td>The array of directories where database files are stored.</td>
</tr>
<tr>
<td>DeadlockResolution</td>
<td>The deadlock detector configuration, specifying what lock request(s) should be rejected. As transactions acquire locks on behalf of a single locker ID, rejecting a lock request associated with a transaction normally requires the transaction be aborted.</td>
</tr>
<tr>
<td>EncryptAlgorithm</td>
<td>The algorithm used by the Berkeley DB library to perform encryption and decryption.</td>
</tr>
<tr>
<td>ErrorFeedback</td>
<td>The mechanism for reporting detailed error messages to the application.</td>
</tr>
<tr>
<td>ErrorPrefix</td>
<td>The prefix string that appears before error messages issued by Berkeley DB.</td>
</tr>
<tr>
<td><strong>EventNotify</strong></td>
<td>A delegate which is called to notify the process of specific Berkeley DB events.</td>
</tr>
<tr>
<td><strong>Feedback</strong></td>
<td>Monitor progress within long running operations. If true, flush database writes to the backing disk before returning from the write system call, rather than flushing database writes explicitly in a separate system call, as necessary.</td>
</tr>
<tr>
<td><strong>ForceFlush</strong></td>
<td>If true, the object is free-threaded; that is, concurrently usable by multiple threads in the address space.</td>
</tr>
<tr>
<td><strong>FreeThreaded</strong></td>
<td>The database environment home directory.</td>
</tr>
<tr>
<td><strong>Home</strong></td>
<td>If true, Berkeley DB will page-fault shared regions into memory when initially creating or joining a Berkeley DB environment.</td>
</tr>
<tr>
<td><strong>InitRegions</strong></td>
<td>The intermediate directory permissions.</td>
</tr>
<tr>
<td><strong>IntermediateDirMode</strong></td>
<td>The current lock conflicts array.</td>
</tr>
<tr>
<td><strong>LockConflictMatrix</strong></td>
<td>If true, lock shared Berkeley DB environment files and memory-mapped databases into memory.</td>
</tr>
<tr>
<td><strong>Lockdown</strong></td>
<td>The number of lock table partitions used in the Berkeley DB environment.</td>
</tr>
<tr>
<td><strong>LockPartitions</strong></td>
<td>A value, in microseconds, representing lock timeouts.</td>
</tr>
<tr>
<td><strong>LockTimeout</strong></td>
<td>If true, Berkeley DB will automatically remove log files that are no longer needed.</td>
</tr>
<tr>
<td><strong>LogAutoRemove</strong></td>
<td>The size of the in-memory log buffer, in bytes</td>
</tr>
<tr>
<td><strong>LogBufferSize</strong></td>
<td>The path of a directory to be used as the location of logging files. Log files</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>LogDir</td>
<td>The directory created by the Log Manager subsystem will be created in this directory.</td>
</tr>
<tr>
<td>LogFileMode</td>
<td>The absolute file mode for created log files. This property is only useful for the rare Berkeley DB application that does not control its umask value.</td>
</tr>
<tr>
<td>LogForceSync</td>
<td>If true, Berkeley DB will flush log writes to the backing disk before returning from the write system call, rather than flushing log writes explicitly in a separate system call, as necessary.</td>
</tr>
<tr>
<td>LogInMemory</td>
<td>If true, transaction logs are maintained in memory rather than on disk. This means that transactions exhibit the ACI (atomicity, consistency, and isolation) properties, but not D (durability).</td>
</tr>
<tr>
<td>LogNoBuffer</td>
<td>If true, system buffering is turned off for Berkeley DB log files to avoid double caching.</td>
</tr>
<tr>
<td>LogRegionSize</td>
<td>The size of the underlying logging area of the Berkeley DB environment, in bytes.</td>
</tr>
<tr>
<td>LogZeroOnCreate</td>
<td>If true, all pages of a log file are zeroed when that log file is created.</td>
</tr>
<tr>
<td>MaxCacheSize</td>
<td>The maximum cache size</td>
</tr>
<tr>
<td>MaxLockers</td>
<td>The maximum number of locking entities supported by the Berkeley DB environment.</td>
</tr>
<tr>
<td>MaxLocks</td>
<td>The maximum number of locks supported by the Berkeley DB environment.</td>
</tr>
<tr>
<td>MaxLogFileSize</td>
<td>The maximum size of a single file in the log, in bytes. Because LSN Offsets are unsigned four-byte values, the size may not be larger than the maximum</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Max Mutexes</strong></td>
<td>The total number of mutexes allocated</td>
</tr>
<tr>
<td><strong>Max Objects</strong></td>
<td>The maximum number of locked objects</td>
</tr>
<tr>
<td><strong>Max Open Files</strong></td>
<td>The number of file descriptors the library will open concurrently when flushing dirty pages from the cache.</td>
</tr>
<tr>
<td><strong>Max Sequential Writes</strong></td>
<td>The number of sequential write operations scheduled by the library when flushing dirty pages from the cache.</td>
</tr>
<tr>
<td><strong>Max Transactions</strong></td>
<td>The number of active transactions supported by the environment. This value bounds the size of the memory allocated for transactions. Child transactions are counted as active until they either commit or abort.</td>
</tr>
<tr>
<td><strong>MMap Size</strong></td>
<td>The maximum file size, in bytes, for a file to be mapped into the process address space. If no value is specified, it defaults to 10MB.</td>
</tr>
<tr>
<td><strong>Mutex Alignment</strong></td>
<td>The mutex alignment, in bytes.</td>
</tr>
<tr>
<td><strong>Mutex Increment</strong></td>
<td>The number of additional mutexes allocated.</td>
</tr>
<tr>
<td><strong>No Buffer</strong></td>
<td>If true, turn off system buffering of Berkeley DB database files to avoid double caching.</td>
</tr>
<tr>
<td><strong>No Locking</strong></td>
<td>If true, Berkeley DB will grant all requested mutual exclusion mutexes and database locks without regard for their actual availability. This functionality should never be used for purposes other than debugging.</td>
</tr>
<tr>
<td><strong>No MMap</strong></td>
<td>If true, Berkeley DB will copy read-only database files into the local cache instead of potentially mapping them</td>
</tr>
</tbody>
</table>
into process memory. If true, Berkeley DB will ignore any panic state in the database environment. (Database environments in a panic state normally refuse all attempts to call Berkeley DB functions, throwing `RunRecoveryException`.) This functionality should never be used for purposes other than debugging.

The number of times that test-and-set mutexes should spin without blocking.

**NoPanic**

The value defaults to 1 on uniprocessor systems and to 50 times the number of processors on multiprocessor systems.

**NumTestAndSetSpins**

If true, overwrite files stored in encrypted formats before deleting them.

**Overwrite**

If true, allocate region memory from the heap instead of from memory backed by the filesystem or system shared memory.

**Private**

If true, Berkeley DB will have checked to see if recovery needed to be performed before opening the database environment.

**Register**

The amount of time the replication manager's transport function waits to collect enough acknowledgments from replication group clients, before giving up and returning a failure indication.

**RepAckTimeout**

The default wait time is 1 second.

If true, the replication master sends groups of records to the clients in a single network transfer.

**RepBulkTransfer**

The amount of time a master site will delay between completing a

**RepCheckpointDelay**
checkpoint and writing a checkpoint record into the log.

The value, relative to **RepClockskewSlow**, of the fastest clock in the group of sites.

**RepClockskewSlow**

The value of the slowest clock in the group of sites.

**RepConnectionRetry**

The amount of time the replication manager will wait before trying to re-establish a connection to another site after a communication failure. The default wait time is 30 seconds.

If true, the client should delay synchronizing to a newly declared master (defaults to false). Clients configured in this way will remain unsynchronized until the application calls `RepSync()`. Configure the amount of time the replication manager will wait before retrying a failed election. The default wait time is 10 seconds.

**RepElectionRetry**

The timeout period for an election. The default timeout is 2 seconds.

An optional configuration timeout period to wait for full election participation the first time the replication group finds a master. By default this option is turned off and normal election timeouts are used. (See the Elections section in the Berkeley DB Reference Guide for more information.)

The amount of time the replication manager, running at a client site, waits for some message activity on the
<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RepHeartbeatMonitor</td>
<td>The frequency at which the replication manager, running at a master site, broadcasts a heartbeat message in an otherwise idle system. When 0 (the default), no heartbeat messages will be sent.</td>
</tr>
<tr>
<td>RepHeartbeatSend</td>
<td>The amount of time a client grants its master lease to a master. When using master leases all sites in a replication group must use the same lease timeout value. There is no default value. If leases are desired, this method must be called prior to calling RepStartClient() or RepStartMaster().</td>
</tr>
<tr>
<td>RepLeaseTimeout</td>
<td>The amount of time a client grants its master lease to a master. When using master leases all sites in a replication group must use the same lease timeout value. There is no default value. If leases are desired, this method must be called prior to calling RepStartClient() or RepStartMaster().</td>
</tr>
<tr>
<td>RepMgrAckPolicy</td>
<td>Specify how master and client sites will handle acknowledgment of replication messages which are necessary for &quot;permanent&quot; records. The current implementation requires all sites in a replication group configure the same acknowledgement policy.</td>
</tr>
<tr>
<td>RepMgrLocalSite</td>
<td>The host information for the local system.</td>
</tr>
<tr>
<td>RepMgrRemoteSites</td>
<td>The status of the sites currently known by the replication manager. If true, the replication master will not automatically re-initialize outdated clients (defaults to false).</td>
</tr>
<tr>
<td>RepNoAutoInit</td>
<td>If true, Berkeley DB method calls that would normally block while clients are</td>
</tr>
</tbody>
</table>
**RepNoBlocking**
in recovery will return errors immediately (defaults to false).

**RepNSites**
The total number of sites in the replication group.

**RepPriority**
The database environment's priority in replication group elections. A special value of 0 indicates that this environment cannot be a replication group master. If not configured, then a default value of 100 is used.

**RepRetransmissionRequestMax**
The maximum number of microseconds a client waits before requesting retransmission.

**RepRetransmissionRequestMin**
The minimum number of microseconds a client waits before requesting retransmission.

**RepStrict2Site**
Replication Manager observes the strict "majority" rule in managing elections, even in a group with only 2 sites. This means the client in a 2-site group will be unable to take over as master if the original master fails or becomes disconnected. (See the Elections section in the Berkeley DB Reference Guide for more information.) Both sites in the replication group should have the same value for this parameter.

**RepTransmitLimitBytes**
The bytes component of the byte-count limit on the amount of data that will be transmitted from a site in response to a single message processed by `RepProcessMessage(DatabaseEntry, DatabaseEntry, Int32)`.

The gigabytes component of the byte-count limit on the amount of data that will be transmitted from a site in
<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RepTransmitLimitGBytes</td>
<td>response to a single message processed by <code>RepProcessMessage(DatabaseEntry, DatabaseEntry, Int32)</code></td>
</tr>
<tr>
<td>RepTransport</td>
<td>The delegate used to transmit data using the replication application's communication infrastructure.</td>
</tr>
<tr>
<td>RepUseMasterLeases</td>
<td>If true, master leases will be used for this site (defaults to false).</td>
</tr>
<tr>
<td>RunFatalRecovery</td>
<td>If true, catastrophic recovery was run on this environment before opening it for normal use.</td>
</tr>
<tr>
<td>RunRecovery</td>
<td>If true, normal recovery was run on this environment before opening it for normal use.</td>
</tr>
<tr>
<td>SequentialWritePause</td>
<td>The number of microseconds the thread of control will pause before scheduling further write operations.</td>
</tr>
<tr>
<td>SetThreadID</td>
<td>A delegate that returns a unique identifier pair for the current thread of control.</td>
</tr>
<tr>
<td>SetThreadName</td>
<td>A delegate that formats a process ID and thread ID identifier pair.</td>
</tr>
<tr>
<td>SystemMemory</td>
<td>If true, allocate region memory from system shared memory instead of from heap memory or memory backed by the filesystem.</td>
</tr>
<tr>
<td>TempDir</td>
<td>The path of a directory to be used as the location of temporary files.</td>
</tr>
<tr>
<td>ThreadCount</td>
<td>An approximate number of threads in the database environment.</td>
</tr>
<tr>
<td>ThreadIsAlive</td>
<td>A delegate that returns if a thread of control (either a true thread or a process) is still running. If true, database calls timing out based on lock or transaction timeout values</td>
</tr>
<tr>
<td>Variable</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>TimeNotGranted</strong></td>
<td>will throw <strong>LockNotGrantedException</strong> instead of <strong>DeadlockException</strong>.</td>
</tr>
<tr>
<td><strong>TxnNoSync</strong></td>
<td>If true, Berkeley DB will not write or synchronously flush the log on transaction commit.</td>
</tr>
<tr>
<td></td>
<td>If true and a lock is unavailable for any Berkeley DB operation performed in the context of a transaction, cause the operation to throw <strong>DeadlockException</strong> (or <strong>LockNotGrantedException</strong> if configured with <strong>TimeNotGranted</strong>).</td>
</tr>
<tr>
<td><strong>TxnNoWait</strong></td>
<td>If true, all transactions in the environment will be started as if <strong>Snapshot</strong> was passed to <strong>BeginTransaction</strong>(). and all non-transactional cursors will be opened as if <strong>SnapshotIsolation</strong> was passed to <strong>Cursor</strong>().</td>
</tr>
<tr>
<td><strong>TxnSnapshot</strong></td>
<td>A value, in microseconds, representing transaction timeouts.</td>
</tr>
<tr>
<td><strong>TxnTimestamp</strong></td>
<td>The recovery timestamp</td>
</tr>
<tr>
<td><strong>TxnWriteNoSync</strong></td>
<td>If true, Berkeley DB will write, but will not synchronously flush, the log on transaction commit.</td>
</tr>
<tr>
<td></td>
<td>The Berkeley DB process' environment may be permitted to specify information to be used when naming files; see Berkeley DB File Naming in the Programmer's Reference Guide for more information.</td>
</tr>
<tr>
<td><strong>UseEnvironmentVars</strong></td>
<td>If true, all databases in the environment will be opened as if <strong>UseMVCC</strong> was set.</td>
</tr>
<tr>
<td><strong>UseMVCC</strong></td>
<td>If true, locking for the Berkeley DB Concurrent Data Store product was initialized.</td>
</tr>
<tr>
<td>Feature</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>UsingLocking</td>
<td>If true, the locking subsystem was initialized.</td>
</tr>
<tr>
<td>UsingLogging</td>
<td>If true, the logging subsystem was initialized.</td>
</tr>
<tr>
<td>UsingMPool</td>
<td>If true, the shared memory buffer pool subsystem was initialized.</td>
</tr>
<tr>
<td>UsingReplication</td>
<td>If true, the replication subsystem was initialized.</td>
</tr>
<tr>
<td>UsingTxns</td>
<td>If true, the transaction subsystem was initialized.</td>
</tr>
<tr>
<td>Verbosity</td>
<td>Specific additional informational and debugging messages in the Berkeley DB message output.</td>
</tr>
<tr>
<td>YieldCPU</td>
<td>If true, Berkeley DB will yield the processor immediately after each page or mutex acquisition.</td>
</tr>
</tbody>
</table>
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The DatabaseEnvironment type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArchivableDatabaseFiles</td>
<td>The database files that need to be archived in order to recover the database from catastrophic failure. If any of the database files have not been accessed during the lifetime of the current log files, they will not be included in this list. It is also possible that some of the files referred to by the log have since been deleted. The names of all of the log files that are no longer in use (for example, that are no longer involved in transactions), and that may safely be archived for catastrophic recovery and then removed from the system.</td>
</tr>
<tr>
<td>ArchivableLogFiles</td>
<td>Allocate a locker ID in an environment configured for Berkeley DB Concurrent Data Store applications. The names of all of the log files that are no longer in use (for example, that are no longer involved in transactions), and that may safely be archived for catastrophic recovery and then removed from the system.</td>
</tr>
<tr>
<td>BeginCDSGroup</td>
<td>BeginTransaction is overloaded.</td>
</tr>
<tr>
<td>BeginTransaction</td>
<td>Close the Berkeley DB environment, freeing allocated resources and closing any underlying subsystems.</td>
</tr>
<tr>
<td>Checkpoint</td>
<td>Close the Berkeley DB environment, freeing allocated resources and closing any underlying subsystems.</td>
</tr>
<tr>
<td>Close</td>
<td>Close the Berkeley DB environment, freeing allocated resources and closing any underlying subsystems.</td>
</tr>
<tr>
<td>DetectDeadlocks</td>
<td>Run one iteration of the deadlock detector. The deadlock detector traverses the lock table and marks one of the participating lock requesters for rejection in each deadlock it finds.</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified Object is equal to the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>FailCheck</td>
<td>Check for threads of control (either a true thread or a process) that have exited while manipulating Berkeley DB library data structures, while holding a logical database lock, or with an unresolved transaction. A transaction that was never aborted or committed.</td>
</tr>
</tbody>
</table>
GetHashCode
Serves as a hash function for a particular type (Inherited from Object.)

GetType
Gets the Type of the current instance. (Inherited from Object.)

LockingSystemStats
Overloaded.

LogFile
Map an LSN object to a log filename

LogFiles
The names of all of the log files

LogFlush
Overloaded.

LoggingSystemStats
Overloaded.

LogWrite
Append a record to the log

MPoolSystemStats
Overloaded.

MutexSystemStats
Overloaded.

Open
Instantiate a new DatabaseEnvironment object and open the Berkeley DB environment represented by home.

Set the panic state for the database environment. (Database environments in a panic state normally refuse all attempts to call Berkeley DB functions, throwing RunRecoveryException.)

PrintLockingSystemStats
Overloaded.

PrintLoggingSystemStats
Overloaded.

PrintMPoolSystemStats
Overloaded.

PrintMutexSystemStats
Overloaded.

PrintReplicationSystemStats
Overloaded.

PrintRepMgrSystemStats
Overloaded.

PrintStats
Overloaded.

PrintSubsystemStats
Overloaded.

PrintTransactionSystemStats
Overloaded.

Recover
Restore transactions that were prepared, but not resolved at the time of the system shut down to their state prior to the shut down or crash, including any locks previously held.

Overloaded.

The environment regions, including any back
are removed. Any log or database files and the environment directory are not removed.

If there are processes that have called `Open(String, DatabaseEnvironmentConfig)` without calling `()` (that is, there are processes currently using the environment), `Remove` will fail without further action.

Calling `Remove` should not be necessary for most applications because the Berkeley DB environment is cleaned up as part of normal database recovery procedures. However, applications may want to call `Remove` as part of application shut down to free up system resources. For example, if `SystemMemory` specified to `Open(String, DatabaseEnvironmentConfig)`, it may be useful to call `Remove` in order to release system shared memory segments that have been allocated. Or, on architectures in which mutexes require allocation of underlying system resources, it may be useful to call `Remove` in order to release those resources. Alternatively, if recovery is not required because no database state is maintained across failures, and no system resources need to be released, it is possible to clean up an environment by simply removing all the Berkeley DB files in the database environment's directories.

In multithreaded applications, only a single thread should call `Remove`.

- **RemoveDB**
  - Overloaded.
  - Remove log files that are no longer needed. Automatic log file removal is likely to make catastrophic recovery impossible.

- **RemoveUnusedLogFiles**
  - Overloaded.

- **RenameDB**
  - Overloaded.

`RepHoldElection` is not called by most replication applications. It should only be called by applications with a need to elect a new primary.
implementing their own network transport layer explicitly holding replication group elections handling replication messages outside of the replication manager framework.

If the election is successful, Berkeley DB will notify the application of the results of the election by either the REP_ELECTED or REP_NEW_MASTER events (see EventNotify for more information). The application is responsible for adjusting its relationship to the other database environments in the replication group, including directing all database update messages to the newly selected master, in accordance with the results of the election.

The thread of control that calls RepHoldElection must not be the thread of control that processes incoming messages; processing the incoming messages is necessary to successfully complete an election.

Before calling this method, the RepTransport delegate must already have been configured to send replication messages.

There are two ways to build Berkeley DB replication applications: the most common approach is to use the Berkeley DB library "replication manager" support where the Berkeley DB library manages the replication group, including network transport, all replication message processing and acknowledgment, and group elections. Applications using the replication manager support generally make the following calls:

1. Configure the local site in the replication group using RepMgrLocalSite.
2. Call `RepMgrAddRemoteSite(ReplicationHostAddress)` to configure the remote site(s) in the replication group.

3. Configure the message acknowledgment (RepMgrAckPolicy) which provides the replication group's transactional needs.

4. Configure the local site's election priority; `RepPriority`.

5. Call `RepMgrStartClient(Int32)` or `RepMgrStartMaster(Int32)` to start the replication application.

For more information on building replication applications, please see the Replication Getting Started Guide included in the Berkeley DB documentation.

Applications with special needs (for example, applications using network protocols not supported by the Berkeley DB replication manager), must perform additional configuration and call other Berkeley DB replication methods. For more information on building advanced replication applications, please see the Replication API section in the Berkeley DB Programmer's Reference Guide for more information.

Starting the replication manager consists of opening the TCP/IP listening socket to accept incoming connections, and starting all necessary background threads. When multiple processes share a database environment, only one process can open the listening socket; `RepMgrStartClient(Int32)` (and `RepMgrStartMaster(Int32)`) automatically open the socket in the first process to call it, and skips in the later calls from other processes.

Start the replication manager as a master site, not call for an election.

Overloaded.
- **RepProcessMessage**
  Process an incoming replication message sent by a member of the replication group to the local database environment.

- **RepSetClockskew**
  Set the clock skew ratio among replication group members based on the fastest and slowest measurements among the group for use with master leases.

- **RepSetRetransmissionRequest**
  Set a threshold for the minimum and maximum that a client waits before requesting retransmission of a missing message.

- **RepSetTransmitLimit**
  Set a byte-count limit on the amount of data to be transmitted from a site in response to a single message processed by `RepProcessMessage(DatabaseEntry, DatabaseEntry, Int32)`. The limit is not a hard limit, and the record that exceeds the limit is the last record to be sent.

- **RepSetTransport**
  Initialize the communication infrastructure for a database environment participating in a replicated application.

Overloaded.

RepStartClient is not called by most replication applications. It should only be called by applications implementing their own network transport layer explicitly holding replication group elections and handling replication messages outside of the replication manager framework.

Replication master environments are the only environments where replicated databases may be modified. Replication client environments are read-only as long as they are clients. Replication client environments may be upgraded to be replication master environments in the case that the current master fails or there is no master present. If master leases are in use, this method cannot be used to appoint a master and should only be used to configure a database environment as a master as the result of an election.
Before calling this method, the RepTransport delegate must already have been configured to send replication messages.

Overloaded.

RepStartMaster is not called by most replication applications. It should only be called by applications implementing their own network transport layer explicitly holding replication group elections handling replication messages outside of the replication manager framework.

Replication master environments are the only environments where replicated databases may be modified. Replication client environments are read-only as long as they are clients. Replication client environments may be upgraded to be replication master environments in the case that the current master fails or there is no master present. If master leases are in use, this method cannot be used to appoint a new master, and should only be used to configure a database environment as a master as the result of an election.

Before calling this method, the RepTransport delegate must already have been configured to send replication messages.

- RepStartMaster
- RepSync
- ResetFileID
- ResetLSN
- SetMaxSequentialWrites
- SyncMemPool

Force master synchronization to begin for this client.

Allow database files to be copied, and then used in the same database environment as the original.

Allow database files to be moved from one transactional database environment to another.

Limit the number of sequential write operations scheduled by the library when flushing dirty pages from the cache.

Overloaded.

Returns a String that represents the current Object.
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ToString</strong></td>
<td>(Inherited from <a href="#">Object</a>).</td>
</tr>
<tr>
<td><strong>TransactionSystemStats</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>TrickleCleanMemPool</strong></td>
<td>Ensure that a specified percent of the pages in the cache are clean, by writing dirty pages to their backing files. Overloaded.</td>
</tr>
<tr>
<td><strong>WriteToLog</strong></td>
<td>WriteToLog allows applications to include information in the database environment log files, for later review using the db_printlog utility. This method is intended for debugging and performance tuning.</td>
</tr>
</tbody>
</table>
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The database files that need to be archived in order to recover the database from catastrophic failure. If any of the database files have not been accessed during the lifetime of the current log files, they will not included in this list. It is also possible that some of the files referred to by the log have since been deleted from the system.

**Namespace:** [BerkeleyDB](https://example.com)

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public List<string> ArchivableDatabaseFiles(
    bool AbsolutePaths
)

Visual Basic (Declaration)

Public Function ArchivableDatabaseFiles ( _
    AbsolutePaths As Boolean _
) As List(Of String)

Visual C++

public:
List<String>^ ArchivableDatabaseFiles(  
    bool AbsolutePaths
)

Parameters

AbsolutePaths
Type: System::Boolean
If true, all pathnames are returned as absolute pathnames, instead of relative to the database home directory.

Return Value

The database files that need to be archived in order to recover the database from catastrophic failure.
Remarks

See the db_archive utility for more information on database archival procedures.
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The names of all of the log files that are no longer in use (for example, that are no longer involved in active transactions), and that may safely be archived for catastrophic recovery and then removed from the system.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
### Syntax

**C#**

```csharp
public List<string> ArchivableLogFiles(
    bool AbsolutePaths
)
```

**Visual Basic (Declaration)**

```vbnet
Public Function ArchivableLogFiles ( _
    AbsolutePaths As Boolean _
) As List(Of String)
```

**Visual C++**

```cpp
public:
List<String^>^ ArchivableLogFiles(
    bool AbsolutePaths
)
```

### Parameters

**AbsolutePaths**

Type: System::::Boolean

If true, all pathnames are returned as absolute pathnames, instead of relative to the database home directory.

### Return Value

The names of all of the log files that are no longer in use
Remarks

The Berkeley DB interfaces to the database environment logging subsystem (for example, `Abort`) may allocate log cursors and have open file descriptors for log files as well. On operating systems where filesystem related system calls (for example, rename and unlink on Windows/NT) can fail if a process has an open file descriptor for the affected file, attempting to move or remove the log files listed by ArchivableLogFiles may fail. All Berkeley DB internal use of log cursors operates on active log files only and furthermore, is short-lived in nature. So, an application seeing such a failure should be restructured to retry the operation until it succeeds. (Although this is not likely to be necessary; it is hard to imagine a reason to move or rename a log file in which transactions are being logged or aborted.)

See the db_archive utility for more information on database archival procedures.
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Allocate a locker ID in an environment configured for Berkeley DB Concurrent Data Store applications.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
**Syntax**

**C#**

```csharp
public Transaction BeginCDSGroup()
```

**Visual Basic (Declaration)**

```vbnet
Public Function BeginCDSGroup As Transaction
```

**Visual C++**

```cpp
public:
Transaction^ BeginCDSGroup()
```

**Return Value**

A Transaction object that uniquely identifies the locker ID
Remarks

Calling \texttt{Commit()} will discard the allocated locker ID.

See Berkeley DB Concurrent Data Store applications in the Programmer's Reference Guide for more information about when this is required.
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation

DatabaseEnvironment...::BeginTransaction Method

DatabaseEnvironment Class  See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>BeginTransaction()</code></td>
<td>Create a new transaction in the environment, with the default configuration.</td>
</tr>
<tr>
<td><code>BeginTransaction(TransactionConfig)</code></td>
<td>Create a new transaction in the environment.</td>
</tr>
<tr>
<td><code>BeginTransaction(TransactionConfig, Transaction)</code></td>
<td>Create a new transaction in the environment.</td>
</tr>
</tbody>
</table>
See Also

DatabaseEnvironment Class
DatabaseEnvironment Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Create a new transaction in the environment, with the default configuration.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public Transaction BeginTransaction()

Visual Basic (Declaration)

Public Function BeginTransaction As Transaction

Visual C++

public:
Transaction^ BeginTransaction()

Return Value

A new transaction object
See Also

DatabaseEnvironment Class
BeginTransaction Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Create a new transaction in the environment.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

```csharp
public Transaction BeginTransaction(
    TransactionConfig cfg
)
```

Visual Basic (Declaration)

```vbnet
Public Function BeginTransaction (_
    As TransactionConfig _
) As Transaction
```

Visual C++

```cpp
public:
    Transaction^ BeginTransaction(
        TransactionConfig^ cfg
    )
```

Parameters

cfg

Type: BerkeleyDB::TransactionConfig
The configuration properties for the transaction

Return Value

A new transaction object
See Also

DatabaseEnvironment Class
BeginTransaction Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Create a new transaction in the environment.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

**C#**

```csharp
public Transaction BeginTransaction(
    TransactionConfig cfg,
    Transaction parent)
```

**Visual Basic (Declaration)**

```vbnet
Public Function BeginTransaction ( _
    cfg As TransactionConfig, _
    parent As Transaction _
) As Transaction
```

**Visual C++**

```cpp
public:
    Transaction^ BeginTransaction(
        TransactionConfig^ cfg,
        Transaction^ parent)
```

**Parameters**

cfg
   Type: BerkeleyDB::TransactionConfig
   The configuration properties for the transaction

parent
   Type: BerkeleyDB::Transaction
   If the non-null, the new transaction will be a nested transaction, with parent as the new transaction's parent. Transactions may be nested to any level.

**Return Value**

A new transaction object
Remarks

In the presence of distributed transactions and two-phase commit, only the parental transaction, that is a transaction without a parent specified, should be passed as an parameter to `Prepare(array<Byte>[][])`.
See Also

DatabaseEnvironment Class
beginTransaction Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
DatabaseEnvironment.Checkpoint Method

See Also

DatabaseEnvironment Class
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Checkpoint()()()</code></td>
<td>Flush the underlying memory pool, write a checkpoint record to the log, and then flush the log, even if there has been no activity since the last checkpoint. If there has been any logging activity in the database environment since the last checkpoint, flush the underlying memory pool, write a checkpoint record to the log, and then flush the log.</td>
</tr>
<tr>
<td><code>Checkpoint(UInt32, UInt32)</code></td>
<td></td>
</tr>
</tbody>
</table>
See Also

DatabaseEnvironment Class
DatabaseEnvironment Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Flush the underlying memory pool, write a checkpoint record to the log, and then flush the log, even if there has been no activity since the last checkpoint.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public void Checkpoint()

Visual Basic (Declaration)

Public Sub Checkpoint

Visual C++

public:
void Checkpoint()
See Also

DatabaseEnvironment Class
Checkpoint Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If there has been any logging activity in the database environment since the last checkpoint, flush the underlying memory pool, write a checkpoint record to the log, and then flush the log.

Namespace: BerkeleyDB
Syntax

C#

public void Checkpoint(
    uint kbytesWritten,
    uint minutesElapsed
)

Visual Basic (Declaration)

Public Sub Checkpoint ( _
    kbytesWritten As UInteger, _
    minutesElapsed As UInteger _
)

Visual C++

public:
void Checkpoint(
    unsigned int kbytesWritten,
    unsigned int minutesElapsed
)

Parameters

kbytesWritten
Type: System::UInt32
A checkpoint will be done if more than kbytesWritten kilobytes of log data have been written since the last checkpoint.

minutesElapsed
Type: System::UInt32
A checkpoint will be done if more than minutesElapsed minutes have passed since the last checkpoint.
See Also

DatabaseEnvironment Class
Checkpoint Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Close the Berkeley DB environment, freeing any allocated resources and closing any underlying subsystems.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public void Close()

Visual Basic (Declaration)

Public Sub Close

Visual C++

public:
void Close()
Remarks

The object should not be closed while any other handle that refers to it is not yet closed; for example, database environment handles must not be closed while database objects remain open, or transactions in the environment have not yet been committed or aborted.

Where the environment was configured with UseTxns, calling Close aborts any unresolved transactions. Applications should not depend on this behavior for transactions involving Berkeley DB databases; all such transactions should be explicitly resolved. The problem with depending on this semantic is that aborting an unresolved transaction involving database operations requires a database handle. Because the database handles should have been closed before calling Close, it will not be possible to abort the transaction, and recovery will have to be run on the Berkeley DB environment before further operations are done.

In multithreaded applications, only a single thread may call Close.
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
DatabaseEnvironment.DetectDeadlocks Method

Run one iteration of the deadlock detector. The deadlock detector traverses the lock table and marks one of the participating lock requesters for rejection in each deadlock it finds.

Namespace: BerkeleyDB
Syntax

C#

```csharp
public uint DetectDeadlocks(
    DeadlockPolicy atype
)
```

Visual Basic (Declaration)

```vbnet
Public Function DetectDeadlocks ( _
    atype As DeadlockPolicy _
) As UInteger
```

Visual C++

```cpp
public:
    unsigned int DetectDeadlocks(
        DeadlockPolicy^ atype
    )
```

Parameters

atype
Type: BerkeleyDB::::DeadlockPolicy
Specify which lock request(s) to reject

Return Value

The number of lock requests that were rejected.
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Check for threads of control (either a true thread or a process) that have exited while manipulating Berkeley DB library data structures, while holding a logical database lock, or with an unresolved transaction (that is, a transaction that was never aborted or committed).

Namespace: BerkeleyDB
**Syntax**

**C#**

```csharp
public void FailCheck()
```

**Visual Basic (Declaration)**

```vbnet
Public Sub FailCheck
```

**Visual C++**

```cpp
public:
    void FailCheck()
```
Remarks

For more information, see Architecting Data Store and Concurrent Data Store applications, and Architecting Transactional Data Store applications, both in the Berkeley DB Programmer's Reference Guide.

FailCheck is based on the SetThreadID and ThreadIsAlive delegates. Applications calling FailCheck must have already set ThreadIsAlive, and must have configured ThreadCount.

If FailCheck determines a thread of control exited while holding database read locks, it will release those locks. If FailCheck determines a thread of control exited with an unresolved transaction, the transaction will be aborted. In either of these cases, FailCheck will return successfully and the application may continue to use the database environment.

In either of these cases, FailCheck will also report the process and thread IDs associated with any released locks or aborted transactions. The information is printed to a specified output channel (see [!:MessageFile] for more information), or passed to an application delegate (see [!:MessageCall] for more information).

If FailCheck determines a thread of control has exited such that database environment recovery is required, it will throw RunRecoveryException. In this case, the application should not continue to use the database environment. For a further description as to the actions the application should take when this failure occurs, see Handling failure in Data Store and Concurrent Data Store applications, and Handling failure in Transactional Data Store applications, both in the Berkeley DB Programmer's Reference Guide.
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation

DatabaseEnvironment..:::LockingSystemStats Method

DatabaseEnvironment Class  See Also
### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>LockingSystemStats()</code></td>
<td>The locking subsystem statistics</td>
</tr>
<tr>
<td><code>LockingSystemStats(Boolean)</code></td>
<td>The locking subsystem statistics</td>
</tr>
</tbody>
</table>
See Also

DatabaseEnvironment Class  
DatabaseEnvironment Members  
BerkeleyDB Namespace  

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The locking subsystem statistics

**Namespace:**  [BerkeleyDB](https://www.berkeleydb.com)

**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
## Syntax

### C#

```csharp
public LockStats LockingSystemStats()
```

### Visual Basic (Declaration)

```vbnet
Public Function LockingSystemStats As LockStats
```

### Visual C++

```cpp
public:
LockStats^ LockingSystemStats()
```

## Return Value

The locking subsystem statistics
See Also

DatabaseEnvironment Class
LockingSystemStats Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The locking subsystem statistics

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

```csharp
public LockStats LockingSystemStats(
    bool clearStats
)
```

Visual Basic (Declaration)

```vbnet
Public Function LockingSystemStats ( _
    clearStats As Boolean _
) As LockStats
```

Visual C++

```cpp
public:
LockStats^ LockingSystemStats(
    bool clearStats
)
```

Parameters

clearStats
  Type: System:::Boolean
  If true, reset statistics after returning their values.

Return Value

The locking subsystem statistics
See Also

DatabaseEnvironment Class
LockingSystemStats Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
DatabaseEnvironment.LogFile Method

Map an LSN object to a log filename

Namespace: BerkeleyDB
Syntax

C#

public string LogFile(
    LSN logSeqNum
)

Visual Basic (Declaration)

Public Function LogFile ( _
    logSeqNum As LSN _
) As String

Visual C++

public:
String^ LogFile(
    LSN^ logSeqNum
)

Parameters

logSeqNum
    Type: BerkeleyDB::LSN
    The DB_LSN structure for which a filename is wanted.

Return Value

The name of the file containing the record named by logSeqNum.
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The names of all of the log files

Namespace: BerkeleyDB
Syntax

C#

public List<string> LogFiles(
    bool AbsolutePaths
)

Visual Basic (Declaration)

Public Function LogFiles ( _
    AbsolutePaths As Boolean _
) As List(Of String)

Visual C++

public:
    List<String^>^ LogFiles(
    bool AbsolutePaths
)

Parameters

AbsolutePaths
    Type: System:::Boolean
    If true, all pathnames are returned as absolute pathnames, instead of relative to the database home directory.

Return Value

All the log filenames, regardless of whether or not they are in use.
Remarks

The Berkeley DB interfaces to the database environment logging subsystem (for example, `Abort(0)` may allocate log cursors and have open file descriptors for log files as well. On operating systems where filesystem related system calls (for example, rename and unlink on Windows/NT) can fail if a process has an open file descriptor for the affected file, attempting to move or remove the log files listed by LogFiles may fail. All Berkeley DB internal use of log cursors operates on active log files only and furthermore, is short-lived in nature. So, an application seeing such a failure should be restructured to retry the operation until it succeeds. (Although this is not likely to be necessary; it is hard to imagine a reason to move or rename a log file in which transactions are being logged or aborted.)

See the db_archive utility for more information on database archival procedures.
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation
DatabaseEnvironment LogFlush Method

See Also

DatabaseEnvironment Class
# Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LogFlush()()()</td>
<td>Write all log records to disk.</td>
</tr>
<tr>
<td>LogFlush(LSN)</td>
<td>Write log records to disk.</td>
</tr>
</tbody>
</table>
See Also

DatabaseEnvironment Class
DatabaseEnvironment Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Write all log records to disk.

Namespace:  BerkeleyDB
Syntax

C#

public void LogFlush()

Visual Basic (Declaration)

Public Sub LogFlush

Visual C++

public:
void LogFlush()
See Also

DatabaseEnvironment Class
LogFlush Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Write log records to disk.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public void LogFlush(
    LSN logSeqNum
)

Visual Basic (Declaration)

Public Sub LogFlush (_
    logSeqNum As LSN _
)

Visual C++

public:
void LogFlush(
    LSN^ logSeqNum
)

Parameters

logSeqNum
Type: BerkeleyDB::LSN
All log records with LSN values less than or equal to logSeqNum are written to disk. If null, all records in the log are flushed.
See Also

DatabaseEnvironment Class
LogFlush Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation

DatabaseEnvironment::LoggingSystemStats Method

(DatabaseEnvironment Class) See Also
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>LoggingSystemStats()</code></td>
<td>The logging subsystem statistics</td>
</tr>
<tr>
<td><code>LoggingSystemStats(Boolean)</code></td>
<td>The logging subsystem statistics</td>
</tr>
</tbody>
</table>
See Also

DatabaseEnvironment Class
DatabaseEnvironment Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The logging subsystem statistics

**Namespace:**  BerkeleyDB
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
**Syntax**

**C#**

```csharp
public LogStats LoggingSystemStats()
```

**Visual Basic (Declaration)**

```vbnet
Public Function LoggingSystemStats As LogStats
```

**Visual C++**

```cpp
public:
LogStats^ LoggingSystemStats()
```

**Return Value**

The logging subsystem statistics
See Also

DatabaseEnvironment Class
LoggingSystemStats Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The logging subsystem statistics

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
### Syntax

**C#**

```csharp
public LogStats LoggingSystemStats(
    bool clearStats
)
```

**Visual Basic (Declaration)**

```vbnet
Public Function LoggingSystemStats ( _
    clearStats As Boolean _
) As LogStats
```

**Visual C++**

```cpp
public:
    LogStats^ LoggingSystemStats(
        bool clearStats
    )
```

### Parameters

`clearStats`

Type: `System::::Boolean`

If true, reset statistics after returning their values.

### Return Value

The logging subsystem statistics
See Also

DatabaseEnvironment Class
LoggingSystemStats Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Append a record to the log

**Namespace:**  [BerkeleyDB](http://example.com/BerkeleyDB)  
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
**Syntax**

**C#**

```csharp
public LSN LogWrite(
    DatabaseEntry dbt,
    bool flush
)
```

**Visual Basic (Declaration)**

```vbnet
Public Function LogWrite ( 
    dbt As DatabaseEntry, 
    flush As Boolean 
) As LSN
```

**Visual C++**

```cpp
public:
    LSN^ LogWrite(
        DatabaseEntry^ dbt,
        bool flush
    )
```

**Parameters**

dbt
Type: BerkeleyDB::DatabaseEntry
The record to write to the log.

flush
Type: System::Boolean
If true, the log is forced to disk after this record is written, guaranteeing that all records with LSN values less than or equal to the one being "put" are on disk before LogWrite returns.

**Return Value**
The LSN of the written record
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation

DatabaseEnvironment...:::MPoolSystemStats Method

DatabaseEnvironment Class  See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>MPoolSystemStats()</code></td>
<td>The memory pool (that is, the buffer cache) subsystem statistics</td>
</tr>
<tr>
<td><code>MPoolSystemStats(Boolean)</code></td>
<td>The memory pool (that is, the buffer cache) subsystem statistics</td>
</tr>
</tbody>
</table>
See Also

DatabaseEnvironment Class
DatabaseEnvironment Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The memory pool (that is, the buffer cache) subsystem statistics

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public MPoolStats MPoolSystemStats()

Visual Basic (Declaration)

Public Function MPoolSystemStats As MPoolStats

Visual C++

public:
MPoolStats^ MPoolSystemStats()

Return Value

The memory pool subsystem statistics
See Also

DatabaseEnvironment Class
MPoolSystemStats Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation
DatabaseEnvironment:::MPoolSystemStats Method (Boolean)

See Also

The memory pool (that is, the buffer cache) subsystem statistics

Namespace: BerkeleyDB
Syntax

C#

```csharp
public MPoolStats MPoolSystemStats(
    bool clearStats
)
```

Visual Basic (Declaration)

```vbnet
Public Function MPoolSystemStats ( _
    clearStats As Boolean _
) As MPoolStats
```

Visual C++

```cpp
public:
MPoolStats^ MPoolSystemStats(
    bool clearStats
)
```

Parameters

clearStats
  Type: System::::Boolean
  If true, reset statistics after returning their values.

Return Value

The memory pool subsystem statistics
See Also

DatabaseEnvironment Class
MPoolSystemStats Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
DatabaseEnvironment.

::

MutexSystemStats Method

See Also

DatabaseEnvironment Class

See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MutexSystemStats()()</td>
<td>The mutex subsystem statistics</td>
</tr>
<tr>
<td>MutexSystemStats(Boolean)</td>
<td>The mutex subsystem statistics</td>
</tr>
</tbody>
</table>
See Also

DatabaseEnvironment Class
DatabaseEnvironment Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation
DatabaseEnvironment...::.MutexSystemStats Method

DatabaseEnvironment Class  See Also

The mutex subsystem statistics

Namespace:  BerkeleyDB
### Syntax

**C#**

```csharp
public MutexStats MutexSystemStats()
```

**Visual Basic (Declaration)**

Public Function MutexSystemStats As MutexStats

**Visual C++**

```cpp
public:
MutexStats^ MutexSystemStats()
```

### Return Value

The mutex subsystem statistics
See Also

DatabaseEnvironment Class
MutexSystemStats Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The mutex subsystem statistics

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public MutexStats MutexSystemStats(
    bool clearStats
)

Visual Basic (Declaration)

Public Function MutexSystemStats ( _
    clearStats As Boolean _
) As MutexStats

Visual C++

public:
    MutexStats^ MutexSystemStats(
        bool clearStats
    )

Parameters

clearStats
    Type: System::::Boolean
    If true, reset statistics after returning their values.

Return Value

The mutex subsystem statistics
See Also

DatabaseEnvironment Class
MutexSystemStats Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Instantiate a new DatabaseEnvironment object and open the Berkeley DB environment represented by home.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public static DatabaseEnvironment Open(
        string home,
        DatabaseEnvironmentConfig cfg
    )

Visual Basic (Declaration)

Public Shared Function Open ( _
    home As String, _
    cfg As DatabaseEnvironmentConfig _
) As DatabaseEnvironment

Visual C++

public:
static DatabaseEnvironment^ Open(
        String^ home,
        DatabaseEnvironmentConfig^ cfg
    )

Parameters

home
    Type: System::String
    The database environment's home directory. For more information on home, and filename resolution in general, see Berkeley DB File Naming in the Programmer's Reference Guide.

cfg
    Type: BerkeleyDB::DatabaseEnvironmentConfig
    The environment's configuration

Return Value
A new, open DatabaseEnvironment object
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Set the panic state for the database environment. (Database environments in a panic state normally refuse all attempts to call Berkeley DB functions, throwing RunRecoveryException.)

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

**C#**

public void Panic()

**Visual Basic (Declaration)**

Public Sub Panic

**Visual C++**

public:
void Panic()
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation

DatabaseEnvironment:::PrintLockingSystemStats Method

DatabaseEnvironment Class   See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>PrintLockingSystemStats()()</code></td>
<td>Display the locking subsystem statistical information, as described by LockStats.</td>
</tr>
<tr>
<td><code>PrintLockingSystemStats(Boolean, Boolean)</code></td>
<td>Display the locking subsystem statistical information, as described by LockStats.</td>
</tr>
<tr>
<td><code>PrintLockingSystemStats(Boolean, Boolean, Boolean, Boolean, Boolean)</code></td>
<td>Display the locking subsystem statistical information, as described by LockStats.</td>
</tr>
</tbody>
</table>
See Also

DatabaseEnvironment Class
DatabaseEnvironment Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Display the locking subsystem statistical information, as described by LockStats.

**Namespace:**  BerkeleyDB  
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll)  
**Version:**  4.8.24.0
Syntax

C#

public void PrintLockingSystemStats()

Visual Basic (Declaration)

Public Sub PrintLockingSystemStats

Visual C++

public:
void PrintLockingSystemStats()
See Also

DatabaseEnvironment Class
PrintLockingSystemStats Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Display the locking subsystem statistical information, as described by LockStats.

Namespace: BerkeleyDB
**Syntax**

**C#**

```csharp
public void PrintLockingSystemStats(
    bool PrintAll,
    bool ClearStats
)
```

**Visual Basic (Declaration)**

```vbnet
Public Sub PrintLockingSystemStats ( _
    PrintAll As Boolean, _
    ClearStats As Boolean _
)
```

**Visual C++**

```cpp
public:
void PrintLockingSystemStats(  
    bool PrintAll,  
    bool ClearStats  
)
```

**Parameters**

**PrintAll**
- Type: `System::System::Boolean`
- If true, display all available information.

**ClearStats**
- Type: `System::System::Boolean`
- If true, reset statistics after displaying their values.
See Also

DatabaseEnvironment Class
PrintLockingSystemStats Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
DatabaseEnvironment...::PrintLockingSystemStats Method (Boolean, Boolean, Boolean, Boolean, Boolean, Boolean)

Display the locking subsystem statistical information, as described by LockStats.

Namespace: BerkeleyDB
Syntax

C#

public void PrintLockingSystemStats(
    bool PrintAll,
    bool ClearStats,
    bool ConflictMatrix,
    bool Lockers,
    bool Objects,
    bool Parameters
)

Visual Basic (Declaration)

Public Sub PrintLockingSystemStats (_
    PrintAll As Boolean, _
    ClearStats As Boolean, _
    ConflictMatrix As Boolean, _
    Lockers As Boolean, _
    Objects As Boolean, _
    Parameters As Boolean _
)

Visual C++

public:
void PrintLockingSystemStats(
    bool PrintAll,
    bool ClearStats,
    bool ConflictMatrix,
    bool Lockers,
    bool Objects,
    bool Parameters
)

Parameters

PrintAll
    Type: System::Boolean
    If true, display all available information.
ClearStats
Type: $\text{System} \ldots : \text{Boolean}$
If true, reset statistics after displaying their values.

ConflictMatrix
Type: $\text{System} \ldots : \text{Boolean}$
If true, display the lock conflict matrix.

Lockers
Type: $\text{System} \ldots : \text{Boolean}$
If true, Display the lockers within hash chains.

Objects
Type: $\text{System} \ldots : \text{Boolean}$
If true, display the lock objects within hash chains.

Parameters
Type: $\text{System} \ldots : \text{Boolean}$
If true, display the locking subsystem parameters.
See Also

DatabaseEnvironment Class
PrintLockingSystemStats Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation

DatabaseEnvironment....PrintLoggingSystemStats Method

DatabaseEnvironment Class  See Also
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PrintLoggingSystemStats()()</td>
<td>Display the logging subsystem statistical information, as described by LogStats.</td>
</tr>
<tr>
<td>PrintLoggingSystemStats(Boolean, Boolean)</td>
<td>Display the logging subsystem statistical information, as described by LogStats.</td>
</tr>
</tbody>
</table>
See Also

DatabaseEnvironment Class
DatabaseEnvironment Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Display the logging subsystem statistical information, as described by LogStats.

Namespace: BerkeleyDB
Syntax

C#

public void PrintLoggingSystemStats()

Visual Basic (Declaration)

Public Sub PrintLoggingSystemStats

Visual C++

public:
void PrintLoggingSystemStats()
See Also

DatabaseEnvironment Class
PrintLoggingSystemStats Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Display the logging subsystem statistical information, as described by LogStats.

Namespace: BerkeleyDB
Syntax

C#

```csharp
public void PrintLoggingSystemStats(
    bool PrintAll,
    bool ClearStats
)
```

Visual Basic (Declaration)

```vbnet
Public Sub PrintLoggingSystemStats (_
    PrintAll As Boolean, _
    ClearStats As Boolean _
)
```

Visual C++

```cpp
public:
void PrintLoggingSystemStats(
    bool PrintAll,
    bool ClearStats
)
```

Parameters

PrintAll
Type: `System::Boolean`
If true, display all available information.

ClearStats
Type: `System::Boolean`
If true, reset statistics after displaying their values.
See Also

DatabaseEnvironment Class
PrintLoggingSystemStats Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
DatabaseEnvironment Class  See Also

Berkeley DB .NET API Documentation
DatabaseEnvironment....:PrintMPoolSystemStats Method
### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>PrintMPoolSystemStats()</code></td>
<td>Display the memory pool (that is, buffer cache) subsystem statistical information, as described by <code>MPoolStats</code>.</td>
</tr>
<tr>
<td><code>PrintMPoolSystemStats(Boolean, Boolean)</code></td>
<td>Display the memory pool (that is, buffer cache) subsystem statistical information, as described by <code>MPoolStats</code>.</td>
</tr>
<tr>
<td><code>PrintMPoolSystemStats(Boolean, Boolean, Boolean)</code></td>
<td>Display the memory pool (that is, buffer cache) subsystem statistical information, as described by <code>MPoolStats</code>.</td>
</tr>
</tbody>
</table>
See Also

DatabaseEnvironment Class
DatabaseEnvironment Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Display the memory pool (that is, buffer cache) subsystem statistical information, as described by `MPoolStats`.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public void PrintMPoolSystemStats()

Visual Basic (Declaration)

Public Sub PrintMPoolSystemStats

Visual C++

public:
void PrintMPoolSystemStats()
See Also

DatabaseEnvironment Class
PrintMPoolSystemStats Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Display the memory pool (that is, buffer cache) subsystem statistical information, as described by MPoolStats.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
**Syntax**

**C#**

```csharp
public void PrintMPoolSystemStats(
    bool PrintAll,
    bool ClearStats
)
```

**Visual Basic (Declaration)**

```vbnet
Public Sub PrintMPoolSystemStats ( _
    PrintAll As Boolean, _
    ClearStats As Boolean _
)
```

**Visual C++**

```cpp
public:
void PrintMPoolSystemStats(
    bool PrintAll,
    bool ClearStats
)
```

**Parameters**

**PrintAll**
Type: `System::::Boolean`
If true, display all available information.

**ClearStats**
Type: `System::::Boolean`
If true, reset statistics after displaying their values.
See Also

DatabaseEnvironment Class
PrintMPoolSystemStats Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
DatabaseEnvironment...::PrintMPoolSystemStats Method (Boolean, Boolean, Boolean)

See Also

Display the memory pool (that is, buffer cache) subsystem statistical information, as described by MPoolStats.

Namespace: BerkeleyDB
### Syntax

#### C#

```csharp
public void PrintMPoolSystemStats(
    bool PrintAll,
    bool ClearStats,
    bool HashChains
)
```

#### Visual Basic (Declaration)

```vbnet
Public Sub PrintMPoolSystemStats (_
    PrintAll As Boolean, _
    ClearStats As Boolean, _
    HashChains As Boolean _
)
```

#### Visual C++

```cpp
public:
void PrintMPoolSystemStats(
    bool PrintAll,
    bool ClearStats,
    bool HashChains
)
```

### Parameters

**PrintAll**
- Type: `System:::Boolean`
- If true, display all available information.

**ClearStats**
- Type: `System:::Boolean`
- If true, reset statistics after displaying their values.

**HashChains**
- Type: `System:::Boolean`
If true, display the buffers with hash chains.
See Also

DatabaseEnvironment Class
PrintMPoolSystemStats Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation

DatabaseEnvironment...::PrintMutexSystemStats Method

DatabaseEnvironment Class  See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PrintMutexSystemStats()()()</td>
<td>Display the mutex subsystem statistical information, as described by MutexStats.</td>
</tr>
<tr>
<td>PrintMutexSystemStats(Boolean, Boolean)</td>
<td>Display the mutex subsystem statistical information, as described by MutexStats.</td>
</tr>
</tbody>
</table>
See Also

DatabaseEnvironment Class
DatabaseEnvironment Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Display the mutex subsystem statistical information, as described by [MutexStats](#).

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll)  
**Version:** 4.8.24.0
Syntax

C#

public void PrintMutexSystemStats()

Visual Basic (Declaration)

Public Sub PrintMutexSystemStats

Visual C++

public:
void PrintMutexSystemStats()
See Also

DatabaseEnvironment Class
PrintMutexSystemStats Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
DatabaseEnvironment...::.PrintMutexSystemStats Method (Boolean, Boolean)

DatabaseEnvironment Class  See Also

Display the mutex subsystem statistical information, as described by MutexStats.

Namespace:  BerkeleyDB
## Syntax

### C#

```csharp
public void PrintMutexSystemStats(
    bool PrintAll,
    bool ClearStats
)
```

### Visual Basic (Declaration)

```vbnet
Public Sub PrintMutexSystemStats (_
    PrintAll As Boolean, _
    ClearStats As Boolean _
)
```

### Visual C++

```cpp
public:
void PrintMutexSystemStats(
    bool PrintAll,
    bool ClearStats
)
```

## Parameters

**PrintAll**
- Type: `System::::Boolean`
- If true, display all available information.

**ClearStats**
- Type: `System::::Boolean`
- If true, reset statistics after displaying their values.
See Also

DatabaseEnvironment Class
PrintMutexSystemStats Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>PrintReplicationSystemStats()</code></td>
<td>Display the replication subsystem statistical information, as described by <code>ReplicationStats</code>.</td>
</tr>
<tr>
<td><code>PrintReplicationSystemStats(Boolean, Boolean)</code></td>
<td>Display the replication subsystem statistical information, as described by <code>ReplicationStats</code>.</td>
</tr>
</tbody>
</table>
See Also

DatabaseEnvironment Class
DatabaseEnvironment Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Display the replication subsystem statistical information, as described by `ReplicationStats`.

**Namespace:** [BerkeleyDB](https://docs.oracle.com/cd/E18677_01/net.800/82017/index.html)

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public void PrintReplicationSystemStats()

Visual Basic (Declaration)

Public Sub PrintReplicationSystemStats

Visual C++

public:
void PrintReplicationSystemStats()
See Also

DatabaseEnvironment Class
PrintReplicationSystemStats Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
DatabaseEnvironment...::PrintReplicationSystemStats Method (Boolean, Boolean)

See Also

Display the replication subsystem statistical information, as described by ReplicationStats.

Namespace: BerkeleyDB
Syntax

C#

```csharp
public void PrintReplicationSystemStats(
    bool PrintAll,
    bool ClearStats
)
```

Visual Basic (Declaration)

```vbnet
Public Sub PrintReplicationSystemStats ( _
    PrintAll As Boolean, _
    ClearStats As Boolean _
)
```

Visual C++

```cpp
public:
void PrintReplicationSystemStats(
    bool PrintAll,
    bool ClearStats
)
```

Parameters

PrintAll
Type: `System::::Boolean`
If true, display all available information.

ClearStats
Type: `System::::Boolean`
If true, reset statistics after displaying their values.
See Also

DatabaseEnvironment Class
PrintReplicationSystemStats Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation

DatabaseEnvironment:::PrintRepMgrSystemStats Method

DatabaseEnvironment Class  See Also
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PrintRepMgrSystemStats()()</td>
<td>Display the replication manager statistical information, as described by RepMgrStats.</td>
</tr>
<tr>
<td>PrintRepMgrSystemStats(Boolean, Boolean)</td>
<td>Display the replication manager statistical information, as described by RepMgrStats.</td>
</tr>
</tbody>
</table>
See Also

DatabaseEnvironment Class
DatabaseEnvironment Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Display the replication manager statistical information, as described by `RepMgrStats`.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
### Syntax

**C#**

```csharp
public void PrintRepMgrSystemStats()
```

**Visual Basic (Declaration)**

```vbnet
Public Sub PrintRepMgrSystemStats
```

**Visual C++**

```cpp
public:
void PrintRepMgrSystemStats()
```
See Also

DatabaseEnvironment Class
PrintRepMgrSystemStats Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
DatabaseEnvironment...::.PrintRepMgrSystemStats Method (Boolean, Boolean)

DatabaseEnvironment Class  See Also

Display the replication manager statistical information, as described by RepMgrStats.

Namespace:  BerkeleyDB
## Syntax

### C#

```csharp
public void PrintRepMgrSystemStats(  
    bool PrintAll,  
    bool ClearStats
)
```

### Visual Basic (Declaration)

```vbnet
Public Sub PrintRepMgrSystemStats (_
    PrintAll As Boolean, _
    ClearStats As Boolean _
)
```

### Visual C++

```cpp
public:
void PrintRepMgrSystemStats(  
    bool PrintAll,  
    bool ClearStats
)
```

## Parameters

**PrintAll**
Type: `System::::Boolean`
If true, display all available information.

**ClearStats**
Type: `System::::Boolean`
If true, reset statistics after displaying their values.
See Also

DatabaseEnvironment Class
PrintRepMgrSystemStats Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
DatabaseEnvironment Class

See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PrintStats()</td>
<td>Display the locking subsystem statistical information, as described by <a href="#">LockStats</a>.</td>
</tr>
<tr>
<td>PrintStats(Boolean, Boolean)</td>
<td>Display the locking subsystem statistical information, as described by <a href="#">LockStats</a>.</td>
</tr>
</tbody>
</table>
See Also

DatabaseEnvironment Class
DatabaseEnvironment Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Display the locking subsystem statistical information, as described by `LockStats`.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public void PrintStats()

Visual Basic (Declaration)

Public Sub PrintStats

Visual C++

public:
void PrintStats()
See Also

DatabaseEnvironment Class
PrintStats Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Display the locking subsystem statistical information, as described by LockStats.

Namespace: BerkeleyDB
Syntax

C#

    public void PrintStats(
        bool PrintAll,
        bool ClearStats
    )

Visual Basic (Declaration)

    Public Sub PrintStats(_
        PrintAll As Boolean, _
        ClearStats As Boolean _
    )

Visual C++

    public:
    void PrintStats(
        bool PrintAll,
        bool ClearStats
    )

Parameters

PrintAll
    Type: System::Boolean
    If true, display all available information.

ClearStats
    Type: System::Boolean
    If true, reset statistics after displaying their values.
See Also

DatabaseEnvironment Class
PrintStats Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
DatabaseEnvironment...PrintSubsystemStats Method

DatabaseEnvironment Class  See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>PrintSubsystemStats()</code></td>
<td>Display the locking subsystem statistical information, as described by <a href="#">LockStats</a>.</td>
</tr>
<tr>
<td><code>PrintSubsystemStats(Boolean, Boolean)</code></td>
<td>Display the locking subsystem statistical information, as described by <a href="#">LockStats</a>.</td>
</tr>
</tbody>
</table>
See Also

DatabaseEnvironment Class
DatabaseEnvironment Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Display the locking subsystem statistical information, as described by `LockStats`.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public void PrintSubsystemStats()

Visual Basic (Declaration)

Public Sub PrintSubsystemStats

Visual C++

public:
void PrintSubsystemStats()
See Also

DatabaseEnvironment Class
PrintSubsystemStats Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Display the locking subsystem statistical information, as described by LockStats.

**Namespace:**  BerkeleyDB  
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll)  
**Version:**  4.8.24.0
### Syntax

**C#**

```csharp
public void PrintSubsystemStats(
    bool PrintAll,
    bool ClearStats
)
```

**Visual Basic (Declaration)**

```vbnet
Public Sub PrintSubsystemStats (_
    PrintAll As Boolean, _
    ClearStats As Boolean _
)
```

**Visual C++**

```cpp
public:
void PrintSubsystemStats(
    bool PrintAll,
    bool ClearStats
)
```

### Parameters

**PrintAll**

- Type: `System::Boolean`
- If true, display all available information.

**ClearStats**

- Type: `System::Boolean`
- If true, reset statistics after displaying their values.
See Also

DatabaseEnvironment Class
PrintSubsystemStats Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation

DatabaseEnvironment...:::PrintTransactionSystemStats Method

DatabaseEnvironment Class  See Also
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="link" alt="PrintTransactionSystemStats()" /></td>
<td>Display the transaction subsystem statistical information, as described by <a href="#">TransactionStats</a>.</td>
</tr>
<tr>
<td><img src="link" alt="PrintTransactionSystemStats(Boolean, Boolean)" /></td>
<td>Display the transaction subsystem statistical information, as described by <a href="#">TransactionStats</a>.</td>
</tr>
</tbody>
</table>
See Also

DatabaseEnvironment Class
DatabaseEnvironment Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Display the transaction subsystem statistical information, as described by `TransactionStats`.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public void PrintTransactionSystemStats()

Visual Basic (Declaration)

Public Sub PrintTransactionSystemStats

Visual C++

public:
void PrintTransactionSystemStats()
See Also

- DatabaseEnvironment Class
- PrintTransactionSystemStats Overload
- BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
DatabaseEnvironment..::.PrintTransactionSystemStats Method (Boolean, Boolean)

See Also

Display the transaction subsystem statistical information, as described by TransactionStats.

Namespace: BerkeleyDB
Syntax

C#

```csharp
public void PrintTransactionSystemStats(
    bool PrintAll,
    bool ClearStats
)
```

Visual Basic (Declaration)

```vbnet
Public Sub PrintTransactionSystemStats ( _
    PrintAll As Boolean, _
    ClearStats As Boolean _
)
```

Visual C++

```cpp
public:
void PrintTransactionSystemStats(
    bool PrintAll,
    bool ClearStats
)
```

Parameters

PrintAll
  Type: System::Boolean
  If true, display all available information.

ClearStats
  Type: System::Boolean
  If true, reset statistics after displaying their values.
See Also

DatabaseEnvironment Class
PrintTransactionSystemStats Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
DatabaseEnvironment::Recover Method

Restore transactions that were prepared, but not yet resolved at the time of the system shut down or crash, to their state prior to the shut down or crash, including any locks previously held.

Namespace: BerkeleyDB
Syntax

C#

```csharp
public PreparedTransaction[] Recover(
    uint count,
    bool resume
)
```

Visual Basic (Declaration)

```vb
Public Function Recover (_
    count As UInteger, _
    resume As Boolean _
) As PreparedTransaction()
```

Visual C++

```cpp
public:
array<PreparedTransaction^>^ Recover(
    unsigned int count,
    bool resume
)
```

Parameters

count
Type: System::::UInt32
The maximum number of PreparedTransaction objects to return.

resume
Type: System::::Boolean
If true, continue returning a list of prepared, but not yet resolved transactions, starting where the last call to Recover left off. If false, begins a new pass over all prepared, but not yet completed transactions, regardless of whether they have already been returned in previous calls to Recover.

Return Value
A list of the prepared transactions
Remarks

Calls to Recover from different threads of control should not be intermixed in the same environment.
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
DatabaseEnvironment:::Remove Method

DatabaseEnvironment Class  See Also

The environment regions, including any backing files, are removed. Any log or database files and the environment directory are not removed.

If there are processes that have called Open(String, DatabaseEnvironmentConfig) without calling Close() (that is, there are processes currently using the environment), Remove will fail without further action.

Calling Remove should not be necessary for most applications because the Berkeley DB environment is cleaned up as part of normal database recovery procedures. However, applications may want to call Remove as part of application shut down to free up system resources. For example, if SystemMemory was specified to Open(String, DatabaseEnvironmentConfig), it may be useful to call Remove in order to release system shared memory segments that have been allocated. Or, on architectures in which mutexes require allocation of underlying system resources, it may be useful to call Remove in order to release those resources. Alternatively, if recovery is not required because no database state is maintained across failures, and no system resources need to be released, it is possible to clean up an environment by simply removing all the Berkeley DB files in the database environment's directories.

In multithreaded applications, only a single thread may call Remove.
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Remove(String)</td>
<td>Destroy a Berkeley DB environment if it is not currently in use.</td>
</tr>
<tr>
<td>Remove(String, Boolean)</td>
<td>Destroy a Berkeley DB environment if it is not currently in use.</td>
</tr>
</tbody>
</table>
See Also

[DatabaseEnvironment Class](#)
[DatabaseEnvironment Members](#)
[BerkeleyDB Namespace](#)

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Destroy a Berkeley DB environment if it is not currently in use.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
### Syntax

**C#**

```csharp
public static void Remove(
    string db_home
)
```

**Visual Basic (Declaration)**

```vbnet
Public Shared Sub Remove (_
    db_home As String _
)
```

**Visual C++**

```cpp
public:
static void Remove(
    String^ db_home
)
```

### Parameters

**db_home**

- Type: `System::String`
- The database environment to be removed.
See Also

DatabaseEnvironment Class
Remove Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation

DatabaseEnvironment..:::Remove Method (String, Boolean)

DatabaseEnvironment Class  See Also

Destroy a Berkeley DB environment if it is not currently in use.

**Namespace:**  BerkeleyDB

**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public static void Remove(
    string db_home,
    bool force
)

Visual Basic (Declaration)

Public Shared Sub Remove ( _
    db_home As String, _
    force As Boolean _
)

Visual C++

public:
static void Remove(
    String^ db_home,
    bool force
)

Parameters

db_home
Type: System::String
The database environment to be removed.

force
Type: System::Boolean
If true, the environment is removed, regardless of any processes that may still using it, and no locks are acquired during this process.
Remarks

Generally, force is specified only when applications were unable to shut down cleanly, and there is a risk that an application may have died holding a Berkeley DB lock.)

The result of attempting to forcibly destroy the environment when it is in use is unspecified. Processes using an environment often maintain open file descriptors for shared regions within it. On UNIX systems, the environment removal will usually succeed, and processes that have already joined the region will continue to run in that region without change. However, processes attempting to join the environment will either fail or create new regions. On other systems in which the unlink(2) system call will fail if any process has an open file descriptor for the file (for example Windows/NT), the region removal will fail.
See Also

DatabaseEnvironment Class
Remove Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation

DatabaseEnvironment...::RemoveDB Method

DatabaseEnvironment Class  See Also
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>RemoveDB(String, Boolean)</code></td>
<td>Remove the underlying file represented by file, incidentally removing all of the databases it contained.</td>
</tr>
<tr>
<td><code>RemoveDB(String, Boolean, Transaction)</code></td>
<td>Remove the underlying file represented by file, incidentally removing all of the databases it contained.</td>
</tr>
<tr>
<td><code>RemoveDB(String, String, Boolean)</code></td>
<td>Remove the database specified by file and database. If no database is specified, the underlying file represented by file is removed, incidentally removing all of the databases it contained.</td>
</tr>
<tr>
<td><code>RemoveDB(String, String, Boolean, Transaction)</code></td>
<td>Remove the database specified by file and database. If no database is specified, the underlying file represented by file is removed, incidentally removing all of the databases it contained.</td>
</tr>
</tbody>
</table>
See Also

DatabaseEnvironment Class
DatabaseEnvironment Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Remove the underlying file represented by file, incidentally removing all of the databases it contained.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

**C#**

```csharp
public void RemoveDB(
    string file,
    bool autoCommit
)
```

**Visual Basic (Declaration)**

```vbnet
Public Sub RemoveDB (_
    file As String, _
    autoCommit As Boolean _
)
```

**Visual C++**

```cpp
public:
void RemoveDB(
    String^ file,
    bool autoCommit
)
```

**Parameters**

**file**

Type: `System:::String`
The physical file to be removed.

**autoCommit**

Type: `System:::Boolean`
If true, enclose RemoveDB within a transaction. If the call succeeds, changes made by the operation will be recoverable. If the call fails, the operation will have made no changes.
See Also

DatabaseEnvironment Class
RemoveDB Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
RemoveDB Method (String, Boolean, Transaction)

Remove the underlying file represented by file, incidentally removing all of the databases it contained.

Namespace: BerkeleyDB
**Syntax**

**C#**

```csharp
public void RemoveDB(
    string file,
    bool autoCommit,
    Transaction txn
)
```

**Visual Basic (Declaration)**

```vbnet
Public Sub RemoveDB (_
    file As String, _
    autoCommit As Boolean, _
    txn As Transaction _
)
```

**Visual C++**

```cpp
public:
void RemoveDB(
    String^ file,
    bool autoCommit,
    Transaction^ txn
)
```

**Parameters**

**file**

Type: `System::::String`

The physical file to be removed.

**autoCommit**

Type: `System::::Boolean`

If true, enclose RemoveDB within a transaction. If the call succeeds, changes made by the operation will be recoverable. If the call fails, the operation will have made no changes.
txn

Type: BerkeleyDB::::Transaction
If the operation is part of an application-specified transaction, txn is a Transaction object returned from BeginTransaction(); if the operation is part of a Berkeley DB Concurrent Data Store group, txn is a handle returned from BeginCDSGroup(); otherwise null. If null, but autoCommit or AutoCommit is true, the operation will be implicitly transaction protected.
See Also

DatabaseEnvironment Class
RemoveDB Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Remove the database specified by file and database. If no database is specified, the underlying file represented by file is removed, incidentally removing all of the databases it contained.

**Namespace:**  BerkeleyDB  
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll)  
**Version:**  4.8.24.0
## Syntax

### C#

```csharp
public void RemoveDB(
    string file,
    string database,
    bool autoCommit
)
```

### Visual Basic (Declaration)

```vbnet
Public Sub RemoveDB (_
    file As String, _
    database As String, _
    autoCommit As Boolean _
)
```

### Visual C++

```cpp
public:
void RemoveDB(
    String^ file,
    String^ database,
    bool autoCommit
)
```

## Parameters

**file**
- Type: `System::String`
- The physical file which contains the database(s) to be removed.

**database**
- Type: `System::String`
- The database to be removed.

**autoCommit**
- Type: `System::Boolean`
If true, enclose RemoveDB within a transaction. If the call succeeds, changes made by the operation will be recoverable. If the call fails, the operation will have made no changes.
See Also

DatabaseEnvironment Class
RemoveDB Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Remove the database specified by file and database. If no database is specified, the underlying file represented by file is removed, incidentally removing all of the databases it contained.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public void RemoveDB(
    string file,
    string database,
    bool autoCommit,
    Transaction txn
)

Visual Basic (Declaration)

Public Sub RemoveDB (_
    file As String, _
    database As String, _
    autoCommit As Boolean, _
    txn As Transaction _
)

Visual C++

public:
    void RemoveDB(
        String^ file,
        String^ database,
        bool autoCommit,
        Transaction^ txn
    )

Parameters

file
    Type: System::String
    The physical file which contains the database(s) to be removed.

database
    Type: System::String
    The database to be removed.
autoCommit
Type: System::Boolean
If true, enclose RemoveDB within a transaction. If the call succeeds, changes made by the operation will be recoverable. If the call fails, the operation will have made no changes.

txn
Type: BerkeleyDB::Transaction
If the operation is part of an application-specified transaction, txn is a Transaction object returned from BeginTransaction(); if the operation is part of a Berkeley DB Concurrent Data Store group, txn is a handle returned from BeginCDSGroup(); otherwise null. If null, but autoCommit or AutoCommit is true, the operation will be implicitly transaction protected.
See Also

DatabaseEnvironment Class
RemoveDB Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Remove log files that are no longer needed. Automatic log file removal is likely to make catastrophic recovery impossible.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public void RemoveUnusedLogFiles()

Visual Basic (Declaration)

Public Sub RemoveUnusedLogFiles

Visual C++

public:
void RemoveUnusedLogFiles()
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
DatabaseEnvironment Class  See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RenameDB(String, String, Boolean)</td>
<td>Rename the underlying file represented by file using the value supplied to newname, incidentally renaming all of the databases it contained.</td>
</tr>
<tr>
<td>RenameDB(String, String, Boolean, Transaction)</td>
<td>Rename the underlying file represented by file using the value supplied to newname, incidentally renaming all of the databases it contained.</td>
</tr>
<tr>
<td>RenameDB(String, String, String, Boolean)</td>
<td>Rename the database specified by file and database to newname. If no database is specified, the underlying file represented by file is renamed using the value supplied to newname, incidentally renaming all of the databases it contained.</td>
</tr>
<tr>
<td>RenameDB(String, String, String, Boolean, Transaction)</td>
<td>Rename the database specified by file and database to newname. If no database is specified, the underlying file represented by file is renamed using the value supplied to newname, incidentally renaming all of the databases it contained.</td>
</tr>
</tbody>
</table>
See Also

DatabaseEnvironment Class
DatabaseEnvironment Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
RenameDB Method (String, String, Boolean)

**DatabaseEnvironment Class**  **See Also**

Rename the underlying file represented by file using the value supplied to newname, incidentally renaming all of the databases it contained.

**Namespace:**  **BerkeleyDB**

**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
### Syntax

**C#**

```csharp
public void RenameDB(
    string file,
    string newname,
    bool autoCommit
)
```

**Visual Basic (Declaration)**

```vbnet
Public Sub RenameDB (_
    file As String, _
    newname As String, _
    autoCommit As Boolean _
)
```

**Visual C++**

```cpp
public:
void RenameDB(
    String^ file,
    String^ newname,
    bool autoCommit
)
```

### Parameters

**file**

- **Type:** `System::String`
- The physical file to be renamed.

**newname**

- **Type:** `System::String`
- The new name of the database or file.

**autoCommit**

- **Type:** `System::Boolean`
If true, enclose RenameDB within a transaction. If the call succeeds, changes made by the operation will be recoverable. If the call fails, the operation will have made no changes.
See Also

DatabaseEnvironment Class
RenameDB Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
RenameDB Method (String, String, Boolean, Transaction)

DatabaseEnvironment Class  See Also

Rename the underlying file represented by file using the value supplied to newname, incidentally renaming all of the databases it contained.

Namespace:  BerkeleyDB
Syntax

C#

public void RenameDB(
    string file,
    string newName,
    bool autoCommit,
    Transaction txn
)

Visual Basic (Declaration)

Public Sub RenameDB ( _
    file As String, _
    newName As String, _
    autoCommit As Boolean, _
    txn As Transaction _
)

Visual C++

public:
    void RenameDB(
        String^ file,
        String^ newName,
        bool autoCommit,
        Transaction^ txn
    )

Parameters

file
    Type: System::String
    The physical file to be renamed.

eickname
    Type: System::String
    The new name of the database or file.
autoCommit
Type: System::::Boolean
If true, enclose RenameDB within a transaction. If the call succeeds, changes made by the operation will be recoverable. If the call fails, the operation will have made no changes.

txn
Type: BerkeleyDB::::Transaction
If the operation is part of an application-specified transaction, txn is a Transaction object returned from BeginTransaction(); if the operation is part of a Berkeley DB Concurrent Data Store group, txn is a handle returned from BeginCDSGroup(); otherwise null. If null, but autoCommit or AutoCommit is true, the operation will be implicitly transaction protected.
See Also

DatabaseEnvironment Class
RenameDB Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
C#  Visual Basic  
Visual C++

Berkeley DB .NET API Documentation

DatabaseEnvironment..::.RenameDB Method (String, String, String, Boolean)

DatabaseEnvironment Class  See Also

Rename the database specified by file and database to newname. If no database is specified, the underlying file represented by file is renamed using the value supplied to newname, incidentally renaming all of the databases it contained.

Namespace:  BerkeleyDB
## Syntax

### C#

```csharp
public void RenameDB(
    string file,
    string database,
    string newname,
    bool autoCommit
)
```

### Visual Basic (Declaration)

```vbnet
Public Sub RenameDB (_
    file As String, _
    database As String, _
    newname As String, _
    autoCommit As Boolean _
)
```

### Visual C++

```cpp
public:
void RenameDB(
    String^ file,
    String^ database,
    String^ newname,
    bool autoCommit
)
```

## Parameters

**file**

Type: `System::String`

The physical file which contains the database(s) to be renamed.

**database**

Type: `System::String`

The database to be renamed.
newname
Type: System::String
The new name of the database or file.

autoCommit
Type: System::Boolean
If true, enclose RenameDB within a transaction. If the call succeeds, changes made by the operation will be recoverable. If the call fails, the operation will have made no changes.
See Also

DatabaseEnvironment Class
RenameDB Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation
DatabaseEnvironment....RenameDB Method (String, String, String, Boolean, Transaction)

**DatabaseEnvironment Class**  [See Also](#)

Rename the database specified by file and database to newname. If no database is specified, the underlying file represented by file is renamed using the value supplied to newname, incidentally renaming all of the databases it contained.

**Namespace:** [BerkeleyDB](#)
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

```csharp
public void RenameDB(
    string file,
    string database,
    string newName,
    bool autoCommit,
    Transaction txn
)
```

Visual Basic (Declaration)

```vbnet
Public Sub RenameDB (_
    file As String, _
    database As String, _
    newName As String, _
    autoCommit As Boolean, _
    txn As Transaction _
)
```

Visual C++

```cpp
public:
void RenameDB(
    String^ file,
    String^ database,
    String^ newName,
    bool autoCommit,
    Transaction^ txn
)
```

Parameters

file

Type: System::::String

The physical file which contains the database(s) to be renamed.

database
Type: `System::: String`
The database to be renamed.

`newname`
Type: `System::: String`
The new name of the database or file.

`autoCommit`
Type: `System::: Boolean`
If true, enclose RenameDB within a transaction. If the call succeeds, changes made by the operation will be recoverable. If the call fails, the operation will have made no changes.

`txn`
Type: `BerkeleyDB::: Transaction`
If the operation is part of an application-specified transaction, `txn` is a Transaction object returned from `BeginTransaction()`; if the operation is part of a Berkeley DB Concurrent Data Store group, `txn` is a handle returned from `BeginCDSGroup()`; otherwise null. If null, but `autoCommit` or `AutoCommit` is true, the operation will be implicitly transaction protected.
See Also

DatabaseEnvironment Class
RenameDB Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
RepHoldElection is not called by most replication applications. It should only be called by applications implementing their own network transport layer, explicitly holding replication group elections and handling replication messages outside of the replication manager framework.

If the election is successful, Berkeley DB will notify the application of the results of the election by means of either the REP_ELECTED or REP_NEWMASTER events (see EventNotify for more information). The application is responsible for adjusting its relationship to the other database environments in the replication group, including directing all database updates to the newly selected master, in accordance with the results of the election.

The thread of control that calls RepHoldElection must not be the thread of control that processes incoming messages; processing the incoming messages is necessary to successfully complete an election.

Before calling this method, the RepTransport delegate must already have been configured to send replication messages.
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RepHoldElection()()</td>
<td>Hold an election for the master of a replication group.</td>
</tr>
<tr>
<td>RepHoldElection(UInt32)</td>
<td>Hold an election for the master of a replication group.</td>
</tr>
<tr>
<td>RepHoldElection(UInt32, UInt32)</td>
<td>Hold an election for the master of a replication group.</td>
</tr>
</tbody>
</table>
See Also

DatabaseEnvironment Class
DatabaseEnvironment Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Hold an election for the master of a replication group.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public void RepHoldElection()

Visual Basic (Declaration)

Public Sub RepHoldElection

Visual C++

public:
void RepHoldElection()
See Also

- DatabaseEnvironment Class
- RepHoldElection Overload
- BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Hold an election for the master of a replication group.

**Namespace**: BerkeleyDB  
**Assembly**: libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public void RepHoldElection(
  uint nsites
)

Visual Basic (Declaration)

Public Sub RepHoldElection ( _
  nsites As UInteger _
)

Visual C++

public:
void RepHoldElection( _
  unsigned int nsites
)

Parameters

nsites
  Type: System::::UInt32
  The number of replication sites expected to participate in the election. Once the current site has election information from that many sites, it will short-circuit the election and immediately cast its vote for a new master. This parameter must be no less than nvotes, or 0 if the election should use RepNSites. If an application is using master leases, then the value must be 0 and RepNSites must be used.
See Also

DatabaseEnvironment Class
RepHoldElection Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Hold an election for the master of a replication group.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
## Syntax

### C#

```csharp
public void RepHoldElection(
    uint nsites,
    uint nvotes
)
```

### Visual Basic (Declaration)

```vbnet
Public Sub RepHoldElection (_
    nsites As UInteger, _
    nvotes As UInteger _
)
```

### Visual C++

```cpp
public:
void RepHoldElection(
    unsigned int nsites,
    unsigned int nvotes
)
```

## Parameters

### nsites
Type: `System::::UInt32`

The number of replication sites expected to participate in the election. Once the current site has election information from that many sites, it will short-circuit the election and immediately cast its vote for a new master. This parameter must be no less than `nvotes`, or 0 if the election should use `RepNSites`. If an application is using master leases, then the value must be 0 and `RepNSites` must be used.

### nvotes
Type: `System::::UInt32`

The minimum number of replication sites from which the current site must
have election information, before the current site will cast a vote for a new master. This parameter must be no greater than nsites, or 0 if the election should use the value \(((\text{nsites} / 2) + 1)\).
See Also

DatabaseEnvironment Class
RepHoldElection Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation

DatabaseEnvironment...::ReplicationSystemStats Method

DatabaseEnvironment Class  See Also
# Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ReplicationSystemStats()()</td>
<td>The replication subsystem statistics</td>
</tr>
<tr>
<td>ReplicationSystemStats(Boolean)</td>
<td>The replication subsystem statistics</td>
</tr>
</tbody>
</table>
See Also

DatabaseEnvironment Class
DatabaseEnvironment Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The replication subsystem statistics

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public ReplicationStats ReplicationSystemStats()

Visual Basic (Declaration)

Public Function ReplicationSystemStats As ReplicationStats

Visual C++

public:
ReplicationStats^ ReplicationSystemStats()

Return Value

The replication subsystem statistics
See Also

DatabaseEnvironment Class
ReplicationSystemStats Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The replication subsystem statistics

**Namespace:** BerkeleyDB
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public ReplicationStats ReplicationSystemStats(
    bool clearStats
)

Visual Basic (Declaration)

Public Function ReplicationSystemStats (_
    clearStats As Boolean _
) As ReplicationStats

Visual C++

public:
    ReplicationStats^ ReplicationSystemStats(
        bool clearStats
    )

Parameters

clearStats
    Type: System::Boolean
    If true, reset statistics after returning their values.

Return Value

The replication subsystem statistics
See Also

DatabaseEnvironment Class
ReplicationSystemStats Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation
DatabaseEnvironment...:
RepMgrAddRemoteSite Method
DatabaseEnvironment Class  See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RepMgrAddRemoteSite(ReplicationHostAddress)</td>
<td>Add a new replication site to the replication manager's list of known sites. It is not necessary for all sites in a replication group to know about all other sites in the group.</td>
</tr>
<tr>
<td>RepMgrAddRemoteSite(ReplicationHostAddress, Boolean)</td>
<td>Add a new replication site to the replication manager's list of known sites. It is not necessary for all sites in a replication group to know about all other sites in the group.</td>
</tr>
</tbody>
</table>
See Also

DatabaseEnvironment Class
DatabaseEnvironment Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Add a new replication site to the replication manager's list of known sites. It is not necessary for all sites in a replication group to know about all other sites in the group.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

```csharp
public int RepMgrAddRemoteSite(ReplicationHostAddress Host)
```

Visual Basic (Declaration)

```vbnet
Public Function RepMgrAddRemoteSite( _
    Host As ReplicationHostAddress _
) As Integer
```

Visual C++

```cpp
public:
    int RepMgrAddRemoteSite(ReplicationHostAddress^ Host)
```

Parameters

Host

Type: BerkeleyDB..:::ReplicationHostAddress

The remote site's address

Return Value

The environment ID assigned to the remote site
See Also

DatabaseEnvironment Class
RepMgrAddRemoteSite Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Add a new replication site to the replication manager's list of known sites. It is not necessary for all sites in a replication group to know about all other sites in the group.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

```csharp
public int RepMgrAddRemoteSite(
    ReplicationHostAddress Host,
    bool isPeer
)
```

Visual Basic (Declaration)

```vbnet
Public Function RepMgrAddRemoteSite ( _
    Host As ReplicationHostAddress, _
    isPeer As Boolean _
) As Integer
```

Visual C++

```cpp
public:
    int RepMgrAddRemoteSite(
        ReplicationHostAddress^ Host,
        bool isPeer
    )
```

Parameters

Host
Type: `BerkeleyDB::ReplicationHostAddress`
The remote site's address

isPeer
Type: `System::Boolean`
If true, configure client-to-client synchronization with the specified remote site.

Return Value

The environment ID assigned to the remote site
Remarks

Currently, the replication manager framework only supports a single client peer, and the last specified peer is used.
See Also

DatabaseEnvironment Class
RepMgrAddRemoteSite Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
There are two ways to build Berkeley DB replication applications: the most common approach is to use the Berkeley DB library "replication manager" support, where the Berkeley DB library manages the replication group, including network transport, all replication message processing and acknowledgment, and group elections. Applications using the replication manager support generally make the following calls:

1. Configure the local site in the replication group, `RepMgrLocalSite`.
2. Call `RepMgrAddRemoteSite(ReplicationHostAddress)` to configure the remote site(s) in the replication group.
3. Configure the message acknowledgment policy (`RepMgrAckPolicy`) which provides the replication group's transactional needs.
4. Configure the local site's election priority, `RepPriority`.
5. Call `RepMgrStartClient(Int32)` or `RepMgrStartMaster(Int32)` to start the replication application.

For more information on building replication manager applications, please see the Replication Getting Started Guide included in the Berkeley DB documentation.

Applications with special needs (for example, applications using network protocols not supported by the Berkeley DB replication manager), must perform additional configuration and call other Berkeley DB replication methods. For more information on building advanced replication applications, please see the Base Replication API section in the Berkeley DB Programmer's Reference Guide for more information.

Starting the replication manager consists of opening the TCP/IP listening socket to accept incoming connections, and starting all necessary background threads.
When multiple processes share a database environment, only one process can open the listening socket; `RepMgrStartClient(Int32)` (and `RepMgrStartMaster(Int32)`) automatically open the socket in the first process to call it, and skips this step in the later calls from other processes.
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RepMgrStartClient(Int32)</td>
<td>Start the replication manager as a client site, and do not call for an election.</td>
</tr>
<tr>
<td>RepMgrStartClient(Int32, Boolean)</td>
<td>Start the replication manager as a client site, and optionally call for an election.</td>
</tr>
</tbody>
</table>
See Also

DatabaseEnvironment Class
DatabaseEnvironment Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Start the replication manager as a client site, and do not call for an election.

**Namespace:** [BerkeleyDB](#)

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public void RepMgrStartClient(
    int nthreads
)

Visual Basic (Declaration)

Public Sub RepMgrStartClient ( _
    nthreads As Integer _
)

Visual C++

public:
void RepMgrStartClient(
    int nthreads
)

Parameters

nthreads
  Type: System::::Int32
  Specify the number of threads of control created and dedicated to processing replication messages. In addition to these message processing threads, the replication manager creates and manages a few of its own threads of control.
See Also

DatabaseEnvironment Class
RepMgrStartClient Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Start the replication manager as a client site, and optionally call for an election.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

```csharp
public void RepMgrStartClient(
    int nthreads,
    bool holdElection
)
```

Visual Basic (Declaration)

```vbnet
Public Sub RepMgrStartClient (
    nthreads As Integer,
    holdElection As Boolean
)
```

Visual C++

```cpp
public:
void RepMgrStartClient(
    int nthreads,
    bool holdElection
)
```

Parameters

nthreads
Type: `System::Int32`
Specify the number of threads of control created and dedicated to processing replication messages. In addition to these message processing threads, the replication manager creates and manages a few of its own threads of control.

holdElection
Type: `System::Boolean`
If true, start as a client, and call for an election if no master is found.
See Also

DatabaseEnvironment Class
RepMgrStartClient Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Start the replication manager as a master site, and do not call for an election.

**Namespace:**  BerkeleyDB  
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

```csharp
public void RepMgrStartMaster(
    int nthreads
)
```

Visual Basic (Declaration)

```vbnet
Public Sub RepMgrStartMaster ( _
    nthreads As Integer _
)
```

Visual C++

```cpp
public:
void RepMgrStartMaster(
    int nthreads
)
```

Parameters

nthreads
Type: `System::::Int32`
Specify the number of threads of control created and dedicated to processing replication messages. In addition to these message processing threads, the replication manager creates and manages a few of its own threads of control.
Remarks

There are two ways to build Berkeley DB replication applications: the most common approach is to use the Berkeley DB library "replication manager" support, where the Berkeley DB library manages the replication group, including network transport, all replication message processing and acknowledgment, and group elections. Applications using the replication manager support generally make the following calls:

1. Configure the local site in the replication group, `RepMgrLocalSite`.
2. Call `RepMgrAddRemoteSite(ReplicationHostAddress)` to configure the remote site(s) in the replication group.
3. Configure the message acknowledgment policy (`RepMgrAckPolicy`) which provides the replication group's transactional needs.
4. Configure the local site's election priority, `RepPriority`.
5. Call `RepMgrStartClient(Int32)` or `RepMgrStartMaster(Int32)` to start the replication application.

For more information on building replication manager applications, please see the Replication Getting Started Guide included in the Berkeley DB documentation.

Applications with special needs (for example, applications using network protocols not supported by the Berkeley DB replication manager), must perform additional configuration and call other Berkeley DB replication methods. For more information on building advanced replication applications, please see the Base Replication API section in the Berkeley DB Programmer's Reference Guide for more information.

Starting the replication manager consists of opening the TCP/IP listening socket to accept incoming connections, and starting all necessary background threads. When multiple processes share a database environment, only one process can open the listening socket; `RepMgrStartMaster(Int32)` (and `RepMgrStartClient(Int32)`) automatically open the socket in the first process to call it, and skips this step in the later calls from other processes.
See Also

DatabaseEnvironment Class  
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation
DatabaseEnvironment...:::RepMgrSystemStats Method
DatabaseEnvironment Class  See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>RepMgrSystemStats()</code></td>
<td>The replication manager statistics</td>
</tr>
<tr>
<td><code>RepMgrSystemStats(Boolean)</code></td>
<td>The replication manager statistics</td>
</tr>
</tbody>
</table>
See Also

DatabaseEnvironment Class
DatabaseEnvironment Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The replication manager statistics

Namespace: BerkeleyDB
Syntax

C#

public RepMgrStats RepMgrSystemStats()

Visual Basic (Declaration)

Public Function RepMgrSystemStats As RepMgrStats

Visual C++

public:
 RepMgrStats^ RepMgrSystemStats()

Return Value

The replication manager statistics
See Also

DatabaseEnvironment Class
RepMgrSystemStats Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation

DatabaseEnvironment...::RepMgrSystemStats Method (Boolean)

DatabaseEnvironment Class  See Also

The replication manager statistics

Namespace:  BerkeleyDB
Syntax

C#

```csharp
public RepMgrStats RepMgrSystemStats(
    bool clearStats
)
```

Visual Basic (Declaration)

```vbnet
Public Function RepMgrSystemStats (_
    clearStats As Boolean _
) As RepMgrStats
```

Visual C++

```cpp
public:
    RepMgrStats^ RepMgrSystemStats(
        bool clearStats
    )
```

Parameters

clearStats
- Type: System::Boolean
- If true, reset statistics after returning their values.

Return Value

The replication manager statistics
See Also

DatabaseEnvironment Class
RepMgrSystemStats Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Process an incoming replication message sent by a member of the replication group to the local database environment.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public RepProcMsgResult RepProcessMessage(
    DatabaseEntry control,
    DatabaseEntry rec,
    int envid
)

Visual Basic (Declaration)

Public Function RepProcessMessage (_
    control As DatabaseEntry, _
    rec As DatabaseEntry, _
    envid As Integer _
) As RepProcMsgResult

Visual C++

public:
RepProcMsgResult^ RepProcessMessage(
    DatabaseEntry^ control,
    DatabaseEntry^ rec,
    int envid
)

Parameters

ccontrol
    Type: BerkeleyDB:::DatabaseEntry
    A copy of the control parameter specified by Berkeley DB on the sending environment.

rec
    Type: BerkeleyDB:::DatabaseEntry
    A copy of the rec parameter specified by Berkeley DB on the sending environment.
envid
   Type: System::Int32
   The local identifier that corresponds to the environment that sent the message to be processed (see Replication environment IDs in the Programmer's Reference Guide for more information).

Return Value

The result of processing a message
Remarks

RepProcessMessage is not called by most replication applications. It should only be called by applications implementing their own network transport layer, explicitly holding replication group elections and handling replication messages outside of the replication manager framework.

For implementation reasons, all incoming replication messages must be processed using the same DatabaseEnvironment object. It is not required that a single thread of control process all messages, only that all threads of control processing messages use the same object.

Before calling this method, the RepTransport delegate must already have been configured to send replication messages.
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Set the clock skew ratio among replication group members based on the fastest and slowest measurements among the group for use with master leases.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
```csharp
public void RepSetClockskew(
    uint fast,
    uint slow
)
```

**Visual Basic (Declaration)**

```vbnet
Public Sub RepSetClockskew (_
    fast AsUInteger, _
    slow AsUInteger _
)
```

**Visual C++**

```c++
public:
void RepSetClockskew(
    unsigned int fast,
    unsigned int slow
)
```

**Parameters**

**fast**

Type: System::::UInt32
The value, relative to slow, of the fastest clock in the group of sites.

**slow**

Type: System::::UInt32
The value of the slowest clock in the group of sites.
**Remarks**

Calling this method is optional, the default values for clock skew assume no skew. The user must also configure leases via `RepUseMasterLeases`. Additionally, the user must also set the master lease timeout via `RepLeaseTimeout` and the number of sites in the replication group via `RepNSites`. These settings may be configured in any order. For a description of the clock skew values, see Clock skew in the Berkeley DB Programmer's Reference Guide. For a description of master leases, see Master leases in the Berkeley DB Programmer's Reference Guide.

These arguments can be used to express either raw measurements of a clock timing experiment or a percentage across machines. For instance a group of sites have a 2% variance, then fast should be set to 102, and slow should be set to 100. Or, for a 0.03% difference, you can use 10003 and 10000 respectively.
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
DatabaseEnvironment...:::RepSetRetransmissionRequest Method

Set a threshold for the minimum and maximum time that a client waits before requesting retransmission of a missing message.

Namespace: BerkeleyDB
Syntax

C#

public void RepSetRetransmissionRequest(
    uint min,
    uint max
)

Visual Basic (Declaration)

Public Sub RepSetRetransmissionRequest (_
    min As UInteger, _
    max As UInteger_
)

Visual C++

public:
void RepSetRetransmissionRequest(
    unsigned int min,
    unsigned int max
)

Parameters

min
Type: System::::UInt32
The minimum number of microseconds a client waits before requesting retransmission.

max
Type: System::::UInt32
The maximum number of microseconds a client waits before requesting retransmission.
Remarks

If the client detects a gap in the sequence of incoming log records or database pages, Berkeley DB will wait for at least min microseconds before requesting retransmission of the missing record. Berkeley DB will double that amount before requesting the same missing record again, and so on, up to a maximum threshold of max microseconds.

These values are thresholds only. Since Berkeley DB has no thread available in the library as a timer, the threshold is only checked when a thread enters the Berkeley DB library to process an incoming replication message. Any amount of time may have passed since the last message arrived and Berkeley DB only checks whether the amount of time since a request was made is beyond the threshold value or not.

By default the minimum is 40000 and the maximum is 1280000 (1.28 seconds). These defaults are fairly arbitrary and the application likely needs to adjust these. The values should be based on expected load and performance characteristics of the master and client host platforms and transport infrastructure as well as round-trip message time.
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Set a byte-count limit on the amount of data that will be transmitted from a site in response to a single message processed by `RepProcessMessage(DatabaseEntry, DatabaseEntry, Int32)`. The limit is not a hard limit, and the record that exceeds the limit is the last record to be sent.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

```csharp
public void RepSetTransmitLimit(
    uint GBytes,
    uint Bytes
)
```

Visual Basic (Declaration)

```vbnet
Public Sub RepSetTransmitLimit (
    _
    GBytes AsUInteger, _
    Bytes AsUInteger _
)
```

Visual C++

```cpp
public:
    void RepSetTransmitLimit(
        unsigned int GBytes,
        unsigned int Bytes
    )
```

Parameters

GBytes
Type: `System::::UInt32`
The number of gigabytes which, when added to Bytes, specifies the maximum number of bytes that will be sent in a single call to `RepProcessMessage(DatabaseEntry, DatabaseEntry, Int32)`.

Bytes
Type: `System::::UInt32`
The number of bytes which, when added to GBytes, specifies the maximum number of bytes that will be sent in a single call to `RepProcessMessage(DatabaseEntry, DatabaseEntry, Int32)`.
Remarks

Record transmission throttling is turned on by default with a limit of 10MB.

If both GBytes and Bytes are zero, then the transmission limit is turned off.
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
DatabaseEnvironment...:::RepSetTransport Method

DatabaseEnvironment Class  See Also

Initialize the communication infrastructure for a database environment participating in a replicated application.

Namespace:  BerkeleyDB
Syntax

C#

public void RepSetTransport(
    int envid,
    ReplicationTransportDelegate transport
)

Visual Basic (Declaration)

Public Sub RepSetTransport ( _
    envid As Integer, _
    transport As ReplicationTransportDelegate _
)

Visual C++

public:
void RepSetTransport(
    int envid,
    ReplicationTransportDelegate^ transport
)

Parameters

envid

Type: System::Int32
The local environment's ID. It must be a non-negative integer and uniquely identify this Berkeley DB database environment (see Replication environment IDs in the Programmer's Reference Guide for more information).

transport

Type: BerkeleyDB::ReplicationTransportDelegate
The delegate used to transmit data using the replication application's communication infrastructure.
**Remarks**

RepSetTransport is not called by most replication applications. It should only be called by applications implementing their own network transport layer, explicitly holding replication group elections and handling replication messages outside of the replication manager framework.
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
RepStartClient is not called by most replication applications. It should only be called by applications implementing their own network transport layer, explicitly holding replication group elections and handling replication messages outside of the replication manager framework.

Replication master environments are the only database environments where replicated databases may be modified. Replication client environments are read-only as long as they are clients. Replication client environments may be upgraded to be replication master environments in the case that the current master fails or there is no master present. If master leases are in use, this method cannot be used to appoint a master, and should only be used to configure a database environment as a master as the result of an election.

Before calling this method, the RepTransport delegate must already have been configured to send replication messages.
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RepStartClient()</td>
<td>Configure the database environment as a client in a group of replicated</td>
</tr>
<tr>
<td></td>
<td>database environments.</td>
</tr>
<tr>
<td>RepStartClient(DatabaseEntry)</td>
<td>Configure the database environment as a client in a group of replicated</td>
</tr>
<tr>
<td></td>
<td>database environments.</td>
</tr>
</tbody>
</table>
See Also

DatabaseEnvironment Class
DatabaseEnvironment Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Configure the database environment as a client in a group of replicated database environments.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public void RepStartClient()

Visual Basic (Declaration)

Public Sub RepStartClient

Visual C++

public:
void RepStartClient()
See Also

DatabaseEnvironment Class
RepStartClient Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Configure the database environment as a client in a group of replicated database environments.

**Namespace:**  BerkeleyDB  
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public void RepStartClient(
    DatabaseEntry cdata
)

Visual Basic (Declaration)

Public Sub RepStartClient ( _
    cdata As DatabaseEntry _
)

Visual C++

public:
void RepStartClient(
    DatabaseEntry^ cdata
)

Parameters

cdata

Type: BerkeleyDB::DatabaseEntry
An opaque data item that is sent over the communication infrastructure when the client comes online (see Connecting to a new site in the Programmer's Reference Guide for more information). If no such information is useful, cdata should be null.
See Also

DatabaseEnvironment Class
RepStartClient Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
RepStartMaster is not called by most replication applications. It should only be called by applications implementing their own network transport layer, explicitly holding replication group elections and handling replication messages outside of the replication manager framework.

Replication master environments are the only database environments where replicated databases may be modified. Replication client environments are read-only as long as they are clients. Replication client environments may be upgraded to be replication master environments in the case that the current master fails or there is no master present. If master leases are in use, this method cannot be used to appoint a master, and should only be used to configure a database environment as a master as the result of an election.

Before calling this method, the RepTransport delegate must already have been configured to send replication messages.
# Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RepStartMaster()()()</td>
<td>Configure the database environment as a master in a group of replicated</td>
</tr>
<tr>
<td></td>
<td>database environments.</td>
</tr>
<tr>
<td>RepStartMaster(DatabaseEntry)</td>
<td>Configure the database environment as a master in a group of replicated</td>
</tr>
<tr>
<td></td>
<td>database environments.</td>
</tr>
</tbody>
</table>
See Also

DatabaseEnvironment Class
DatabaseEnvironment Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Configure the database environment as a master in a group of replicated database environments.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public void RepStartMaster()

Visual Basic (Declaration)

Public Sub RepStartMaster

Visual C++

public:
void RepStartMaster()
See Also

DatabaseEnvironment Class
RepStartMaster Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Configure the database environment as a master in a group of replicated database environments.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public void RepStartMaster(
    DatabaseEntry cdata
)

Visual Basic (Declaration)

Public Sub RepStartMaster ( _
    cdata As DatabaseEntry _
)

Visual C++

public:
void RepStartMaster(
    DatabaseEntry^ cdata
)

Parameters

cdata
Type: BerkeleyDB..:::DatabaseEntry
An opaque data item that is sent over the communication infrastructure when the client comes online (see Connecting to a new site in the Programmer's Reference Guide for more information). If no such information is useful, cdata should be null.
See Also

DatabaseEnvironment Class
RepStartMaster Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
DatabaseEnvironment::RepSync Method

Force master synchronization to begin for this client.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public void RepSync()

Visual Basic (Declaration)

Public Sub RepSync

Visual C++

public:
void RepSync()
Remarks

This method is the other half of setting RepDelayClientSync.

If an application has configured delayed master synchronization, the application must synchronize explicitly (otherwise the client will remain out-of-date and will ignore all database changes forwarded from the replication group master). RepSync may be called any time after the client application learns that the new master has been established (by receiving REP_NEWMASTER).

Before calling this method, the RepTransport delegate must already have been configured to send replication messages.
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
DatabaseEnvironment ResetFileID Method

Allow database files to be copied, and then the copy used in the same database environment as the original.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public void ResetFileID(
    string file,
    bool encrypted
)

Visual Basic (Declaration)

Public Sub ResetFileID ( _
    file As String,
    encrypted As Boolean _
)

Visual C++

public:
void ResetFileID(
    String^ file,
    bool encrypted
)

Parameters

file
Type: System::String
The name of the physical file in which new file IDs are to be created.

encrypted
Type: System::Boolean
If true, the file contains encrypted databases.
Remarks

All databases contain an ID string used to identify the database in the database environment cache. If a physical database file is copied, and used in the same environment as another file with the same ID strings, corruption can occur. ResetFileID creates new ID strings for all of the databases in the physical file.

ResetFileID modifies the physical file, in-place. Applications should not reset IDs in files that are currently in use.
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Allow database files to be moved from one transactional database environment to another.

**Namespace:** [BerkeleyDB](https://www.oracle.com/technetwork/database/database-technology/berkeley-db-net-introduction-091240.html)

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public void ResetLSN(
    string file,
    bool encrypted
)

Visual Basic (Declaration)

Public Sub ResetLSN (_
    file As String,
    encrypted As Boolean _
)

Visual C++

public:
    void ResetLSN(
    String^ file,
    bool encrypted
)

Parameters

file
    Type: System::String

encrypted
    Type: System::Boolean
Remarks

Database pages in transactional database environments contain references to the environment's log files (that is, log sequence numbers, or LSNs). Copying or moving a database file from one database environment to another, and then modifying it, can result in data corruption if the LSNs are not first cleared.

Note that LSNs should be reset before moving or copying the database file into a new database environment, rather than moving or copying the database file and then resetting the LSNs. Berkeley DB has consistency checks that may be triggered if an application calls ResetLSN on a database in a new environment when the database LSNs still reflect the old environment.

The ResetLSN method modifies the physical file, in-place. Applications should not reset LSNs in files that are currently in use.
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
DatabaseEnvironment::SetMaxSequentialWrites Method

Limit the number of sequential write operations scheduled by the library when flushing dirty pages from the cache.

Namespace: BerkeleyDB
Syntax

C#

```csharp
public void SetMaxSequentialWrites(
    int maxWrites,
    uint pause
)
```

Visual Basic (Declaration)

```vbnet
Public Sub SetMaxSequentialWrites ( _
    maxWrites As Integer, _
    pause As UInteger _
)
```

Visual C++

```cpp
public:
void SetMaxSequentialWrites(  
    int maxWrites,
    unsigned int pause
)
```

Parameters

maxWrites

Type: `System::::Int32`

The maximum number of sequential write operations scheduled by the library when flushing dirty pages from the cache, or 0 if there is no limitation on the number of sequential write operations.

pause

Type: `System::::UInt32`

The number of microseconds the thread of control should pause before scheduling further write operations. It must be specified as an unsigned 32-bit number of microseconds, limiting the maximum pause to roughly 71 minutes.
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation
DatabaseEnvironment::SyncMemPool Method

DatabaseEnvironment Class  See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SyncMemPool()()</td>
<td>Flush all modified pages in the cache to their backing files.</td>
</tr>
<tr>
<td>SyncMemPool(LSN)</td>
<td>Flush modified pages in the cache with log sequence numbers less than minLSN to their backing files.</td>
</tr>
</tbody>
</table>
See Also

DatabaseEnvironment Class
DatabaseEnvironment Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Flush all modified pages in the cache to their backing files.

**Namespace:** [BerkeleyDB](#)  
**Assembly:** [libdb_dotnet48](#) (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public void SyncMemPool()

Visual Basic (Declaration)

Public Sub SyncMemPool

Visual C++

public:
void SyncMemPool()
Remarks

Pages in the cache that cannot be immediately written back to disk (for example, pages that are currently in use by another thread of control) are waited for and written to disk as soon as it is possible to do so.
See Also

DatabaseEnvironment Class
SyncMemPool Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Flush modified pages in the cache with log sequence numbers less than minLSN to their backing files.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public void SyncMemPool(
    LSN minLSN
)

Visual Basic (Declaration)

Public Sub SyncMemPool ( _
    minLSN As LSN _
)

Visual C++

public:
void SyncMemPool(
    LSN^ minLSN
)

Parameters

minLSN

Type: BerkeleyDB::::LSN
All modified pages with a log sequence number less than the minLSN parameter are written to disk. If null, all modified pages in the cache are written to disk.
Remarks

Pages in the cache that cannot be immediately written back to disk (for example, pages that are currently in use by another thread of control) are waited for and written to disk as soon as it is possible to do so.
See Also

DatabaseEnvironment Class
SyncMemPool Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation
DatabaseEnvironment...:::TransactionSystemStats Method

DatabaseEnvironment Class  See Also
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>TransactionSystemStats()</code></td>
<td>The transaction subsystem statistics</td>
</tr>
<tr>
<td><code>TransactionSystemStats(Boolean)</code></td>
<td>The transaction subsystem statistics</td>
</tr>
</tbody>
</table>
See Also

DatabaseEnvironment Class
DatabaseEnvironment Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The transaction subsystem statistics

**Namespace:** [BerkeleyDB](https://berkeleydb.com)

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

**C#**

public TransactionStats TransactionSystemStats()

**Visual Basic (Declaration)**

Public Function TransactionSystemStats As TransactionStats

**Visual C++**

public:
TransactionStats^ TransactionSystemStats()  

**Return Value**

The transaction subsystem statistics
See Also

DatabaseEnvironment Class
TransactionSystemStats Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation

DatabaseEnvironment...:::TransactionSystemStats Method (Boolean)

**DatabaseEnvironment Class**  See Also

The transaction subsystem statistics

**Namespace:**  BerkeleyDB

**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
**Syntax**

**C#**

```csharp
public TransactionStats TransactionSystemStats(
    bool clearStats
)
```

**Visual Basic (Declaration)**

```vbnet
Public Function TransactionSystemStats ( _
    clearStats As Boolean _
) As TransactionStats
```

**Visual C++**

```cpp
public:
TransactionStats^ TransactionSystemStats(
    bool clearStats
)
```

**Parameters**

clearStats
Type: System::::Boolean
If true, reset statistics after returning their values.

**Return Value**

The transaction subsystem statistics
See Also

DatabaseEnvironment Class
TransactionSystemStats Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Ensure that a specified percent of the pages in the cache are clean, by writing dirty pages to their backing files.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

```csharp
public int TrickleCleanMemPool(
    int pctClean
)
```

Visual Basic (Declaration)

```vbnet
Public Function TrickleCleanMemPool ( _
    pctClean As Integer _
) As Integer
```

Visual C++

```cpp
public:
    int TrickleCleanMemPool(
        int pctClean
    )
```

Parameters

pctClean

Type: `System::::Int32`

The percent of the pages in the cache that should be clean.

Return Value

The number of pages written to reach the specified percentage is copied.
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
WriteToLog allows applications to include information in the database environment log files, for later review using the db_printlog utility. This method is intended for debugging and performance tuning.
### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>WriteToLog(String)</code></td>
<td>Append an informational message to the Berkeley DB database environment log files.</td>
</tr>
<tr>
<td><code>WriteToLog(String, Transaction)</code></td>
<td>Append an informational message to the Berkeley DB database environment log files.</td>
</tr>
</tbody>
</table>
See Also

DatabaseEnvironment Class
DatabaseEnvironment Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Se Also

Append an informational message to the Berkeley DB database environment log files.

**Namespace:**  BerkeleyDB

**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public void WriteToLog(
    string str
)

Visual Basic (Declaration)

Public Sub WriteToLog (_
    str As String _
)

Visual C++

public:
void WriteToLog(
    String^ str
)

Parameters

str

Type: System::String
The message to append to the log files
See Also

DatabaseEnvironment Class
WriteToLog Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Append an informational message to the Berkeley DB database environment log files.

**Namespace:**  BerkeleyDB

**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public void WriteToLog(
    string str,
    Transaction txn
)

Visual Basic (Declaration)

Public Sub WriteToLog ( _
    str As String, _
    txn As Transaction _
)

Visual C++

public:
void WriteToLog(
    String^ str,
    Transaction^ txn
)

Parameters

str

Type: System::String
The message to append to the log files

txn

Type: BerkeleyDB::Transaction
If the operation is part of an application-specified transaction, txn is a Transaction object returned from BeginTransaction(); otherwise null.
See Also

DatabaseEnvironment Class
WriteToLog Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `DatabaseEnvironment` type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AutoCommit</td>
<td>If true, database operations for which no explicit transaction handle was specified, and which modify databases in the database environment, will be automatically enclosed within a transaction.</td>
</tr>
<tr>
<td>CacheSize</td>
<td>The size of the shared memory buffer pool -- that is, the cache.</td>
</tr>
<tr>
<td>CDB_ALLDB</td>
<td>If true, Berkeley DB Concurrent Data Store applications will perform locking on an environment-wide basis rather than on a per-database basis.</td>
</tr>
<tr>
<td>Create</td>
<td>If true, Berkeley DB subsystems will create any underlying files, as necessary.</td>
</tr>
<tr>
<td>DataDirs</td>
<td>The array of directories where database files are stored.</td>
</tr>
<tr>
<td>DeadlockResolution</td>
<td>The deadlock detector configuration, specifying what lock request(s) should be rejected. As transactions acquire locks on behalf of a single locker ID, rejecting a lock request associated with a transaction normally requires the transaction be aborted.</td>
</tr>
<tr>
<td>EncryptAlgorithm</td>
<td>The algorithm used by the Berkeley DB library to perform encryption and decryption.</td>
</tr>
<tr>
<td>ErrorFeedback</td>
<td>The mechanism for reporting detailed error messages to the application.</td>
</tr>
<tr>
<td>ErrorPrefix</td>
<td>The prefix string that appears before error messages issued by Berkeley DB.</td>
</tr>
</tbody>
</table>
**EventNotify**
A delegate which is called to notify the process of specific Berkeley DB events.

**Feedback**
Monitor progress within long running operations.
If true, flush database writes to the backing disk before returning from the write system call, rather than flushing database writes explicitly in a separate system call, as necessary.

**ForceFlush**
If true, the object is free-threaded; that is, concurrently usable by multiple threads in the address space.

**FreeThreaded**
If true, Berkeley DB will page-fault shared regions into memory when initially creating or joining a Berkeley DB environment.

**Home**
The database environment home directory.

**InitRegions**
If true, Berkeley DB will page-fault shared regions into memory when initially creating or joining a Berkeley DB environment.

**IntermediateDirMode**
The intermediate directory permissions.

**LockConflictMatrix**
The current lock conflicts array.
If true, lock shared Berkeley DB environment files and memory-mapped databases into memory.

**Lockdown**
The number of lock table partitions used in the Berkeley DB environment.

**LockPartitions**
A value, in microseconds, representing lock timeouts.

**LockTimeout**
If true, Berkeley DB will automatically remove log files that are no longer needed.

**LogAutoRemove**
The size of the in-memory log buffer, in bytes.

**LogBufferSize**
The path of a directory to be used as the location of logging files. Log files
LogDir
created by the Log Manager subsystem will be created in this directory.

LogFileMode
The absolute file mode for created log files. This property is only useful for the rare Berkeley DB application that does not control its umask value.

If true, Berkeley DB will flush log writes to the backing disk before returning from the write system call, rather than flushing log writes explicitly in a separate system call, as necessary.

LogForceSync
If true, Berkeley DB will flush log writes to the backing disk before returning from the write system call, rather than flushing log writes explicitly in a separate system call, as necessary.

If true, Berkeley DB will flush log writes to the backing disk before returning from the write system call, rather than flushing log writes explicitly in a separate system call, as necessary.

LogInMemory
If true, transaction logs are maintained in memory rather than on disk. This means that transactions exhibit the ACI (atomicity, consistency, and isolation) properties, but not D (durability).

If true, transaction logs are maintained in memory rather than on disk. This means that transactions exhibit the ACI (atomicity, consistency, and isolation) properties, but not D (durability).

LogNoBuffer
If true, system buffering is turned off for Berkeley DB log files to avoid double caching.

The size of the underlying logging area of the Berkeley DB environment, in bytes.

LogZeroOnCreate
If true, all pages of a log file are zeroed when that log file is created.

MaxCacheSize
The maximum cache size

The maximum cache size

MaxLockers
The maximum number of locking entities supported by the Berkeley DB environment.

The maximum number of locking entities supported by the Berkeley DB environment.

MaxLocks
The maximum size of a single file in the log, in bytes. Because LSN Offsets are unsigned four-byte values, the size may not be larger than the maximum
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MaxMutexes</strong></td>
<td>The total number of mutexes allocated</td>
</tr>
<tr>
<td><strong>MaxObjects</strong></td>
<td>The maximum number of locked objects</td>
</tr>
<tr>
<td><strong>MaxOpenFiles</strong></td>
<td>The number of file descriptors the library will open concurrently when flushing dirty pages from the cache.</td>
</tr>
<tr>
<td><strong>MaxSequentialWrites</strong></td>
<td>The number of sequential write operations scheduled by the library when flushing dirty pages from the cache.</td>
</tr>
<tr>
<td><strong>MaxTransactions</strong></td>
<td>The number of active transactions supported by the environment. This value bounds the size of the memory allocated for transactions. Child transactions are counted as active until they either commit or abort.</td>
</tr>
<tr>
<td><strong>MMapSize</strong></td>
<td>The maximum file size, in bytes, for a file to be mapped into the process address space. If no value is specified, it defaults to 10MB.</td>
</tr>
<tr>
<td><strong>MutexAlignment</strong></td>
<td>The mutex alignment, in bytes.</td>
</tr>
<tr>
<td><strong>MutexIncrement</strong></td>
<td>The number of additional mutexes allocated.</td>
</tr>
<tr>
<td><strong>NoBuffer</strong></td>
<td>If true, turn off system buffering of Berkeley DB database files to avoid double caching.</td>
</tr>
<tr>
<td><strong>NoLocking</strong></td>
<td>If true, Berkeley DB will grant all requested mutual exclusion mutexes and database locks without regard for their actual availability. This functionality should never be used for purposes other than debugging.</td>
</tr>
<tr>
<td><strong>NoMMap</strong></td>
<td>If true, Berkeley DB will copy read-only database files into the local cache instead of potentially mapping them</td>
</tr>
</tbody>
</table>
NoPanic

If true, Berkeley DB will ignore any panic state in the database environment. (Database environments in a panic state normally refuse all attempts to call Berkeley DB functions, throwing RunRecoveryException.) This functionality should never be used for purposes other than debugging.

The number of times that test-and-set mutexes should spin without blocking.

NumTestAndSetSpins

The value defaults to 1 on uniprocessor systems and to 50 times the number of processors on multiprocessor systems.

If true, overwrite files stored in encrypted formats before deleting them.

Overwrite

If true, allocate region memory from the heap instead of from memory backed by the filesystem or system shared memory.

Private

If true, Berkeley DB will have checked to see if recovery needed to be performed before opening the database environment.

Register

The amount of time the replication manager's transport function waits to collect enough acknowledgments from replication group clients, before giving up and returning a failure indication. The default wait time is 1 second.

RepAckTimeout

If true, the replication master sends groups of records to the clients in a single network transfer

RepBulkTransfer

The amount of time a master site will delay between completing a
checkpoint and writing a checkpoint record into the log.

The value, relative to `RepClockskewSlow`, of the fastest clock in the group of sites.

The value of the slowest clock in the group of sites.

The amount of time the replication manager will wait before trying to re-establish a connection to another site after a communication failure. The default wait time is 30 seconds.

If true, the client should delay synchronizing to a newly declared master (defaults to false). Clients configured in this way will remain unsynchronized until the application calls `RepSync()`.

Configure the amount of time the replication manager will wait before retrying a failed election. The default wait time is 10 seconds.

The timeout period for an election. The default timeout is 2 seconds.

An optional configuration timeout period to wait for full election participation the first time the replication group finds a master. By default this option is turned off and normal election timeouts are used. (See the Elections section in the Berkeley DB Reference Guide for more information.)

The amount of time the replication manager, running at a client site, waits for some message activity on the
**RepHeartbeatMonitor**

connection from the master (heartbeats or other messages) before concluding that the connection has been lost. When 0 (the default), no monitoring is performed.

**RepHeartbeatSend**

The frequency at which the replication manager, running at a master site, broadcasts a heartbeat message in an otherwise idle system. When 0 (the default), no heartbeat messages will be sent.

The amount of time a client grants its master lease to a master. When using master leases all sites in a replication group must use the same lease timeout value. There is no default value. If leases are desired, this method must be called prior to calling `RepStartClient()` or `RepStartMaster()`.

**RepLeaseTimeout**

Specify how master and client sites will handle acknowledgment of replication messages which are necessary for "permanent" records. The current implementation requires all sites in a replication group configure the same acknowledgement policy.

**RepMgrAckPolicy**

The host information for the local system.

**RepMgrLocalSite**

The status of the sites currently known by the replication manager.

If true, the replication master will not automatically re-initialize outdated clients (defaults to false).

**RepMgrRemoteSites**

If true, Berkeley DB method calls that would normally block while clients are
RepNoBlocking

In recovery will return errors immediately (defaults to false).

RepNSites

The total number of sites in the replication group.

The database environment's priority in replication group elections. A special value of 0 indicates that this environment cannot be a replication group master. If not configured, then a default value of 100 is used.

RepPriority

A special value of 0 indicates that this environment cannot be a replication group master. If not configured, then a default value of 100 is used.

RepRetransmissionRequestMax

The maximum number of microseconds a client waits before requesting retransmission.

RepRetransmissionRequestMin

The minimum number of microseconds a client waits before requesting retransmission.

RepStrict2Site

Replication Manager observes the strict "majority" rule in managing elections, even in a group with only 2 sites. This means the client in a 2-site group will be unable to take over as master if the original master fails or becomes disconnected. (See the Elections section in the Berkeley DB Reference Guide for more information.) Both sites in the replication group should have the same value for this parameter.

RepTransmitLimitBytes

The bytes component of the byte-count limit on the amount of data that will be transmitted from a site in response to a single message processed by RepProcessMessage(DatabaseEntry, DatabaseEntry, Int32).

The gigabytes component of the byte-count limit on the amount of data that will be transmitted from a site in
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RepTransmitLimitGBytes</td>
<td>response to a single message processed by RepProcessMessage(DatabaseEntry, DatabaseEntry, Int32)</td>
</tr>
<tr>
<td>RepTransport</td>
<td>The delegate used to transmit data using the replication application's communication infrastructure.</td>
</tr>
<tr>
<td>RepUseMasterLeases</td>
<td>If true, master leases will be used for this site (defaults to false).</td>
</tr>
<tr>
<td>RunFatalRecovery</td>
<td>If true, catastrophic recovery was run on this environment before opening it for normal use.</td>
</tr>
<tr>
<td>RunRecovery</td>
<td>If true, normal recovery was run on this environment before opening it for normal use.</td>
</tr>
<tr>
<td>SequentialWritePause</td>
<td>The number of microseconds the thread of control will pause before scheduling further write operations.</td>
</tr>
<tr>
<td>SetThreadID</td>
<td>A delegate that returns a unique identifier pair for the current thread of control.</td>
</tr>
<tr>
<td>SetThreadName</td>
<td>A delegate that formats a process ID and thread ID identifier pair.</td>
</tr>
<tr>
<td>SystemMemory</td>
<td>If true, allocate region memory from system shared memory instead of from heap memory or memory backed by the filesystem.</td>
</tr>
<tr>
<td>TempDir</td>
<td>The path of a directory to be used as the location of temporary files.</td>
</tr>
<tr>
<td>ThreadCount</td>
<td>An approximate number of threads in the database environment.</td>
</tr>
<tr>
<td>ThreadIsAlive</td>
<td>A delegate that returns if a thread of control (either a true thread or a process) is still running.</td>
</tr>
<tr>
<td></td>
<td>If true, database calls timing out based on lock or transaction timeout values</td>
</tr>
<tr>
<td><strong>Key</strong></td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>TimeNotGranted</strong></td>
<td>will throw <strong>LockNotGrantedException</strong> instead of <strong>DeadlockException</strong>.</td>
</tr>
<tr>
<td><strong>TxnNoSync</strong></td>
<td>If true, Berkeley DB will not write or synchronously flush the log on</td>
</tr>
<tr>
<td></td>
<td>transaction commit.</td>
</tr>
<tr>
<td><strong>TxnNoWait</strong></td>
<td>If true and a lock is unavailable for any Berkeley DB operation performed in</td>
</tr>
<tr>
<td></td>
<td>the context of a transaction, cause the operation to throw <strong>DeadlockException</strong> (or <strong>LockNotGrantedException</strong> if configured with <strong>TimeNotGranted</strong>).</td>
</tr>
<tr>
<td><strong>TxnSnapshot</strong></td>
<td>If true, all transactions in the environment will be started as if <strong>Snapshot</strong> was passed to <strong>BeginTransaction()</strong>.</td>
</tr>
<tr>
<td></td>
<td>and all non-transactional cursors will be opened as if <strong>SnapshotIsolation</strong> was passed to <strong>Cursor()</strong>.</td>
</tr>
<tr>
<td><strong>TxnTimeout</strong></td>
<td>A value, in microseconds, representing transaction timeouts.</td>
</tr>
<tr>
<td><strong>TxnTimestamp</strong></td>
<td>The recovery timestamp.</td>
</tr>
<tr>
<td><strong>TxnWriteNoSync</strong></td>
<td>If true, Berkeley DB will write, but will not synchronously flush, the log</td>
</tr>
<tr>
<td></td>
<td>on transaction commit.</td>
</tr>
<tr>
<td><strong>UseEnvironmentVars</strong></td>
<td>The Berkeley DB process' environment may be permitted to specify information to be used when naming files; see Berkeley DB File Naming in the Programmer's Reference Guide for more information.</td>
</tr>
<tr>
<td><strong>UseMVCC</strong></td>
<td>If true, all databases in the environment will be opened as if <strong>UseMVCC</strong> was set.</td>
</tr>
<tr>
<td><strong>UsingCDB</strong></td>
<td>If true, locking for the Berkeley DB Concurrent Data Store product was</td>
</tr>
<tr>
<td></td>
<td>initialized.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>UsingLocking</strong></td>
<td>If true, the locking subsystem was initialized.</td>
</tr>
<tr>
<td><strong>UsingLogging</strong></td>
<td>If true, the logging subsystem was initialized.</td>
</tr>
<tr>
<td><strong>UsingMPool</strong></td>
<td>If true, the shared memory buffer pool subsystem was initialized.</td>
</tr>
<tr>
<td><strong>UsingReplication</strong></td>
<td>If true, the replication subsystem was initialized.</td>
</tr>
<tr>
<td><strong>UsingTxns</strong></td>
<td>If true, the transaction subsystem was initialized.</td>
</tr>
<tr>
<td><strong>Verbosity</strong></td>
<td>Specific additional informational and debugging messages in the Berkeley DB message output.</td>
</tr>
<tr>
<td><strong>YieldCPU</strong></td>
<td>If true, Berkeley DB will yield the processor immediately after each page or mutex acquisition.</td>
</tr>
</tbody>
</table>
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, database operations for which no explicit transaction handle was specified, and which modify databases in the database environment, will be automatically enclosed within a transaction.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

```csharp
public bool AutoCommit { get; set; }
```

Visual Basic (Declaration)

```vbnet
Public Property AutoCommit As Boolean
```

Visual C++

```c++
public:
property bool AutoCommit {
    bool get ();
    void set (bool value);
}
```
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The size of the shared memory buffer pool -- that is, the cache.

**Namespace:** BerkeleyDB
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public CacheInfo CacheSize { get; set; }

Visual Basic (Declaration)

Public Property CacheSize As CacheInfo

Visual C++

public:
property CacheInfo^ CacheSize {
    CacheInfo^ get ();
    void set (CacheInfo^ value);
}
Remarks

The cache should be the size of the normal working data set of the application, with some small amount of additional memory for unusual situations. (Note: the working set is not the same as the number of pages accessed simultaneously, and is usually much larger.)

The default cache size is 256KB, and may not be specified as less than 20KB. Any cache size less than 500MB is automatically increased by 25% to account for buffer pool overhead; cache sizes larger than 500MB are used as specified. The maximum size of a single cache is 4GB on 32-bit systems and 10TB on 64-bit systems. (All sizes are in powers-of-two, that is, 256KB is $2^{18}$ not 256,000.) For information on tuning the Berkeley DB cache size, see Selecting a cache size in the Programmer's Reference Guide.
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, Berkeley DB Concurrent Data Store applications will perform locking on an environment-wide basis rather than on a per-database basis.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool CDB_ALLDB { get; }

Visual Basic (Declaration)

Public ReadOnly Property CDB_ALLDB As Boolean

Visual C++

public:
property bool CDB_ALLDB {
    bool get ();
}

}
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, Berkeley DB subsystems will create any underlying files, as necessary.

Namespace: BerkeleyDB
Syntax

C#

```csharp
public bool Create { get; }
```

Visual Basic (Declaration)

```vbnet
Public ReadOnly Property Create As Boolean
```

Visual C++

```cpp
public:
property bool Create {
    bool get ();
}
```
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The array of directories where database files are stored.

**Namespace:** [BerkeleyDB](https://www.oracle.com/technetwork/database/database-technologies/berkeleydb/index.html)

**Assembly:** `libdb_dotnet48` (in `libdb_dotnet48.dll`) Version: 4.8.24.0
Syntax

C#

public List<string> DataDirs { get; }

Visual Basic (Declaration)

Public ReadOnly Property DataDirs As List(Of String)

Visual C++

public:
property List<String^> DataDirs {
    List<String^> get ();
}

See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation

DatabaseEnvironment...:::DeadlockResolution Property

DatabaseEnvironment Class  See Also

The deadlock detector configuration, specifying what lock request(s) should be rejected. As transactions acquire locks on behalf of a single locker ID, rejecting a lock request associated with a transaction normally requires the transaction be aborted.

Namespace:  BerkeleyDB
Syntax

C#

public DeadlockPolicy DeadlockResolution { get; set; }

Visual Basic (Declaration)

Public Property DeadlockResolution As DeadlockPolicy

Visual C++

public:
property DeadlockPolicy^ DeadlockResolution {
    DeadlockPolicy^ get ();
    void set (DeadlockPolicy^ value);
}
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The algorithm used by the Berkeley DB library to perform encryption and decryption.

**Namespace:**  BerkeleyDB  
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public EncryptionAlgorithm EncryptAlgorithm { get; }

Visual Basic (Declaration)

Public ReadOnly Property EncryptAlgorithm As EncryptionAlgorithm

Visual C++

public:
    property EncryptionAlgorithm EncryptAlgorithm {
        EncryptionAlgorithm get ();
    }
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation
DatabaseEnvironment...::.ErrorFeedback Property

DatabaseEnvironment Class  See Also

The mechanism for reporting detailed error messages to the application.

Namespace:  BerkeleyDB
Syntax

C#

public ErrorFeedbackDelegate ErrorFeedback { get; set; }

Visual Basic (Declaration)

Public Property ErrorFeedback As ErrorFeedbackDelegate

Visual C++

public:
property ErrorFeedbackDelegate^ ErrorFeedback {
    ErrorFeedbackDelegate^ get ();
    void set (ErrorFeedbackDelegate^ value);
}
Remarks

When an error occurs in the Berkeley DB library, a `DatabaseException`, or subclass of `DatabaseException`, is thrown. In some cases, however, the exception may be insufficient to completely describe the cause of the error, especially during initial application debugging.

In some cases, when an error occurs, Berkeley DB will call the given delegate with additional error information. It is up to the delegate to display the error message in an appropriate manner.

Setting `ErrorFeedback` to NULL unconfigures the callback interface.

This error-logging enhancement does not slow performance or significantly increase application size, and may be run during normal operation as well as during application debugging.
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The prefix string that appears before error messages issued by Berkeley DB.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public string ErrorPrefix { get; set; }

Visual Basic (Declaration)

Public Property ErrorPrefix As String

Visual C++

public:
property String^ ErrorPrefix {
    String^ get ();
    void set (String^ value);
}
Remarks

For databases opened inside of a DatabaseEnvironment, setting ErrorPrefix affects the entire environment and is equivalent to setting ErrorPrefix.

Setting ErrorPrefix configures operations performed using the specified object, not all operations performed on the underlying database.
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
A delegate which is called to notify the process of specific Berkeley DB events.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public EventNotifyDelegate EventNotify { get; set; }

Visual Basic (Declaration)

Public Property EventNotify As EventNotifyDelegate

Visual C++

public:
property EventNotifyDelegate^ EventNotify {
    EventNotifyDelegate^ get ();
    void set (EventNotifyDelegate^ value);
}
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Monitor progress within long running operations.

**Namespace:** [BerkeleyDB](#)  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll)  
Version: 4.8.24.0
Syntax

C#

public EnvironmentFeedbackDelegate Feedback { get; set; }

Visual Basic (Declaration)

Public Property Feedback As EnvironmentFeedbackDelegate

Visual C++

public:
property EnvironmentFeedbackDelegate^ Feedback {
    EnvironmentFeedbackDelegate^ get ();
    void set (EnvironmentFeedbackDelegate^ value);
}
Remarks

Some operations performed by the Berkeley DB library can take non-trivial amounts of time. The Feedback delegate can be used by applications to monitor progress within these operations. When an operation is likely to take a long time, Berkeley DB will call the specified delegate with progress information.

It is up to the delegate to display this information in an appropriate manner.
If true, flush database writes to the backing disk before returning from the write system call, rather than flushing database writes explicitly in a separate system call, as necessary.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool ForceFlush { get; set; }

Visual Basic (Declaration)

Public Property ForceFlush As Boolean

Visual C++

public:
property bool ForceFlush {
    bool get ();
    void set (bool value);
}
Remarks

This flag may result in inaccurate file modification times and other file-level information for Berkeley DB database files. This flag will almost certainly result in a performance decrease on most systems.
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, the object is free-threaded; that is, concurrently usable by multiple threads in the address space.

Namespace: BerkeleyDB
Syntax

C#

public bool FreeThreaded { get; }

Visual Basic (Declaration)

Public ReadOnly Property FreeThreaded As Boolean

Visual C++

public:
property bool FreeThreaded {
    bool get ();
}


See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The database environment home directory.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public string Home { get; }

Visual Basic (Declaration)

Public ReadOnly Property Home As String

Visual C++

public:
property String^ Home {
    String^ get ();
}

See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, Berkeley DB will page-fault shared regions into memory when initially creating or joining a Berkeley DB environment.

**Namespace:**  BerkeleyDB

**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool InitRegions { get; set; }

Visual Basic (Declaration)

Public Property InitRegions As Boolean

Visual C++

public:
property bool InitRegions {
    bool get ();
    void set (bool value);
}
Remarks

In some applications, the expense of page-faulting the underlying shared memory regions can affect performance. (For example, if the page-fault occurs while holding a lock, other lock requests can convoy, and overall throughput may decrease.)

In addition to page-faulting, Berkeley DB will write the shared regions when creating an environment, forcing the underlying virtual memory and filesystems to instantiate both the necessary memory and the necessary disk space. This can also avoid out-of-disk space failures later on.
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The intermediate directory permissions.

**Namespace:**  BerkeleyDB

**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
**Syntax**

**C#**

```csharp
public string IntermediateDirMode { get; private set; }
```

**Visual Basic (Declaration)**

```
Public Property IntermediateDirMode As String
```

**Visual C++**

```cpp
public:
property String^ IntermediateDirMode {
    String^ get ();
    void set (String^ value);
}
```
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The current lock conflicts array.

**Namespace:** [BerkeleyDB](https://docs.oracle.com/cd/E19612-01/820-9713/index.html)

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public byte[,] LockConflictMatrix { get; private set; }

Visual Basic (Declaration)

Public Property LockConflictMatrix As Byte(,)

Visual C++

public:
property array<unsigned char,2>^ LockConflictMatrix {
array<unsigned char,2>^ get ();
void set (array<unsigned char,2>^ value);
}

See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, lock shared Berkeley DB environment files and memory-mapped databases into memory.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool Lockdown { get; }

Visual Basic (Declaration)

Public ReadOnly Property Lockdown As Boolean

Visual C++

public:
property bool Lockdown {
    bool get ();
}


See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
DatabaseEnvironment.LockPartitions Property

The number of lock table partitions used in the Berkeley DB environment.

Namespace: BerkeleyDB
Syntax

C#

public uint LockPartitions { get; private set; }

Visual Basic (Declaration)

Public Property LockPartitions AsUInteger

Visual C++

public:
property unsigned int LockPartitions {
    unsigned int get ();
    void set (unsigned int value);
}
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
A value, in microseconds, representing lock timeouts.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public uint LockTimeout { get; set; }

Visual Basic (Declaration)

Public Property LockTimeout As UInteger

Visual C++

public:
property unsigned int LockTimeout {
    unsigned int get ();
    void set (unsigned int value);
}

Remarks

All timeouts are checked whenever a thread of control blocks on a lock or when deadlock detection is performed. As timeouts are only checked when the lock request first blocks or when deadlock detection is performed, the accuracy of the timeout depends on how often deadlock detection is performed.

Timeout values specified for the database environment may be overridden on a per-transaction basis, see SetLockTimeout(UInt32).
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, Berkeley DB will automatically remove log files that are no longer needed.

**Namespace:**  [BerkeleyDB](http://example.com)

**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool LogAutoRemove { get; set; }

Visual Basic (Declaration)

Public Property LogAutoRemove As Boolean

Visual C++

public:
property bool LogAutoRemove {
    bool get ();
    void set (bool value);
}
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The size of the in-memory log buffer, in bytes

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public uint LogBufferSize { get; private set; }

Visual Basic (Declaration)

Public Property LogBufferSize As UInteger

Visual C++

public:
    property unsigned int LogBufferSize {
        unsigned int get();
        void set (unsigned int value);
    }
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The path of a directory to be used as the location of logging files. Log files created by the Log Manager subsystem will be created in this directory.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public string LogDir { get; private set; }

Visual Basic (Declaration)

Public Property LogDir As String

Visual C++

public:
property String^ LogDir {
    String^ get ();
    void set (String^ value);
}
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The absolute file mode for created log files. This property is only useful for the rare Berkeley DB application that does not control its umask value.

Namespace: **BerkeleyDB**

Syntax

C#

public int LogFileMode { get; set; }

Visual Basic (Declaration)

Public Property LogFileMode As Integer

Visual C++

public:
property int LogFileMode {
    int get ();
    void set (int value);
}
Remarks

Normally, if Berkeley DB applications set their umask appropriately, all processes in the application suite will have read permission on the log files created by any process in the application suite. However, if the Berkeley DB application is a library, a process using the library might set its umask to a value preventing other processes in the application suite from reading the log files it creates. In this rare case, this property can be used to set the mode of created log files to an absolute value.
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, Berkeley DB will flush log writes to the backing disk before returning from the write system call, rather than flushing log writes explicitly in a separate system call, as necessary.

**Namespace:**  BerkeleyDB  
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll)  
Version: 4.8.24.0
## Syntax

### C#

```csharp
public bool LogForceSync { get; set; }
```

### Visual Basic (Declaration)

```vbnet
Public Property LogForceSync As Boolean
```

### Visual C++

```cpp
public:
property bool LogForceSync {
    bool get ();
    void set (bool value);
}
```
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, transaction logs are maintained in memory rather than on disk. This means that transactions exhibit the ACI (atomicity, consistency, and isolation) properties, but not D (durability).

Namespace: **BerkeleyDB**  
Syntax

C#

public bool LogInMemory { get; }

Visual Basic (Declaration)

Public ReadOnly Property LogInMemory As Boolean

Visual C++

public:
property bool LogInMemory {
    bool get();
}

See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, system buffering is turned off for Berkeley DB log files to avoid double caching.

**Namespace:** [BerkeleyDB](#)  
**Assembly:** `libdb_dotnet48` (in `libdb_dotnet48.dll`) Version: 4.8.24.0
Syntax

C#

public bool LogNoBuffer { get; set; }

Visual Basic (Declaration)

Public Property LogNoBuffer As Boolean

Visual C++

public:
property bool LogNoBuffer {
    bool get ();
    void set (bool value);
}
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
DatabaseEnvironment.LogRegionSize Property

The size of the underlying logging area of the Berkeley DB environment, in bytes.

Namespace: BerkeleyDB
Syntax

C#

public uint LogRegionSize { get; private set; }

Visual Basic (Declaration)

Public Property LogRegionSize As UInteger

Visual C++

public:
property unsigned int LogRegionSize {
    unsigned int get ();
    void set (unsigned int value);
}
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, all pages of a log file are zeroed when that log file is created.

**Namespace:**  [BerkeleyDB](https://www.berkleydb.com/)

**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool LogZeroOnCreate { get; }

Visual Basic (Declaration)

Public ReadOnly Property LogZeroOnCreate As Boolean

Visual C++

public:
property bool LogZeroOnCreate {
    bool get ();
}

See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
DatabaseEnvironment Class  See Also

The maximum cache size

**Namespace:**  BerkeleyDB
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

**C#**

```csharp
public CacheInfo MaxCacheSize { get; private set; }
```

**Visual Basic (Declaration)**

```vbnet
Public Property MaxCacheSize As CacheInfo
```

**Visual C++**

```cpp
public:
property CacheInfo^ MaxCacheSize {
    CacheInfo^ get ();
    void set (CacheInfo^ value);
}
```
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The maximum number of locking entities supported by the Berkeley DB environment.

**Namespace:** BerkeleyDB
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public uint MaxLockers { get; private set; }

Visual Basic (Declaration)

Public Property MaxLockers As UInteger

Visual C++

public:
property unsigned int MaxLockers {
    unsigned int get ();
    void set (unsigned int value);
}
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The maximum number of locks supported by the Berkeley DB environment.

**Namespace:** [BerkeleyDB](#)

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public uint MaxLocks { get; private set; }

Visual Basic (Declaration)

Public Property MaxLocks As UInteger

Visual C++

public:
property unsigned int MaxLocks {
    unsigned int get();
    void set (unsigned int value);
}
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
DatabaseEnvironment MaxLogFileSize Property

The maximum size of a single file in the log, in bytes. Because LSN Offsets are unsigned four-byte values, the size may not be larger than the maximum unsigned four-byte value.

Namespace: BerkeleyDB
Syntax

C#

public uint MaxLogFileSize { get; set; }

Visual Basic (Declaration)

Public Property MaxLogFileSize AsUInteger

Visual C++

public:
property unsigned int MaxLogFileSize {
    unsigned int get();
    void set (unsigned int value);
}
Remarks

When the logging subsystem is configured for on-disk logging, the default size of a log file is 10MB.

When the logging subsystem is configured for in-memory logging, the default size of a log file is 256KB. In addition, the configured log buffer size must be larger than the log file size. (The logging subsystem divides memory configured for in-memory log records into "files", as database environments configured for in-memory log records may exchange log records with other members of a replication group, and those members may be configured to store log records on-disk.) When choosing log buffer and file sizes for in-memory logs, applications should ensure the in-memory log buffer size is large enough that no transaction will ever span the entire buffer, and avoid a state where the in-memory buffer is full and no space can be freed because a transaction that started in the first log "file" is still active.


If no size is specified by the application, the size last specified for the database region will be used, or if no database region previously existed, the default will be used.
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The total number of mutexes allocated

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
## Syntax

### C#

```csharp
public uint MaxMutexes { get; private set; }
```

### Visual Basic (Declaration)

```vbnet
Public Property MaxMutexes As UInteger
```

### Visual C++

```cpp
public:
property unsigned int MaxMutexes {
    unsigned int get ();
    void set (unsigned int value);
}
```
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation
DatabaseEnvironment...::MaxObjects Property

**DatabaseEnvironment Class**  **See Also**

The maximum number of locked objects

**Namespace:**  BerkeleyDB
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

```csharp
public uint MaxObjects { get; private set; }
```

Visual Basic (Declaration)

```vbnet
Public Property MaxObjects As UInteger
```

Visual C++

```cpp
public:
property unsigned int MaxObjects {
 unsigned int get ();
 void set (unsigned int value);
}
```
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The number of file descriptors the library will open concurrently when flushing dirty pages from the cache.

**Namespace:**  [BerkeleyDB](https://www.berkeleydb.net)

**Assembly:**  `libdb_dotnet48` (in `libdb_dotnet48.dll`) Version: 4.8.24.0
Syntax

C#

public int MaxOpenFiles { get; set; }

Visual Basic (Declaration)

Public Property MaxOpenFiles As Integer

Visual C++

public:  
property int MaxOpenFiles {
    int get ();
    void set (int value);
}
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation
DatabaseEnvironment...MaxSequentialWrites Property

DatabaseEnvironment Class  See Also

The number of sequential write operations scheduled by the library when flushing dirty pages from the cache.

Namespace:  BerkeleyDB
Syntax

C#

public int MaxSequentialWrites { get; }

Visual Basic (Declaration)

Public ReadOnly Property MaxSequentialWrites As Integer

Visual C++

public:
property int MaxSequentialWrites {
    int get ();
}

See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The number of active transactions supported by the environment. This value bounds the size of the memory allocated for transactions. Child transactions are counted as active until they either commit or abort.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public uint MaxTransactions { get; set; }

Visual Basic (Declaration)

Public Property MaxTransactions As UInteger

Visual C++

public:
property unsigned int MaxTransactions {
    unsigned int get ();
    void set (unsigned int value);
}
Remarks

Transactions that update multiversion databases are not freed until the last page version that the transaction created is flushed from cache. This means that applications using multi-version concurrency control may need a transaction for each page in cache, in the extreme case.

When all of the memory available in the database environment for transactions is in use, calls to `BeginTransaction()` will fail (until some active transactions complete). If `MaxTransactions` is never set, the database environment is configured to support at least 100 active transactions.
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
DatabaseEnvironment MMapSize Property

The maximum file size, in bytes, for a file to be mapped into the process address space. If no value is specified, it defaults to 10MB.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#
public uint MMapSize { get; set; }

Visual Basic (Declaration)
Public Property MMapSize As UInteger

Visual C++
public:
property unsigned int MMapSize {
    unsigned int get ();
    void set (unsigned int value);
}
Remarks

Files that are opened read-only in the cache (and that satisfy a few other criteria) are, by default, mapped into the process address space instead of being copied into the local cache. This can result in better-than-usual performance because available virtual memory is normally much larger than the local cache, and page faults are faster than page copying on many systems. However, it can cause resource starvation in the presence of limited virtual memory, and it can result in immense process sizes in the presence of large databases.
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The mutex alignment, in bytes.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public uint MutexAlignment { get; private set; }

Visual Basic (Declaration)

Public Property MutexAlignment AsUInteger

Visual C++

public:
property unsigned int MutexAlignment {
    unsigned int get();
    void set (unsigned int value);
}
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
DatabaseEnvironment Class  See Also

The number of additional mutexes allocated.

Namespace:  BerkeleyDB
Syntax

C#

public uint MutexIncrement { get; private set; }

Visual Basic (Declaration)

Public Property MutexIncrement AsUInteger

Visual C++

public:
property unsigned int MutexIncrement {
    unsigned int get();
    void set (unsigned int value);
}
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, turn off system buffering of Berkeley DB database files to avoid double caching.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool NoBuffer { get; set; }

Visual Basic (Declaration)

Public Property NoBuffer As Boolean

Visual C++

public:
property bool NoBuffer {
    bool get ();
    void set (bool value);
}
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, Berkeley DB will grant all requested mutual exclusion mutexes and database locks without regard for their actual availability. This functionality should never be used for purposes other than debugging.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool NoLocking { get; set; }

Visual Basic (Declaration)

Public Property NoLocking As Boolean

Visual C++

public:
property bool NoLocking {
    bool get ();
    void set (bool value);
}
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, Berkeley DB will copy read-only database files into the local cache instead of potentially mapping them into process memory.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool NoMMap { get; set; }

Visual Basic (Declaration)

Public Property NoMMap As Boolean

Visual C++

public:
    property bool NoMMap {
        bool get ();
        void set (bool value);
    }
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace
DatabaseEnvironment::MMapSize

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, Berkeley DB will ignore any panic state in the database environment. (Database environments in a panic state normally refuse all attempts to call Berkeley DB functions, throwing RunRecoveryException.) This functionality should never be used for purposes other than debugging.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool NoPanic { get; set; }

Visual Basic (Declaration)

Public Property NoPanic As Boolean

Visual C++

public:
property bool NoPanic {
    bool get ();
    void set (bool value);
}
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The number of times that test-and-set mutexes should spin without blocking. The value defaults to 1 on uniprocessor systems and to 50 times the number of processors on multiprocessor systems.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public uint NumTestAndSetSpins { get; set; }

Visual Basic (Declaration)

Public Property NumTestAndSetSpins AsUInteger

Visual C++

public:
property unsigned int NumTestAndSetSpins {
    unsigned int get ();
    void set (unsigned int value);
}
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, overwrite files stored in encrypted formats before deleting them.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool Overwrite { get; set; }

Visual Basic (Declaration)

Public Property Overwrite As Boolean

Visual C++

public:
property bool Overwrite {
    bool get ();
    void set (bool value);
}

Remarks

Berkeley DB overwrites files using alternating 0xff, 0x00 and 0xff byte patterns. For file overwriting to be effective, the underlying file must be stored on a fixed-block filesystem. Systems with journaling or logging filesystems will require operating system support and probably modification of the Berkeley DB sources.
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, allocate region memory from the heap instead of from memory backed by the filesystem or system shared memory.

Namespace: BerkeleyDB
Syntax

C#

public bool Private { get; }

Visual Basic (Declaration)

Public ReadOnly Property Private As Boolean

Visual C++

public:
property bool Private {
    bool get ();
}
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, Berkeley DB will have checked to see if recovery needed to be performed before opening the database environment.

**Namespace:**  BerkeleyDB  
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool Register { get; }

Visual Basic (Declaration)

Public ReadOnly Property Register As Boolean

Visual C++

public:
property bool Register {
    bool get ();
}

See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The amount of time the replication manager's transport function waits to collect enough acknowledgments from replication group clients, before giving up and returning a failure indication. The default wait time is 1 second.

**Namespace:**  BerkeleyDB  
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
**Syntax**

**C#**

```csharp
public uint RepAckTimeout { get; set; }
```

**Visual Basic (Declaration)**

```
Public Property RepAckTimeout As UInteger
```

**Visual C++**

```c++
public:
property unsigned int RepAckTimeout {
    unsigned int get ();
    void set (unsigned int value);
}
```
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, the replication master sends groups of records to the clients in a single network transfer

**Namespace:**  BerkeleyDB

**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
C#

```csharp
public bool RepBulkTransfer { get; set; }
```

Visual Basic (Declaration)

```vbnet
Public Property RepBulkTransfer As Boolean
```

Visual C++

```c++
public:
    property bool RepBulkTransfer {
        bool get();
        void set (bool value);
    }
```
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The amount of time a master site will delay between completing a checkpoint and writing a checkpoint record into the log.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
## Syntax

### C#

```csharp
public uint RepCheckpointDelay { get; set; }
```

### Visual Basic (Declaration)

```vbnet
Public Property RepCheckpointDelay As UInteger
```

### Visual C++

```cpp
public:
property unsigned int RepCheckpointDelay {
    unsigned int get ();
    void set (unsigned int value);
}
```
Remarks

This delay allows clients to complete their own checkpoints before the master requires completion of them. The default is 30 seconds. If all databases in the environment, and the environment's transaction log, are configured to reside in memory (never preserved to disk), then, although checkpoints are still necessary, the delay is not useful and should be set to 0.
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The value, relative to `RepClockskewSlow`, of the fastest clock in the group of sites.

**Namespace:**  BerkeleyDB  
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

```
public uint RepClockskewFast { get; }
```

Visual Basic (Declaration)

```
Public ReadOnly Property RepClockskewFast AsUInteger
```

Visual C++

```
public:
property unsigned int RepClockskewFast {
    unsigned int get ();
}
```
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
RepClockskewSlow Property

DatabaseEnvironment Class  See Also

The value of the slowest clock in the group of sites.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public uint RepClockskewSlow { get; }

Visual Basic (Declaration)

Public ReadOnly Property RepClockskewSlow AsUInteger

Visual C++

public:
property unsigned int RepClockskewSlow {
    unsigned int get ();
}
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
DatabaseEnvironment..::.RepConnectionRetry Property

The amount of time the replication manager will wait before trying to re-establish a connection to another site after a communication failure. The default wait time is 30 seconds.

Namespace: BerkeleyDB
 syntax

C#

```csharp
public uint RepConnectionRetry { get; set; }
```

Visual Basic (Declaration)

```vbnet
Public Property RepConnectionRetry AsUInteger
```

Visual C++

```cpp
public:
property unsigned int RepConnectionRetry {
    unsigned int get();
    void set (unsigned int value);
}
```
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, the client should delay synchronizing to a newly declared master (defaults to false). Clients configured in this way will remain unsynchronized until the application calls RepSync().
Syntax

C#

public bool RepDelayClientSync { get; set; }

Visual Basic (Declaration)

Public Property RepDelayClientSync As Boolean

Visual C++

public:
property bool RepDelayClientSync {
    bool get ();
    void set (bool value);
}
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Configure the amount of time the replication manager will wait before retrying a failed election. The default wait time is 10 seconds.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public uint RepElectionRetry { get; set; }

Visual Basic (Declaration)

Public Property RepElectionRetry AsUInteger

Visual C++

public:
property unsigned int RepElectionRetry {
    unsigned int get();
    void set(unsigned int value);
}
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
DatabaseEnvironment::RepElectionTimeout Property

The timeout period for an election. The default timeout is 2 seconds.

Namespace: BerkeleyDB
Syntax

C#

public uint RepElectionTimeout { get; set; }

Visual Basic (Declaration)

Public Property RepElectionTimeout AsUInteger

Visual C++

public:
property unsigned int RepElectionTimeout {
    unsigned int get ();
    void set (unsigned int value);
}
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
An optional configuration timeout period to wait for full election participation the first time the replication group finds a master. By default this option is turned off and normal election timeouts are used. (See the Elections section in the Berkeley DB Reference Guide for more information.)

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public uint RepFullElectionTimeout { get; set; }

Visual Basic (Declaration)

Public Property RepFullElectionTimeout AsUInteger

Visual C++

public:
property unsigned int RepFullElectionTimeout {
unsigned int get ();
void set (unsigned int value);
}
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation

DatabaseEnvironment...:::RepHeartbeatMonitor Property

**DatabaseEnvironment Class**  **See Also**

The amount of time the replication manager, running at a client site, waits for some message activity on the connection from the master (heartbeats or other messages) before concluding that the connection has been lost. When 0 (the default), no monitoring is performed.

**Namespace:** BerkeleyDB
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public uint RepHeartbeatMonitor { get; set; }

Visual Basic (Declaration)

Public Property RepHeartbeatMonitor AsUInteger

Visual C++

public:
property unsigned int RepHeartbeatMonitor {
    unsigned int get ();
    void set (unsigned int value);
}
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
DatabaseEnvironment...:..RepHeartbeatSend Property

DatabaseEnvironment Class  See Also

The frequency at which the replication manager, running at a master site, broadcasts a heartbeat message in an otherwise idle system. When 0 (the default), no heartbeat messages will be sent.

Namespace:  BerkeleyDB
Syntax

C#

public uint RepHeartbeatSend { get; set; }

Visual Basic (Declaration)

Public Property RepHeartbeatSend As UInteger

Visual C++

public:
property unsigned int RepHeartbeatSend {
    unsigned int get ();
    void set (unsigned int value);
}
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation
DatabaseEnvironment...::RepLeaseTimeout Property

**DatabaseEnvironment Class  See Also**

The amount of time a client grants its master lease to a master. When using master leases all sites in a replication group must use the same lease timeout value. There is no default value. If leases are desired, this method must be called prior to calling `RepStartClient()` or `RepStartMaster()`.

**Namespace:**  BerkeleyDB
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public uint Rep Lease Timeout { get; set; }

Visual Basic (Declaration)

Public Property Rep Lease Timeout AsUInteger

Visual C++

public:
property unsigned int Rep Lease Timeout {
    unsigned int get ();
    void set (unsigned int value);
}
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Specify how master and client sites will handle acknowledgment of replication messages which are necessary for "permanent" records. The current implementation requires all sites in a replication group configure the same acknowledgement policy.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public AckPolicy RepMgrAckPolicy { get; set; }

Visual Basic (Declaration)

Public Property RepMgrAckPolicy As AckPolicy

Visual C++

public:
property AckPolicy^ RepMgrAckPolicy {
    AckPolicy^ get ();
    void set (AckPolicy^ value);
}
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace
DatabaseEnvironment::<RepAckTimeout

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The host information for the local system.

Namespace: BerkeleyDB
Syntax

C#

public ReplicationHostAddress RepMgrLocalSite { set; }

Visual Basic (Declaration)

Public WriteOnly Property RepMgrLocalSite As ReplicationHostAddress

Visual C++

public:
property ReplicationHostAddress^ RepMgrLocalSite {
    void set (ReplicationHostAddress^ value);
}
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The status of the sites currently known by the replication manager.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public RepMgrSite[] RepMgrRemoteSites { get; }

Visual Basic (Declaration)

Public ReadOnly Property RepMgrRemoteSites As RepMgrSite()

Visual C++

public:
property array<RepMgrSite>^ RepMgrRemoteSites {
    array<RepMgrSite>^ get ();
}
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, the replication master will not automatically re-initialize outdated clients (defaults to false).

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool RepNoAutoInit { get; set; }

Visual Basic (Declaration)

Public Property RepNoAutoInit As Boolean

Visual C++

public:
property bool RepNoAutoInit {
    bool get ();
    void set (bool value);
}
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, Berkeley DB method calls that would normally block while clients are in recovery will return errors immediately (defaults to false).

**Namespace:** BerkeleyDB
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool RepNoBlocking { get; set; }

Visual Basic (Declaration)

Public Property RepNoBlocking As Boolean

Visual C++

public:
property bool RepNoBlocking {
    bool get ();
    void set (bool value);
}
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The total number of sites in the replication group.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
### Syntax

**C#**

public uint RepNSites { get; set; }

**Visual Basic (Declaration)**

Public Property RepNSites AsUInteger

**Visual C++**

public:

property unsigned int RepNSites {
    unsigned int get();
    void set (unsigned int value);
}
Remarks

This setting is typically used by applications which use the Berkeley DB library "replication manager" support. (However, see also RepHoldElection(), the description of the nsites parameter.)
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
RepPriority Property

The database environment's priority in replication group elections. A special value of 0 indicates that this environment cannot be a replication group master. If not configured, then a default value of 100 is used.

Namespace: BerkeleyDB
Syntax

C#

public uint RepPriority { get; set; }

Visual Basic (Declaration)

Public Property RepPriority AsUInteger

Visual C++

public:
property unsigned_int RepPriority {
    unsigned_int get ();
    void set (unsigned_int value);
}
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
DatabaseEnvironment...:::RepRetransmissionRequestMax Property

DatabaseEnvironment Class  See Also

The maximum number of microseconds a client waits before requesting retransmission.

**Namespace:** BerkshireDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public uint RepRetransmissionRequestMax { get; }

Visual Basic (Declaration)

Public ReadOnly Property RepRetransmissionRequestMax AsUInteger

Visual C++

public:
property unsigned int RepRetransmissionRequestMax {
    unsigned int get ();
}
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
RepRetransmissionRequestMin Property

The minimum number of microseconds a client waits before requesting retransmission.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public uint RepRetransmissionRequestMin { get; }

Visual Basic (Declaration)

Public ReadOnly Property RepRetransmissionRequestMin As UInteger

Visual C++

public:
property unsigned int RepRetransmissionRequestMin {
    unsigned int get ();
}
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Replication Manager observes the strict "majority" rule in managing elections, even in a group with only 2 sites. This means the client in a 2-site group will be unable to take over as master if the original master fails or becomes disconnected. (See the Elections section in the Berkeley DB Reference Guide for more information.) Both sites in the replication group should have the same value for this parameter.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool RepStrict2Site { get; set; }

Visual Basic (Declaration)

Public Property RepStrict2Site As Boolean

Visual C++

public:
property bool RepStrict2Site {
    bool get ();
    void set (bool value);
}
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The bytes component of the byte-count limit on the amount of data that will be transmitted from a site in response to a single message processed by `RepProcessMessage(DatabaseEntry, DatabaseEntry, Int32)`.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public uint RepTransmitLimitBytes { get; }

Visual Basic (Declaration)

Public ReadOnly Property RepTransmitLimitBytes As UInteger

Visual C++

public:
property unsigned int RepTransmitLimitBytes {
    unsigned int get ();
}

See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
DatabaseEnvironment...:::RepTransmitLimitGBytes Property

The gigabytes component of the byte-count limit on the amount of data that will be transmitted from a site in response to a single message processed by RepProcessMessage(DatabaseEntry, DatabaseEntry, Int32).

Namespace: BerkeleyDB
Syntax

C#

public uint RepTransmitLimitGBytes { get; }

Visual Basic (Declaration)

Public ReadOnly Property RepTransmitLimitGBytes AsUInteger

Visual C++

public:
property unsigned int RepTransmitLimitGBytes {
unsigned int get ();
}

See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation
DatabaseEnvironment..::.RepTransport Property

DatabaseEnvironment Class  See Also

The delegate used to transmit data using the replication application's communication infrastructure.

**Namespace:**  BerkeleyDB  
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public ReplicationTransportDelegate RepTransport { get; }

Visual Basic (Declaration)

Public ReadOnly Property RepTransport As ReplicationTransportDelegate

Visual C++

public:
property ReplicationTransportDelegate^ RepTransport {
ReplicationTransportDelegate^ get ();
}

See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, master leases will be used for this site (defaults to false).

**Namespace:**  [BerkeleyDB](#)

**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool RepUseMasterLeases { get; set; }

Visual Basic (Declaration)

Public Property RepUseMasterLeases As Boolean

Visual C++

public:
property bool RepUseMasterLeases {
    bool get ();
    void set (bool value);
}
Remarks

Configuring this option may result in a LeaseExpiredException when attempting to read entries from a database after the site's master lease has expired.
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, catastrophic recovery was run on this environment before opening it for normal use.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool RunFatalRecovery { get; }

Visual Basic (Declaration)

Public ReadOnly Property RunFatalRecovery As Boolean

Visual C++

public:
property bool RunFatalRecovery {
    bool get ();
}

See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, normal recovery was run on this environment before opening it for normal use.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

**C#**

```csharp
public bool RunRecovery { get; }
```

**Visual Basic (Declaration)**

```vbnet
Public ReadOnly Property RunRecovery As Boolean
```

**Visual C++**

```c++
public:
    property bool RunRecovery {
        bool get();
    }
```
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
DatabaseEnvironment.SequentialWritePause Property

The number of microseconds the thread of control will pause before scheduling further write operations.

Namespace: BerkeleyDB
**Syntax**

**C#**

public uint SequentialWritePause { get; }

**Visual Basic (Declaration)**

Public ReadOnly Property SequentialWritePause As UInteger

**Visual C++**

public:
property unsigned int SequentialWritePause {
    unsigned int get ();
}

See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
A delegate that returns a unique identifier pair for the current thread of control.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

```csharp
public SetThreadIDDelegate SetThreadID { get; set; }
```

Visual Basic (Declaration)

```vbnet
Public Property SetThreadID As SetThreadIDDelegate
```

Visual C++

```cpp
public:
   property SetThreadIDDelegate^ SetThreadID {   
   SetThreadIDDelegate^ get ();   
   void set (SetThreadIDDelegate^ value);
   }
```
Remarks

This delegate supports `FailCheck()`. For more information, see Architecting Data Store and Concurrent Data Store applications, and Architecting Transactional Data Store applications, both in the Berkeley DB Programmer's Reference Guide.
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
A delegate that formats a process ID and thread ID identifier pair.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
**Syntax**

**C#**

public SetThreadNameDelegate SetThreadName { get; set; }

**Visual Basic (Declaration)**

Public Property SetThreadName As SetThreadNameDelegate

**Visual C++**

public:
property SetThreadNameDelegate^ SetThreadName {
    SetThreadNameDelegate^ get ();
    void set (SetThreadNameDelegate^ value);
}
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, allocate region memory from system shared memory instead of from heap memory or memory backed by the filesystem.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool SystemMemory { get; }

Visual Basic (Declaration)

Public ReadOnly Property SystemMemory As Boolean

Visual C++

public:
property bool SystemMemory {
    bool get ();
}

See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The path of a directory to be used as the location of temporary files.

**Namespace:**  [BerkeleyDB](https://www.berkeleydb.com)

**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

```csharp
public string TempDir { get; set; }
```

Visual Basic (Declaration)

```vbnet
Public Property TempDir As String
```

Visual C++

```cpp
public:
property String^ TempDir {
    String^ get ();
    void set (String^ value);
}
```
Remarks

The files created to back in-memory access method databases will be created relative to this path. These temporary files can be quite large, depending on the size of the database.

If no directories are specified, the following alternatives are checked in the specified order. The first existing directory path is used for all temporary files.

1. The value of the environment variable TMPDIR.
2. The value of the environment variable TEMP.
3. The value of the environment variable TMP.
4. The value of the environment variable TempFolder.
5. The value returned by the GetTempPath interface.
6. The directory /var/tmp.
7. The directory /usr/tmp.
8. The directory /temp.
9. The directory /tmp.
10. The directory C:/temp.
11. The directory C:/tmp.

Environment variables are only checked if UseEnvironmentVars is true.
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
An approximate number of threads in the database environment.

Namespace:  BerkeleyDB
Syntax

C#

public uint ThreadCount { get; private set; }

Visual Basic (Declaration)

Public Property ThreadCount AsUInteger

Visual C++

public:
property unsigned int ThreadCount {
    unsigned int get ();
    void set (unsigned int value);
}
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
A delegate that returns if a thread of control (either a true thread or a process) is still running.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
C#

public ThreadIsAliveDelegate ThreadIsAlive { get; set; }

Visual Basic (Declaration)

Public Property ThreadIsAlive As ThreadIsAliveDelegate

Visual C++

public:
property ThreadIsAliveDelegate^ ThreadIsAlive {
    ThreadIsAliveDelegate^ get (){
    void set (ThreadIsAliveDelegate^ value);
}
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, database calls timing out based on lock or transaction timeout values will throw **LockNotGrantedException** instead of **DeadlockException**.

**Namespace:**  BerkeleyDB
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool TimeNotGranted { get; set; }

Visual Basic (Declaration)

Public Property TimeNotGranted As Boolean

Visual C++

public:
property bool TimeNotGranted {
    bool get ();
    void set (bool value);
}
Remarks

If true, this allows applications to distinguish between operations which have deadlocked and operations which have exceeded their time limits.
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, Berkeley DB will not write or synchronously flush the log on transaction commit.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool TxnNoSync { get; set; }

Visual Basic (Declaration)

Public Property TxnNoSync As Boolean

Visual C++

public:
property bool TxnNoSync {
    bool get();
    void set (bool value);
}
Remarks

This means that transactions exhibit the ACI (atomicity, consistency, and isolation) properties, but not D (durability); that is, database integrity will be maintained, but if the application or system fails, it is possible some number of the most recently committed transactions may be undone during recovery. The number of transactions at risk is governed by how many log updates can fit into the log buffer, how often the operating system flushes dirty buffers to disk, and how often the log is checkpointed.
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation

DatabaseEnvironment...:::..TxnNoWait Property

DatabaseEnvironment Class  See Also

If true and a lock is unavailable for any Berkeley DB operation performed in the context of a transaction, cause the operation to throw DeadlockException (or LockNotGrantedException if configured with TimeNotGranted).

Namespace: BerkeleyDB
Syntax

**C#**

```csharp
public bool TxnNoWait { get; set; }
```

**Visual Basic (Declaration)**

```vbnet
Public Property TxnNoWait As Boolean
```

**Visual C++**

```cpp
public:
property bool TxnNoWait {
    bool get ();
    void set (bool value);
}
```
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation

DatabaseEnvironment...:::TxnSnapshot Property

DatabaseEnvironment Class  See Also

If true, all transactions in the environment will be started as if Snapshot was passed to BeginTransaction(), and all non-transactional cursors will be opened as if SnapshotIsolation was passed to Cursor().

Namespace:  BerkeleyDB
Syntax

C#

public bool TxnSnapshot { get; set; }

Visual Basic (Declaration)

Public Property TxnSnapshot As Boolean

Visual C++

public:
property bool TxnSnapshot {
    bool get ();
    void set (bool value);
}
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
A value, in microseconds, representing transaction timeouts.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

```csharp
public uint TxnTimeout { get; set; }
```

Visual Basic (Declaration)

```vbnet
Public Property TxnTimeout As UInteger
```

Visual C++

```cpp
public:
    property unsigned int TxnTimeout {
        unsigned int get ()
        void set (unsigned int value);
    }
```
Remarks

All timeouts are checked whenever a thread of control blocks on a lock or when deadlock detection is performed. As timeouts are only checked when the lock request first blocks or when deadlock detection is performed, the accuracy of the timeout depends on how often deadlock detection is performed.

Timeout values specified for the database environment may be overridden on a per-transaction basis, see SetTxnTimeout(UInt32).
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The recovery timestamp

**Namespace:**  [BerkeleyDB](https://www.oracle.com/technetwork/database/database-technologies/berkeley-db-net/overview.html)

**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

```csharp
public DateTime TxnTimestamp { get; private set; }
```

Visual Basic (Declaration)

```vbnet
Public Property TxnTimestamp As DateTime
```

Visual C++

```cpp
public:
property DateTime TxnTimestamp {
    DateTime get ();
    void set (DateTime value);
}
```
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, Berkeley DB will write, but will not synchronously flush, the log on transaction commit.

**Namespace:**  BerkeleyDB  
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool TtxnWriteNoSync { get; set; }

Visual Basic (Declaration)

Public Property TtxnWriteNoSync As Boolean

Visual C++

public:
    property bool TtxnWriteNoSync {
        bool get ();
        void set (bool value);
    }
Remarks

This means that transactions exhibit the ACI (atomicity, consistency, and isolation) properties, but not D (durability); that is, database integrity will be maintained, but if the system fails, it is possible some number of the most recently committed transactions may be undone during recovery. The number of transactions at risk is governed by how often the system flushes dirty buffers to disk and how often the log is checkpointed.
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation

DatabaseEnvironment...::UseEnvironmentVars Property

**DatabaseEnvironment Class**  [See Also](#)

The Berkeley DB process' environment may be permitted to specify information to be used when naming files; see Berkeley DB File Naming in the Programmer's Reference Guide for more information.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool UseEnvironmentVars { get; }

Visual Basic (Declaration)

Public ReadOnly Property UseEnvironmentVars As Boolean

Visual C++

public:
property bool UseEnvironmentVars {
    bool get ();
}

See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, all databases in the environment will be opened as if `UseMVCC` was set.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) **Version:** 4.8.24.0
Syntax

C#

public bool UseMVCC { get; set; }

Visual Basic (Declaration)

Public Property UseMVCC As Boolean

Visual C++

public:
property bool UseMVCC {
    bool get ();
    void set (bool value);
}
Remarks

This flag will be ignored for queue databases for which MVCC is not supported.
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, locking for the Berkeley DB Concurrent Data Store product was initialized.

**Namespace:** BerkeleyDB
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool UsingCDB { get; }

Visual Basic (Declaration)

Public ReadOnly Property UsingCDB As Boolean

Visual C++

public:
property bool UsingCDB {
    bool get ();
}

See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, the locking subsystem was initialized.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool UsingLocking { get; }

Visual Basic (Declaration)

PublicReadOnly Property UsingLocking As Boolean

Visual C++

public:
    property bool UsingLocking {
        bool get();
    }
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, the logging subsystem was initialized.

**Namespace:**  [BerkeleyDB](https://docs.oracle.com/cd/E196110_1/820-1908/bde01800n.html)

**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

```csharp
public bool UsingLogging { get; }
```

Visual Basic (Declaration)

```vbnet
Public ReadOnly Property UsingLogging As Boolean
```

Visual C++

```cpp
public:
property bool UsingLogging {
    bool get ();
}
```
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, the shared memory buffer pool subsystem was initialized.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool UsingMPool { get; }

Visual Basic (Declaration)

Public ReadOnly Property UsingMPool As Boolean

Visual C++

public:
    property bool UsingMPool {
        bool get ();
    }
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, the replication subsystem was initialized.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool UsingReplication { get; }

Visual Basic (Declaration)

Public ReadOnly Property UsingReplication As Boolean

Visual C++

public:
property bool UsingReplication {
    bool get ();
}

See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, the transaction subsystem was initialized.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool UsingTxns { get; }

Visual Basic (Declaration)

Public ReadOnly Property UsingTxns As Boolean

Visual C++

public:
property bool UsingTxns {
    bool get ();
}

See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Specific additional informational and debugging messages in the Berkeley DB message output.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public VerboseMessages Verbosity { get; set; }

Visual Basic (Declaration)

Public Property Verbosity As VerboseMessages

Visual C++

public:
property VerboseMessages^ Verbosity {
    VerboseMessages^ get ();
    void set (VerboseMessages^ value);
}
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, Berkeley DB will yield the processor immediately after each page or mutex acquisition.

Namespace:  BerkeleyDB
Syntax

C#

public bool YieldCPU { get; set; }

Visual Basic (Declaration)

Public Property YieldCPU As Boolean

Visual C++

public:
    property bool YieldCPU {
        bool get ();
        void set (bool value);
    }

Remarks

This functionality should never be used for purposes other than stress testing.
See Also

DatabaseEnvironment Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
A class representing configuration parameters for DatabaseEnvironment

**Namespace:**  BerkeleyDB
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll)  Version: 4.8.24.0
Syntax

**C#**

public class DatabaseEnvironmentConfig

**Visual Basic (Declaration)**

Public Class DatabaseEnvironmentConfig

**Visual C++**

public ref class DatabaseEnvironmentConfig
Inheritance Hierarchy

System..:::.Object
BerkeleyDB..:::.DatabaseEnvironmentConfig
See Also

DatabaseEnvironmentConfig Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The **DatabaseEnvironmentConfig** type exposes the following members.
# Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>DatabaseEnvironmentConfig</code></td>
<td>Create a new object, with default settings</td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <code>Object</code> is equal to the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <code>Type</code> of the current instance. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>SetEncryption</strong></td>
<td>Set the password and algorithm used by the Berkeley DB library to perform encryption and decryption.</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a <code>String</code> that represents the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>AutoCommit</td>
<td>If true, database operations for which no explicit transaction handle was specified, and which modify databases in the database environment, will be automatically enclosed within a transaction. If true, Berkeley DB Concurrent Data Store applications will perform locking on an environment-wide basis rather than on a per-database basis.</td>
</tr>
<tr>
<td>CDB_ALLDB</td>
<td>If true, Berkeley DB subsystems will create any underlying files, as necessary. The path of a directory to be used as the location to create the access method database files. When <code>Open(String, BTreeDatabaseConfig)</code>, <code>Open(String, HashDatabaseConfig)</code>, <code>Open(String, QueueDatabaseConfig)</code> or <code>Open(String, RecnoDatabaseConfig)</code> is used to create a file it will be created relative to this path.</td>
</tr>
<tr>
<td>CreationDir</td>
<td>Paths of directories to be used as the location of the access method database files.</td>
</tr>
<tr>
<td>ErrorFeedback</td>
<td>The mechanism for reporting detailed error messages to the application.</td>
</tr>
<tr>
<td>ErrorPrefix</td>
<td>The prefix string that appears before error messages issued by Berkeley DB.</td>
</tr>
<tr>
<td>EventNotify</td>
<td>A delegate which is called to notify the process of specific Berkeley DB events.</td>
</tr>
<tr>
<td>Feedback</td>
<td>Monitor progress within long running operations. If true, Berkeley DB will flush database writes to the backing disk before returning from the write system call, rather than flushing database writes explicitly in a separate system call, as necessary.</td>
</tr>
<tr>
<td>ForceFlush</td>
<td></td>
</tr>
<tr>
<td>Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------</td>
</tr>
<tr>
<td><strong>FreeThreaded</strong></td>
<td>If true, the created <code>DatabaseEnvironment</code> object will be free-threaded; that is, concurrently usable by multiple threads in the address space. If true, Berkeley DB will page-fault shared regions into memory when initially creating or joining a Berkeley DB environment. In addition, Berkeley DB will write the shared regions when creating an environment, forcing the underlying virtual memory and filesystems to instantiate both the necessary memory and the necessary disk space. This can also avoid out-of-disk space failures later on.</td>
</tr>
<tr>
<td><strong>InitRegions</strong></td>
<td>The permissions for any intermediate directories created by Berkeley DB.</td>
</tr>
<tr>
<td><strong>IntermediateDirMode</strong></td>
<td>If true, lock shared Berkeley DB environment files and memory-mapped databases into memory.</td>
</tr>
<tr>
<td><strong>Lockdown</strong></td>
<td>Configuration for the locking subsystem</td>
</tr>
<tr>
<td><strong>LockSystemCfg</strong></td>
<td>Configuration for the logging subsystem</td>
</tr>
<tr>
<td><strong>LogSystemCfg</strong></td>
<td>Configuration for the memory pool subsystem</td>
</tr>
<tr>
<td><strong>MPoolSystemCfg</strong></td>
<td>Configuration for the mutex subsystem</td>
</tr>
<tr>
<td><strong>MutexSystemCfg</strong></td>
<td>If true, turn off system buffering of Berkeley DB database files to avoid double caching. If true, Berkeley DB will grant all requested mutual exclusion mutexes and database locks without regard for their actual availability. This functionality should never be used for purposes other than debugging.</td>
</tr>
<tr>
<td><strong>NoLocking</strong></td>
<td>If true, Berkeley DB will copy read-only database files into the local cache instead of potentially mapping them into process memory (see <code>MMapSize</code> for further information). If true, Berkeley DB will ignore any panic state in the database environment. (Database environments in a panic state normally refuse all attempts to call Berkeley DB functions, throwing <code>RunRecoveryException</code>. This functionality</td>
</tr>
</tbody>
</table>
should never be used for purposes other than debugging.

- **Overwrite**: If true, overwrite files stored in encrypted formats before deleting them.
- **Private**: If true, allocate region memory from the heap instead of from memory backed by the filesystem or system shared memory.
- **Register**: If true, check to see if recovery needs to be performed before opening the database environment. (For this check to be accurate, all processes using the environment must specify it when opening the environment.)
- **RepSystemCfg**: Configuration for the replication subsystem.
- **RunFatalRecovery**: If true, catastrophic recovery will be run on this environment before opening it for normal use.
- **RunRecovery**: If true, normal recovery will be run on this environment before opening it for normal use.
- **SetThreadID**: A delegate that returns a unique identifier pair for the current thread of control.
- **SystemMemory**: If true, allocate region memory from system shared memory instead of from heap memory or memory backed by the filesystem.
- **TempDir**: The path of a directory to be used as the location of temporary files.
- **ThreadIsAlive**: A delegate that returns if a thread of control (either a true thread or a process) is still running.
- **ThreadName**: A delegate that formats a process ID and thread ID identifier pair.
- **TimeNotGranted**: If true, database calls timing out based on lock or transaction timeout values will throw `LockNotGrantedException` instead of `DeadlockException`. This allows applications to distinguish between operations which have deadlocked and operations which have exceeded their time limits.
<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>TxnNoSync</strong></td>
<td>If true, Berkeley DB will not write or synchronously flush the log on transaction commit.</td>
</tr>
<tr>
<td></td>
<td>If true and a lock is unavailable for any Berkeley DB operation performed in the context of a transaction, cause the operation to throw DeadlockException (or LockNotGrantedException if TimeNotGranted is set).</td>
</tr>
<tr>
<td><strong>TxnNoWait</strong></td>
<td>If true, all transactions in the environment will be started as if Snapshot were passed to BeginTransaction().</td>
</tr>
<tr>
<td></td>
<td>and all non-transactional cursors will be opened as if SnapshotIsolation were passed to Cursor().</td>
</tr>
<tr>
<td><strong>TxnSnapshot</strong></td>
<td>If true, Berkeley DB will write, but will not synchronously flush, the log on transaction commit.</td>
</tr>
<tr>
<td><strong>TxnWriteNoSync</strong></td>
<td>If true, initialize locking for the Berkeley DB Concurrent Data Store product.</td>
</tr>
<tr>
<td><strong>UseCDB</strong></td>
<td>If true, initialize the locking subsystem.</td>
</tr>
<tr>
<td><strong>UseEnvironmentVars</strong></td>
<td>If true, the Berkeley DB process' environment may be permitted to specify information to be used when naming files.</td>
</tr>
<tr>
<td><strong>UseLocking</strong></td>
<td>If true, initialize the locking subsystem.</td>
</tr>
<tr>
<td><strong>UseLogging</strong></td>
<td>If true, initialize the logging subsystem.</td>
</tr>
<tr>
<td><strong>UseMPool</strong></td>
<td>If true, initialize the shared memory buffer pool subsystem.</td>
</tr>
<tr>
<td><strong>UseMVCC</strong></td>
<td>If true, all databases in the environment will be opened as if UseMVCC is passed to Open(String, DatabaseConfig). This flag will be ignored for queue databases for which MVCC is not supported.</td>
</tr>
<tr>
<td><strong>UseReplication</strong></td>
<td>If true, initialize the replication subsystem.</td>
</tr>
<tr>
<td><strong>UseTxns</strong></td>
<td>If true, initialize the transaction subsystem.</td>
</tr>
<tr>
<td><strong>Verbosity</strong></td>
<td>Specific additional informational and debugging messages in the Berkeley DB message output.</td>
</tr>
<tr>
<td></td>
<td>If true, Berkeley DB will yield the processor.</td>
</tr>
</tbody>
</table>
YieldCPU immediately after each page or mutex acquisition. This functionality should never be used for purposes other than stress testing.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EncryptAlgorithm</td>
<td>The algorithm used to perform encryption and decryption.</td>
</tr>
<tr>
<td>EncryptionPassword</td>
<td>The password used to perform encryption and decryption.</td>
</tr>
<tr>
<td>LockTimeout</td>
<td>A value, in microseconds, representing lock timeouts.</td>
</tr>
<tr>
<td>MaxTransactions</td>
<td>The number of active transactions supported by the environment. This value bounds the size of the memory allocated for transactions. Child transactions are counted as active until they either commit or abort.</td>
</tr>
<tr>
<td>ThreadCount</td>
<td>An approximate number of threads in the database environment.</td>
</tr>
<tr>
<td>TxnTimeout</td>
<td>A value, in microseconds, representing transaction timeouts.</td>
</tr>
<tr>
<td>TxnTimestamp</td>
<td>Recover to the time specified by timestamp rather than to the most current possible date.</td>
</tr>
</tbody>
</table>
See Also

DatabaseEnvironmentConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Create a new object, with default settings

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public DatabaseEnvironmentConfig()  

Visual Basic (Declaration)

Public Sub New

Visual C++

public:  
DatabaseEnvironmentConfig()
See Also

DatabaseEnvironmentConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `DatabaseEnvironmentConfig` type exposes the following members.
### Fields

<table>
<thead>
<tr>
<th><strong>Name</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AutoCommit</strong></td>
<td>If true, database operations for which no explicit transaction handle was specified, and which modify databases in the database environment, will be automatically enclosed within a transaction. If true, Berkeley DB Concurrent Data Store applications will perform locking on an environment-wide basis rather than on a per-database basis.</td>
</tr>
<tr>
<td><strong>CDB_ALLDB</strong></td>
<td>If true, Berkeley DB subsystems will create any underlying files, as necessary. The path of a directory to be used as the location to create the access method database files. When <code>Open(String, BTreeDatabaseConfig)</code>, <code>Open(String, HashDatabaseConfig)</code>, <code>Open(String, QueueDatabaseConfig)</code> or <code>Open(String, RecnoDatabaseConfig)</code> is used to create a file it will be created relative to this path.</td>
</tr>
<tr>
<td><strong>Create</strong></td>
<td>The path of a directory to be used as the location to create the access method database files. When <code>Open(String, BTreeDatabaseConfig)</code>, <code>Open(String, HashDatabaseConfig)</code>, <code>Open(String, QueueDatabaseConfig)</code> or <code>Open(String, RecnoDatabaseConfig)</code> is used to create a file it will be created relative to this path.</td>
</tr>
<tr>
<td><strong>CreationDir</strong></td>
<td>Paths of directories to be used as the location of the access method database files.</td>
</tr>
<tr>
<td><strong>DataDirs</strong></td>
<td>The mechanism for reporting detailed error messages to the application. The prefix string that appears before error messages issued by Berkeley DB. A delegate which is called to notify the process of specific Berkeley DB events. Monitor progress within long running operations. If true, Berkeley DB will flush database writes to the backing disk before returning from the write system call, rather than flushing database writes explicitly in a separate system call, as necessary.</td>
</tr>
<tr>
<td><strong>ErrorFeedback</strong></td>
<td></td>
</tr>
<tr>
<td><strong>ErrorPrefix</strong></td>
<td></td>
</tr>
<tr>
<td><strong>EventNotify</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Feedback</strong></td>
<td></td>
</tr>
<tr>
<td><strong>ForceFlush</strong></td>
<td></td>
</tr>
<tr>
<td><strong>FreeThreaded</strong></td>
<td>If true, the created DatabaseEnvironment object will be free-threaded; that is, concurrently usable by multiple threads in the address space. If true, Berkeley DB will page-fault shared regions into memory when initially creating or joining a Berkeley DB environment. In addition, Berkeley DB will write the shared regions when creating an environment, forcing the underlying virtual memory and filesystems to instantiate both the necessary memory and the necessary disk space. This can also avoid out-of-disk space failures later on.</td>
</tr>
<tr>
<td><strong>InitRegions</strong></td>
<td>The permissions for any intermediate directories created by Berkeley DB.</td>
</tr>
<tr>
<td><strong>IntermediateDirMode</strong></td>
<td>If true, lock shared Berkeley DB environment files and memory-mapped databases into memory.</td>
</tr>
<tr>
<td><strong>Lockdown</strong></td>
<td>Configuration for the locking subsystem</td>
</tr>
<tr>
<td><strong>LockSystemCfg</strong></td>
<td>Configuration for the logging subsystem</td>
</tr>
<tr>
<td><strong>LogSystemCfg</strong></td>
<td>Configuration for the memory pool subsystem</td>
</tr>
<tr>
<td><strong>MPoolSystemCfg</strong></td>
<td>Configuration for the mutex subsystem</td>
</tr>
<tr>
<td><strong>MutexSystemCfg</strong></td>
<td>If true, turn off system buffering of Berkeley DB database files to avoid double caching. If true, Berkeley DB will grant all requested mutual exclusion mutexes and database locks without regard for their actual availability. This functionality should never be used for purposes other than debugging.</td>
</tr>
<tr>
<td><strong>NoBuffer</strong></td>
<td>If true, Berkeley DB will copy read-only database files into the local cache instead of potentially mapping them into process memory (see MMapSize for further information).</td>
</tr>
<tr>
<td><strong>NoLocking</strong></td>
<td>If true, Berkeley DB will ignore any panic state in the database environment. (Database environments in a panic state normally refuse all attempts to call Berkeley DB functions, throwing RunRecoveryException. This functionality</td>
</tr>
</tbody>
</table>
The document contains a list of settings and descriptions. Here is the plain text representation:

- **Overwrite**: If true, overwrite files stored in encrypted formats before deleting them.
- **Private**: If true, allocate region memory from the heap instead of from memory backed by the filesystem or system shared memory.
- **Register**: If true, check to see if recovery needs to be performed before opening the database environment. (For this check to be accurate, all processes using the environment must specify it when opening the environment.)
- **RepSystemCfg**: Configuration for the replication subsystem.
- **RunFatalRecovery**: If true, catastrophic recovery will be run on this environment before opening it for normal use.
- **RunRecovery**: If true, normal recovery will be run on this environment before opening it for normal use.
- **SetThreadID**: A delegate that returns a unique identifier pair for the current thread of control.
- **SystemMemory**: If true, allocate region memory from system shared memory instead of from heap memory or memory backed by the filesystem.
- **TempDir**: The path of a directory to be used as the location of temporary files.
- **ThreadIsAlive**: A delegate that returns if a thread of control (either a true thread or a process) is still running.
- **ThreadName**: A delegate that formats a process ID and thread ID identifier pair.
- **TimeNotGranted**: If true, database calls timing out based on lock or transaction timeout values will throw **LockNotGrantedException** instead of **DeadlockException**. This allows applications to distinguish between operations which have deadlocked and operations which have exceeded their time limits.
<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>TxnNoSync</strong></td>
<td>If true, Berkeley DB will not write or synchronously flush the log on transaction commit.</td>
</tr>
<tr>
<td><strong>TxnNoWait</strong></td>
<td>If true and a lock is unavailable for any Berkeley DB operation performed in the context of a transaction, cause the operation to throw DeadlockException (or LockNotGrantedException if TimeNotGranted is set).</td>
</tr>
<tr>
<td><strong>TxnSnapshot</strong></td>
<td>If true, all transactions in the environment will be started as if Snapshot were passed to BeginTransaction(), and all non-transactional cursors will be opened as if SnapshotIsolation were passed to Cursor().</td>
</tr>
<tr>
<td><strong>TxnWriteNoSync</strong></td>
<td>If true, Berkeley DB will write, but will not synchronously flush, the log on transaction commit.</td>
</tr>
<tr>
<td><strong>UseCDB</strong></td>
<td>If true, initialize locking for the Berkeley DB Concurrent Data Store product.</td>
</tr>
<tr>
<td><strong>UseEnvironmentVars</strong></td>
<td>If true, the Berkeley DB process' environment may be permitted to specify information to be used when naming files.</td>
</tr>
<tr>
<td><strong>UseLocking</strong></td>
<td>If true, initialize the locking subsystem.</td>
</tr>
<tr>
<td><strong>UseLogging</strong></td>
<td>If true, initialize the logging subsystem.</td>
</tr>
<tr>
<td><strong>UseMPool</strong></td>
<td>If true, initialize the shared memory buffer pool subsystem.</td>
</tr>
<tr>
<td><strong>UseMVCC</strong></td>
<td>If true, all databases in the environment will be opened as if UseMVCC is passed to Open(String, DatabaseConfig). This flag will be ignored for queue databases for which MVCC is not supported.</td>
</tr>
<tr>
<td><strong>UseReplication</strong></td>
<td>If true, initialize the replication subsystem.</td>
</tr>
<tr>
<td><strong>UseTxns</strong></td>
<td>If true, initialize the transaction subsystem.</td>
</tr>
<tr>
<td><strong>Verbosity</strong></td>
<td>Specific additional informational and debugging messages in the Berkeley DB message output.</td>
</tr>
</tbody>
</table>
immediately after each page or mutex acquisition. This functionality should never be used for purposes other than stress testing.
See Also

DatabaseEnvironmentConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, database operations for which no explicit transaction handle was specified, and which modify databases in the database environment, will be automatically enclosed within a transaction.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool AutoCommit

Visual Basic (Declaration)

Public AutoCommit As Boolean

Visual C++

public:
bool AutoCommit
See Also

DatabaseEnvironmentConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, Berkeley DB Concurrent Data Store applications will perform locking on an environment-wide basis rather than on a per-database basis.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool CDB_ALLDB

Visual Basic (Declaration)

Public CDB_ALLDB As Boolean

Visual C++

public:
    bool CDB_ALLDB
See Also

DatabaseEnvironmentConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, Berkeley DB subsystems will create any underlying files, as necessary.

**Namespace:** [BerkeleyDB](https://docs.oracle.com/cd/E192000_01/jfhe/jfhe.html)

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
### Syntax

**C#**

```csharp
public bool Create
```

**Visual Basic (Declaration)**

```vbnet
Public Create As Boolean
```

**Visual C++**

```cpp
public:
bool Create
```
See Also

DatabaseEnvironmentConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
DatabaseEnvironmentConfig::CreationDir Field

The path of a directory to be used as the location to create the access method database files. When `Open(String, BTreeDatabaseConfig)`, `Open(String, HashDatabaseConfig)`, `Open(String, QueueDatabaseConfig)` or `Open(String, RecnoDatabaseConfig)` is used to create a file it will be created relative to this path.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public string CreationDir

Visual Basic (Declaration)

Public CreationDir As String

Visual C++

public:
String^ CreationDir
Remarks

This path must also exist in DataDirs.

If no database directory is specified, database files must be named either by absolute paths or relative to the environment home directory. See Berkeley DB File Naming in the Programmer's Reference Guide for more information.
See Also

DatabaseEnvironmentConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Paths of directories to be used as the location of the access method database files.

**Namespace:**  BerkeleyDB

**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public List<string> DataDirs

Visual Basic (Declaration)

Public DataDirs As List(Of String)

Visual C++

public: List<String>^ DataDirs
Remarks

Paths specified to `Open(String, DatabaseConfig)` will be searched relative to this path. Paths set using this method are additive, and specifying more than one will result in each specified directory being searched for database files.

If no database directories are specified, database files must be named either by absolute paths or relative to the environment home directory. See Berkeley DB File Naming in the Programmer's Reference Guide for more information.
See Also

DatabaseEnvironmentConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The mechanism for reporting detailed error messages to the application.

**Namespace:**  BerkeleyDB  
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

**C#**

public `ErrorFeedbackDelegate` ErrorFeedback

**Visual Basic (Declaration)**

Public ErrorFeedback As `ErrorFeedbackDelegate`

**Visual C++**

public:
`ErrorFeedbackDelegate`^ ErrorFeedback
Remarks

When an error occurs in the Berkeley DB library, a `DatabaseException`, or subclass of `DatabaseException`, is thrown. In some cases, however, the exception may be insufficient to completely describe the cause of the error, especially during initial application debugging.

In some cases, when an error occurs, Berkeley DB will call the given delegate with additional error information. It is up to the delegate to display the error message in an appropriate manner.

Setting ErrorFeedback to NULL unconfigures the callback interface.

This error-logging enhancement does not slow performance or significantly increase application size, and may be run during normal operation as well as during application debugging.
See Also

DatabaseEnvironmentConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The prefix string that appears before error messages issued by Berkeley DB.

Namespace: BerkeleyDB
Syntax

C#

public string ErrorPrefix

Visual Basic (Declaration)

Public ErrorPrefix As String

Visual C++

public:
String^ ErrorPrefix
Remarks

For databases opened inside of a DatabaseEnvironment, setting ErrorPrefix affects the entire environment and is equivalent to setting ErrorPrefix.
See Also

DatabaseEnvironmentConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
DatabaseEnvironmentConfig::EventNotify Field

A delegate which is called to notify the process of specific Berkeley DB events.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public EventNotifyDelegate EventNotify

Visual Basic (Declaration)

Public EventNotify As EventNotifyDelegate

Visual C++

public:
EventNotifyDelegate^ EventNotify
See Also

DatabaseEnvironmentConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Monitor progress within long running operations.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public EnvironmentFeedbackDelegate Feedback

Visual Basic (Declaration)

Public Feedback As EnvironmentFeedbackDelegate

Visual C++

public: EnvironmentFeedbackDelegate^ Feedback
Remarks

Some operations performed by the Berkeley DB library can take non-trivial amounts of time. The Feedback delegate can be used by applications to monitor progress within these operations. When an operation is likely to take a long time, Berkeley DB will call the specified delegate with progress information.

It is up to the delegate to display this information in an appropriate manner.
See Also

DatabaseEnvironmentConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, Berkeley DB will flush database writes to the backing disk before returning from the write system call, rather than flushing database writes explicitly in a separate system call, as necessary.

**Namespace:**  BerkeleyDB  
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool ForceFlush

Visual Basic (Declaration)

Public ForceFlush As Boolean

Visual C++

public:
bool ForceFlush
Remarks

This is only available on some systems (for example, systems supporting the IEEE/ANSI Std 1003.1 (POSIX) standard O_DSYNC flag, or systems supporting the Windows FILE_FLAG_WRITE_THROUGH flag). This flag may result in inaccurate file modification times and other file-level information for Berkeley DB database files. This flag will almost certainly result in a performance decrease on most systems. This flag is only applicable to certain filesystems (for example, the Veritas VxFS filesystem), where the filesystem's support for trickling writes back to stable storage behaves badly (or more likely, has been misconfigured).
See Also

DatabaseEnvironmentConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, the created `DatabaseEnvironment` object will be free-threaded; that is, concurrently usable by multiple threads in the address space.

**Namespace:** BerkeleyDB  
**Assembly:** `libdb_dotnet48` (in `libdb_dotnet48.dll`) Version: 4.8.24.0
Syntax

C#

public bool FreeThreaded

Visual Basic (Declaration)

Public FreeThreaded As Boolean

Visual C++

public:
bool FreeThreaded
Remarks

Required to be true if the created `DatabaseEnvironment` object will be concurrently used by more than one thread in the process, or if any `Database` objects opened in the scope of the `DatabaseEnvironment` object will be concurrently used by more than one thread in the process.

Required to be true when using the Replication Manager.
See Also

DatabaseEnvironmentConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, Berkeley DB will page-fault shared regions into memory when initially creating or joining a Berkeley DB environment. In addition, Berkeley DB will write the shared regions when creating an environment, forcing the underlying virtual memory and filesystems to instantiate both the necessary memory and the necessary disk space. This can also avoid out-of-disk space failures later on.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool InitRegions

Visual Basic (Declaration)

Public InitRegions As Boolean

Visual C++

public:
  bool InitRegions
Remarks

In some applications, the expense of page-faulting the underlying shared memory regions can affect performance. (For example, if the page-fault occurs while holding a lock, other lock requests can convoy, and overall throughput may decrease.)
See Also

DatabaseEnvironmentConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation

DatabaseEnvironmentConfig::Intermed\ IntermediateDirMode Field

DatabaseEnvironmentConfig Class  See Also

The permissions for any intermediate directories created by Berkeley DB.

Namespace: BerkeleyDB
Syntax

C#

public string IntermediateDirMode

Visual Basic (Declaration)

Public IntermediateDirMode As String

Visual C++

public:
String^ IntermediateDirMode
Remarks

By default, Berkeley DB does not create intermediate directories needed for recovery, that is, if the file /a/b/c/mydatabase is being recovered, and the directory path b/c does not exist, recovery will fail. This default behavior is because Berkeley DB does not know what permissions are appropriate for intermediate directory creation, and creating the directory might result in a security problem.

Directory permissions are interpreted as a string of nine characters, using the character set r (read), w (write), x (execute or search), and - (none). The first character is the read permissions for the directory owner (set to either r or -). The second character is the write permissions for the directory owner (set to either w or -). The third character is the execute permissions for the directory owner (set to either x or -).

Similarly, the second set of three characters are the read, write and execute/search permissions for the directory group, and the third set of three characters are the read, write and execute/search permissions for all others. For example, the string rwx------ would configure read, write and execute/search access for the owner only. The string rwxrwx--- would configure read, write and execute/search access for both the owner and the group. The string rwxr------ would configure read, write and execute/search access for the directory owner and read-only access for the directory group.
See Also

DatabaseEnvironmentConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
DatabaseEnvironmentConfig..:..Lockdown Field

If true, lock shared Berkeley DB environment files and memory-mapped databases into memory.

Namespace: BerkeleyDB
Syntax

C#

public bool Lockdown

Visual Basic (Declaration)

Public Lockdown As Boolean

Visual C++

public:
bool Lockdown
See Also

DatabaseEnvironmentConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
DatabaseEnvironmentConfig.LockSystemCfg Field

Configuration for the locking subsystem

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

**C#**

public `LockingConfig` LockSystemCfg

**Visual Basic (Declaration)**

Public LockSystemCfg As `LockingConfig`

**Visual C++**

public:

`LockingConfig`^ LockSystemCfg
See Also

DatabaseEnvironmentConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
DatabaseEnvironmentConfig...::LogSystemCfg Field

**DatabaseEnvironmentConfig Class**  **See Also**

Configuration for the logging subsystem

**Namespace:**  [BerkeleyDB](https://BerkeleyDB)

**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll)  Version:  4.8.24.0
Syntax

C#

public LogConfig LogSystemCfg

Visual Basic (Declaration)

Public LogSystemCfg As LogConfig

Visual C++

public:
LogConfig LogSystemCfg
See Also

DatabaseEnvironmentConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation

DatabaseEnvironmentConfig:::MPoolSystemCfg Field

DatabaseEnvironmentConfig Class  See Also

Configuration for the memory pool subsystem

Namespace:  BerkeleyDB
Syntax

C#

public MPoolConfig MPoolSystemCfg

Visual Basic (Declaration)

Public MPoolSystemCfg As MPoolConfig

Visual C++

public: MPoolConfig^ MPoolSystemCfg
See Also

DatabaseEnvironmentConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
DatabaseEnvironmentConfig.

::

MutexSystemCfg Field

See Also

Configuration for the mutex subsystem

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public MutexConfig MutexSystemCfg

Visual Basic (Declaration)

Public MutexSystemCfg As MutexConfig

Visual C++

public:
MutexConfig^ MutexSystemCfg
See Also

DatabaseEnvironmentConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, turn off system buffering of Berkeley DB database files to avoid double caching.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool NoBuffer

Visual Basic (Declaration)

Public NoBuffer As Boolean

Visual C++

public:
bool NoBuffer
See Also

- DatabaseEnvironmentConfig Class
- BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, Berkeley DB will grant all requested mutual exclusion mutexes and database locks without regard for their actual availability. This functionality should never be used for purposes other than debugging.

**Namespace:**  BerkeleyDB

**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool NoLocking

Visual Basic (Declaration)

Public NoLocking As Boolean

Visual C++

public:
    bool NoLocking
See Also

DatabaseEnvironmentConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, Berkeley DB will copy read-only database files into the local cache instead of potentially mapping them into process memory (see MMapSize for further information).

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool NoMMap

Visual Basic (Declaration)

Public NoMMap As Boolean

Visual C++

public:
bool NoMMap
See Also

DatabaseEnvironmentConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, Berkeley DB will ignore any panic state in the database environment. (Database environments in a panic state normally refuse all attempts to call Berkeley DB functions, throwing RunRecoveryException. This functionality should never be used for purposes other than debugging.

**Namespace:**  BerkeleyDB  
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool NoPanic

Visual Basic (Declaration)

Public NoPanic As Boolean

Visual C++

public:
bool NoPanic
See Also

DatabaseEnvironmentConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, overwrite files stored in encrypted formats before deleting them.

**Namespace:** [BerkeleyDB](https://www.oracle.com/database/berkeley-db/)  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool Overwrite

Visual Basic (Declaration)

Public Overwrite As Boolean

Visual C++

public:
bool Overwrite
Remarks

Berkeley DB overwrites files using alternating 0xff, 0x00 and 0xff byte patterns. For file overwriting to be effective, the underlying file must be stored on a fixed-block filesystem. Systems with journaling or logging filesystems will require operating system support and probably modification of the Berkeley DB sources.
See Also

DatabaseEnvironmentConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, allocate region memory from the heap instead of from memory backed by the filesystem or system shared memory.

**Namespace:**  BerkeleyDB  
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

**C#**

public bool Private

**Visual Basic (Declaration)**

Public Private As Boolean

**Visual C++**

public:
bool Private
Remarks

This setting implies the environment will only be accessed by a single process (although that process may be multithreaded). This flag has two effects on the Berkeley DB environment. First, all underlying data structures are allocated from per-process memory instead of from shared memory that is accessible to more than a single process. Second, mutexes are only configured to work between threads.

This setting should be false if more than a single process is accessing the environment because it is likely to cause database corruption and unpredictable behavior. For example, if both a server application and Berkeley DB utilities (for example, db_archive, db_checkpoint or db_stat) are expected to access the environment, this setting should be false.
See Also

DatabaseEnvironmentConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, check to see if recovery needs to be performed before opening the database environment. (For this check to be accurate, all processes using the environment must specify it when opening the environment.)

**Namespace:**  [BerkeleyDB](https://berkeleydb.com)

**Assembly:**  `libdb_dotnet48` (in `libdb_dotnet48.dll`) Version: 4.8.24.0
Syntax

C#

public bool Register

Visual Basic (Declaration)

Public Register As Boolean

Visual C++

public:
bool Register
Remarks

If recovery needs to be performed for any reason (including the initial use of this setting), and RunRecovery is also specified, recovery will be performed and the open will proceed normally. If recovery needs to be performed and RunRecovery is not specified, RunRecoveryException will be thrown. If recovery does not need to be performed, RunRecovery will be ignored. See Architecting Transactional Data Store applications in the Programmer's Reference Guide for more information.
See Also

DatabaseEnvironmentConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
DatabaseEnvironmentConfig.

...::RepSystemCfg Field

DatabaseEnvironmentConfig Class  See Also

Configuration for the replication subsystem

Namespace:  BerkeleyDB
Syntax

C#

public ReplicationConfig RepSystemCfg

Visual Basic (Declaration)

Public RepSystemCfg As ReplicationConfig

Visual C++

public: ReplicationConfig^ RepSystemCfg
See Also

DatabaseEnvironmentConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, catastrophic recovery will be run on this environment before opening it for normal use.

**Namespace:**  BerkeleyDB  
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool RunFatalRecovery

Visual Basic (Declaration)

Public RunFatalRecovery As Boolean

Visual C++

public:
bool RunFatalRecovery
Remarks

If true, the Create and UseTxns must also be set, because the regions will be removed and re-created, and transactions are required for application recovery.
See Also

DatabaseEnvironmentConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, normal recovery will be run on this environment before opening it for normal use.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool RunRecovery

Visual Basic (Declaration)

Public RunRecovery As Boolean

Visual C++

public:
bool RunRecovery
Remarks

If true, the Create and UseTxns must also be set, because the regions will be removed and re-created, and transactions are required for application recovery.
See Also

DatabaseEnvironmentConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
A delegate that returns a unique identifier pair for the current thread of control.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public SetThreadIDDelegate SetThreadID

Visual Basic (Declaration)

Public SetThreadID As SetThreadIDDelegate

Visual C++

public:
SetThreadIDDelegate^ SetThreadID
Remarks

This delegate supports `FailCheck()`. For more information, see Architecting Data Store and Concurrent Data Store applications, and Architecting Transactional Data Store applications, both in the Berkeley DB Programmer's Reference Guide.
See Also

DatabaseEnvironmentConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, allocate region memory from system shared memory instead of from heap memory or memory backed by the filesystem.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

**C#**

public bool SystemMemory

**Visual Basic (Declaration)**

Public SystemMemory As Boolean

**Visual C++**

public: bool SystemMemory
Remarks

See Also

DatabaseEnvironmentConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The path of a directory to be used as the location of temporary files.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public string TempDir

Visual Basic (Declaration)

Public TempDir As String

Visual C++

public:
String^ TempDir
Remarks

The files created to back in-memory access method databases will be created relative to this path. These temporary files can be quite large, depending on the size of the database.

If no directories are specified, the following alternatives are checked in the specified order. The first existing directory path is used for all temporary files.

1. The value of the environment variable TMPDIR.
2. The value of the environment variable TEMP.
3. The value of the environment variable TMP.
4. The value of the environment variable TempFolder.
5. The value returned by the GetTempPath interface.
6. The directory /var/tmp.
7. The directory /usr/tmp.
8. The directory /temp.
9. The directory /tmp.
10. The directory C:/temp.
11. The directory C:/tmp.

Environment variables are only checked if UseEnvironmentVars is true.
See Also

DatabaseEnvironmentConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
A delegate that returns if a thread of control (either a true thread or a process) is still running.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public ThreadIsAliveDelegate ThreadIsAlive

Visual Basic (Declaration)

Public ThreadIsAlive As System.Threading.ThreadIsAliveDelegate

Visual C++

public:
ThreadIsAliveDelegate ^ ThreadIsAlive
See Also

DatabaseEnvironmentConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
A delegate that formats a process ID and thread ID identifier pair.

**Namespace:** [BerkeleyDB](https://docs.oracle.com/cd/E17250_01/lib/4.8.24.0/)

**Assembly:** `libdb_dotnet48` (in `libdb_dotnet48.dll`) Version: 4.8.24.0
### Syntax

**C#**

```
public SetThreadNameDelegate ThreadName
```

**Visual Basic (Declaration)**

```
Public ThreadName As SetThreadNameDelegate
```

**Visual C++**

```
public:
SetThreadNameDelegate^ ThreadName
```
See Also

DatabaseEnvironmentConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, database calls timing out based on lock or transaction timeout values will throw `LockNotGrantedException` instead of `DeadlockException`. This allows applications to distinguish between operations which have deadlocked and operations which have exceeded their time limits.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool TimeNotGranted

Visual Basic (Declaration)

Public TimeNotGranted As Boolean

Visual C++

public:
bool TimeNot Granted
See Also

DatabaseEnvironmentConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, Berkeley DB will not write or synchronously flush the log on transaction commit.

**Namespace:**  BerkeleyDB

**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool TxnNoSync

Visual Basic (Declaration)

Public TxnNoSync As Boolean

Visual C++

public:
    bool TxnNoSync
Remarks

This means that transactions exhibit the ACI (atomicity, consistency, and isolation) properties, but not D (durability); that is, database integrity will be maintained, but if the application or system fails, it is possible some number of the most recently committed transactions may be undone during recovery. The number of transactions at risk is governed by how many log updates can fit into the log buffer, how often the operating system flushes dirty buffers to disk, and how often the log is checkpointed.
See Also

DatabaseEnvironmentConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true and a lock is unavailable for any Berkeley DB operation performed in the context of a transaction, cause the operation to throw `DeadlockException` (or `LockNotGrantedException` if `TimeNotGranted` is set).

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool TxnNoWait

Visual Basic (Declaration)

Public TxnNoWait As Boolean

Visual C++

public:
    bool TxnNoWait
See Also

DatabaseEnvironmentConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, all transactions in the environment will be started as if `Snapshot` were passed to `BeginTransaction()`, and all non-transactional cursors will be opened as if `SnapshotIsolation` were passed to `Cursor()`.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool TxnSnapshot

Visual Basic (Declaration)

Public TxnSnapshot As Boolean

Visual C++

public:
bool TxnSnapshot
See Also

DatabaseEnvironmentConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, Berkeley DB will write, but will not synchronously flush, the log on transaction commit.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool TxnWriteNoSync

Visual Basic (Declaration)

Public TxnWriteNoSync As Boolean

Visual C++

public:
    bool TxnWriteNoSync
Remarks

This means that transactions exhibit the ACI (atomicity, consistency, and isolation) properties, but not D (durability); that is, database integrity will be maintained, but if the system fails, it is possible some number of the most recently committed transactions may be undone during recovery. The number of transactions at risk is governed by how often the system flushes dirty buffers to disk and how often the log is checkpointed.
See Also

DatabaseEnvironmentConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, initialize locking for the Berkeley DB Concurrent Data Store product.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool UseCDB

Visual Basic (Declaration)

Public UseCDB As Boolean

Visual C++

public:
    bool UseCDB
Remarks

In this mode, Berkeley DB provides multiple reader/single writer access. The only other subsystem that should be specified with UseCDB flag is UseMPool.
See Also

DatabaseEnvironmentConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, the Berkeley DB process' environment may be permitted to specify information to be used when naming files.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

**C#**

```csharp
public bool UseEnvironmentVars
```

**Visual Basic (Declaration)**

```vbnet
Public UseEnvironmentVars As Boolean
```

**Visual C++**

```cpp
public:
bool UseEnvironmentVars
```
Remarks


Because permitting users to specify which files are used can create security problems, environment information will be used in file naming for all users only if UseEnvironmentVars is true.
See Also

DatabaseEnvironmentConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, initialize the locking subsystem.

**Namespace:** [BerkeleyDB](https://www.berdex.com)

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
**Syntax**

**C#**

public bool UseLocking

**Visual Basic (Declaration)**

Public UseLocking As Boolean

**Visual C++**

public:
    bool UseLocking
Remarks

This subsystem should be used when multiple processes or threads are going to be reading and writing a Berkeley DB database, so that they do not interfere with each other. If all threads are accessing the database(s) read-only, locking is unnecessary. When UseLocking is specified, it is usually necessary to run a deadlock detector, as well. See DetectDeadlocks(DeadlockPolicy) for more information.
See Also

DatabaseEnvironmentConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, initialize the logging subsystem.

**Namespace:**  [BerkeleyDB](#)  
**Assembly:**  `libdb_dotnet48` (in `libdb_dotnet48.dll`) Version: 4.8.24.0
Syntax

C#

public bool UseLogging

Visual Basic (Declaration)

Public UseLogging As Boolean

Visual C++

public:
bool UseLogging
Remarks

This subsystem should be used when recovery from application or system failure is necessary. If the log region is being created and log files are already present, the log files are reviewed; subsequent log writes are appended to the end of the log, rather than overwriting current log entries.
See Also

DatabaseEnvironmentConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, initialize the shared memory buffer pool subsystem.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

```csharp
public bool UseMPool
```

Visual Basic (Declaration)

```vbnet
Public UseMPool As Boolean
```

Visual C++

```c++
public:
bool UseMPool
```
Remarks

This subsystem should be used whenever an application is using any Berkeley DB access method.
See Also

DatabaseEnvironmentConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, all databases in the environment will be opened as if UseMVCC is passed to Open(String, DatabaseConfig). This flag will be ignored for queue databases for which MVCC is not supported.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool UseMVCC

Visual Basic (Declaration)

Public UseMVCC As Boolean

Visual C++

public:
bool UseMVCC
See Also

DatabaseEnvironmentConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, initialize the replication subsystem.

Namespace: BerkeleyDB
Syntax

C#

public bool UseReplication

Visual Basic (Declaration)

Public UseReplication As Boolean

Visual C++

public:
bool UseReplication
Remarks

This subsystem should be used whenever an application plans on using replication. UseReplication requires UseTxns and UseLocking also be set.
See Also

DatabaseEnvironmentConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, initialize the transaction subsystem.

Namespace: BerkeleyDB
Syntax

C#

public bool UseTxns

Visual Basic (Declaration)

Public UseTxns As Boolean

Visual C++

public:
bool UseTxns
Remarks

This subsystem should be used when recovery and atomicity of multiple operations are important. UseTxns implies UseLogging.
See Also

DatabaseEnvironmentConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Specific additional informational and debugging messages in the Berkeley DB message output.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public VerboseMessages Verbosity

Visual Basic (Declaration)

Public Verbosity As VerboseMessages

Visual C++

public:
VerboseMessages^ Verbosity
See Also

DatabaseEnvironmentConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, Berkeley DB will yield the processor immediately after each page or mutex acquisition. This functionality should never be used for purposes other than stress testing.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool YieldCPU

Visual Basic (Declaration)

Public YieldCPU As Boolean

Visual C++

public:
bool YieldCPU
See Also

DatabaseEnvironmentConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `DatabaseEnvironmentConfig` type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>SetEncryption</strong></td>
<td>Set the password and algorithm used by the Berkeley DB library to perform encryption and decryption.</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a <a href="#">String</a> that represents the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
</tbody>
</table>
See Also

DatabaseEnvironmentConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation
DatabaseEnvironmentConfig...:::SetEncryption Method

See Also
DatabaseEnvironmentConfig Class

Set the password and algorithm used by the Berkeley DB library to perform encryption and decryption.

Namespace: BerkeleyDB
Syntax

C#

public void SetEncryption(
    string password,
    EncryptionAlgorithm alg
)

Visual Basic (Declaration)

Public Sub SetEncryption (_
    password As String,
    alg As EncryptionAlgorithm _
)

Visual C++

public:
void SetEncryption(
    String^ password,
    EncryptionAlgorithm alg
)

Parameters

password
Type: System::String
The password used to perform encryption and decryption.

alg
Type: BerkeleyDB::EncryptionAlgorithm
The algorithm used to perform encryption and decryption.
See Also

DatabaseEnvironmentConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `DatabaseEnvironmentConfig` type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EncryptAlgorithm</td>
<td>The algorithm used to perform encryption and decryption.</td>
</tr>
<tr>
<td>EncryptionPassword</td>
<td>The password used to perform encryption and decryption.</td>
</tr>
<tr>
<td>LockTimeout</td>
<td>A value, in microseconds, representing lock timeouts.</td>
</tr>
<tr>
<td>MaxTransactions</td>
<td>The number of active transactions supported by the environment. This value bounds the size of the memory allocated for transactions. Child transactions are counted as active until they either commit or abort.</td>
</tr>
<tr>
<td>ThreadCount</td>
<td>An approximate number of threads in the database environment.</td>
</tr>
<tr>
<td>TxnTimeout</td>
<td>A value, in microseconds, representing transaction timeouts.</td>
</tr>
<tr>
<td>TxnTimestamp</td>
<td>Recover to the time specified by timestamp rather than to the most current possible date.</td>
</tr>
</tbody>
</table>
See Also

DatabaseEnvironmentConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The algorithm used to perform encryption and decryption.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public EncryptionAlgorithm EncryptAlgorithm { get; }

Visual Basic (Declaration)

Public ReadOnly Property EncryptAlgorithm As EncryptionAlgorithm

Visual C++

public:
    property EncryptionAlgorithm EncryptAlgorithm { EncryptionAlgorithm get (); }

See Also

DatabaseEnvironmentConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The password used to perform encryption and decryption.

**Namespace:** [BerkeleyDB](https://www.berdex.com)

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public string EncryptionPassword { get; }

Visual Basic (Declaration)

Public ReadOnly Property EncryptionPassword As String

Visual C++

public:
property String^ EncryptionPassword {
    String^ get ();
}
}
See Also

DatabaseEnvironmentConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
DatabaseEnvironmentConfig.LockTimeout Property

A value, in microseconds, representing lock timeouts.

Namespace: BerkeleyDB
Syntax

C#

public uint LockTimeout { get; set; }

Visual Basic (Declaration)

Public Property LockTimeout As UInteger

Visual C++

public:
property unsigned int LockTimeout {
    unsigned int get ();
    void set (unsigned int value);
}
Remarks

All timeouts are checked whenever a thread of control blocks on a lock or when deadlock detection is performed. As timeouts are only checked when the lock request first blocks or when deadlock detection is performed, the accuracy of the timeout depends on how often deadlock detection is performed.

Timeout values specified for the database environment may be overridden on a per-transaction basis, see SetLockTimeout(UInt32).
See Also

DatabaseEnvironmentConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The number of active transactions supported by the environment. This value bounds the size of the memory allocated for transactions. Child transactions are counted as active until they either commit or abort.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public uint MaxTransactions { get; set; }

Visual Basic (Declaration)

Public Property MaxTransactions AsUInteger

Visual C++

public:
property unsigned int MaxTransactions {
    unsigned int get();
    void set (unsigned int value);
}
Remarks

Transactions that update multiversion databases are not freed until the last page version that the transaction created is flushed from cache. This means that applications using multi-version concurrency control may need a transaction for each page in cache, in the extreme case.

When all of the memory available in the database environment for transactions is in use, calls to BeginTransaction() will fail (until some active transactions complete). If MaxTransactions is never set, the database environment is configured to support at least 100 active transactions.
See Also

DatabaseEnvironmentConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
An approximate number of threads in the database environment.

**Namespace:**  BerkeleyDB  
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
### Syntax

**C#**

```csharp
public uint ThreadCount { get; set; }
```

**Visual Basic (Declaration)**

```vbnet
Public Property ThreadCount AsUInteger
```

**Visual C++**

```cpp
public:
    property unsigned int ThreadCount {
        unsigned int get ();
        void set (unsigned int value);
    }
```
Remarks

ThreadCount must set if `FailCheck()` will be used. ThreadCount does not set the maximum number of threads but is used to determine memory sizing and the thread control block reclamation policy.

If a process has not configured `ThreadIsAlive`, and then attempts to join a database environment configured for failure checking with `FailCheck()`, `SetThreadID`, `ThreadIsAlive` and ThreadCount, the program may be unable to allocate a thread control block and fail to join the environment. This is true of the standalone Berkeley DB utility programs. To avoid problems when using the standalone Berkeley DB utility programs with environments configured for failure checking, incorporate the utility's functionality directly in the application, or call `FailCheck()` before running the utility.
See Also

DatabaseEnvironmentConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation
DatabaseEnvironmentConfig...: TxnTimeout Property

DatabaseEnvironmentConfig Class See Also

A value, in microseconds, representing transaction timeouts.

Namespace: BerkeleyDB
## Syntax

**C#**

```csharp
public uint TxnTimeout { get; set; }
```

**Visual Basic (Declaration)**

```vbnet
Public Property TxnTimeout As UInteger
```

**Visual C++**

```cpp
public:
property unsigned int TxnTimeout {
    unsigned int get ();
    void set (unsigned int value);
}
```
Remarks

All timeouts are checked whenever a thread of control blocks on a lock or when deadlock detection is performed. As timeouts are only checked when the lock request first blocks or when deadlock detection is performed, the accuracy of the timeout depends on how often deadlock detection is performed.

Timeout values specified for the database environment may be overridden on a per-transaction basis, see SetTxnTimeout(UInt32).
See Also

DatabaseEnvironmentConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Recover to the time specified by timestamp rather than to the most current possible date.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

```csharp
public DateTime TxnTimestamp { get; set; }
```

Visual Basic (Declaration)

```vbnet
Public Property TxnTimestamp As DateTime
```

Visual C++

```cpp
public:
property DateTime TxnTimestamp {
    DateTime get ();
    void set (DateTime value);
}
```
Remarks

Once a database environment has been upgraded to a new version of Berkeley DB involving a log format change (see Upgrading Berkeley DB installations in the Programmer's Reference Guide), it is no longer possible to recover to a specific time before that upgrade.
See Also

DatabaseEnvironmentConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Represents errors that occur during Berkley DB operations.

**Namespace:**  BerkeleyDB  
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public class DatabaseException : Exception

Visual Basic (Declaration)

Public Class DatabaseException _
Inherits Exception

Visual C++

public ref class DatabaseException : public Exception
Inheritance Hierarchy

System::: Object
    System::: Exception
        BerkeleyDB::: DatabaseException
            BerkeleyDB::: BadSecondaryException
            BerkeleyDB::: DeadlockException
            BerkeleyDB::: ForeignConflictException
            BerkeleyDB::: FullLogBufferException
            BerkeleyDB::: KeyEmptyException
            BerkeleyDB::: KeyExistException
            BerkeleyDB::: LeaseExpiredException
            BerkeleyDB::: LockNotGrantedException
            BerkeleyDB:::NotFoundException
            BerkeleyDB::: OldVersionException
            BerkeleyDB::: PageNotFoundException
            BerkeleyDB::: RunRecoveryException
            BerkeleyDB::: VerificationException
            BerkeleyDB::: VersionMismatchException
See Also

DatabaseException Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `DatabaseException` type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DatabaseException</td>
<td>Create a new DatabaseException, encapsulating a specific error code.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a> is equal to the current <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td>GetBaseException</td>
<td>When overridden in a derived class, returns the <a href="https://docs.microsoft.com/en-us/dotnet/api/system.exception">Exception</a> that is the root cause of one or more subsequent exceptions. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.exception">Exception</a>.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.) When overridden in a derived class, sets the <a href="https://docs.microsoft.com/en-us/dotnet/api/system.serialization.serializationinfo">SerializationInfo</a> with information about the exception. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.exception">Exception</a>.)</td>
</tr>
<tr>
<td>GetObjectData</td>
<td>Gets the runtime type of the current instance. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.exception">Exception</a>.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Throw an exception which corresponds to the specified Berkeley DB error code.</td>
</tr>
<tr>
<td>ThrowException</td>
<td>Creates and returns a string representation of the current exception. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.exception">Exception</a>.)</td>
</tr>
<tr>
<td>ToString</td>
<td></td>
</tr>
</tbody>
</table>
## Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ErrorCode</td>
<td>The underlying error code from the Berkeley DB C library.</td>
</tr>
</tbody>
</table>
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from <a href="#">Exception</a>.)</td>
</tr>
<tr>
<td>HelpLink</td>
<td>Gets or sets a link to the help file associated with this exception. (Inherited from <a href="#">Exception</a>.)</td>
</tr>
<tr>
<td>InnerException</td>
<td>Gets the <a href="#">Exception</a> instance that caused the current exception. (Inherited from <a href="#">Exception</a>.)</td>
</tr>
<tr>
<td>Message</td>
<td>Gets a message that describes the current exception. (Inherited from <a href="#">Exception</a>.)</td>
</tr>
<tr>
<td>Source</td>
<td>Gets or sets the name of the application or the object that causes the error. (Inherited from <a href="#">Exception</a>.)</td>
</tr>
<tr>
<td>StackTrace</td>
<td>Gets a string representation of the frames on the call stack at the time the current exception was thrown. (Inherited from <a href="#">Exception</a>.)</td>
</tr>
<tr>
<td>TargetSite</td>
<td>Gets the method that throws the current exception. (Inherited from <a href="#">Exception</a>.)</td>
</tr>
</tbody>
</table>
See Also

DatabaseException Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Create a new DatabaseException, encapsulating a specific error code.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public DatabaseException(
    int err
)

Visual Basic (Declaration)

Public Sub New ( _
    err As Integer _
)

Visual C++

public:
DatabaseException(
    int err
)

Parameters

err

Type: System::Int32
The error code to encapsulate.
See Also

DatabaseException Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `DatabaseException` type exposes the following members.
## Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ErrorCode</td>
<td>The underlying error code from the Berkeley DB C library.</td>
</tr>
</tbody>
</table>
See Also

DatabaseException Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The underlying error code from the Berkeley DB C library.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public int ErrorCode

Visual Basic (Declaration)

Public ErrorCode As Integer

Visual C++

public:
    int ErrorCode
See Also

(DatabaseException Class
(BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `DatabaseException` type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a>.</td>
</tr>
<tr>
<td></td>
<td>When overridden in a derived class, returns the <a href="#">Exception</a> that is the root cause of one or more subsequent exceptions.</td>
</tr>
<tr>
<td><strong>GetBaseException</strong></td>
<td>Serves as a hash function for a particular type. When overridden in a derived class, sets the <a href="#">SerializationInfo</a> with information about the exception.</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. When overridden in a derived class, sets the <a href="#">Exception</a>.</td>
</tr>
<tr>
<td><strong>GetObjectData</strong></td>
<td>Gets the runtime type of the current instance. (Inherited from <a href="#">Exception</a>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the runtime type of the current instance. (Inherited from <a href="#">Exception</a>.)</td>
</tr>
<tr>
<td><strong>ThrowException</strong></td>
<td>Throws an exception which corresponds to the specified Berkeley DB error code.</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Creates and returns a string representation of the current exception. (Inherited from <a href="#">Exception</a>.)</td>
</tr>
</tbody>
</table>
See Also

[DatabaseException Class]
[BerkeleyDB Namespace]

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Throw an exception which corresponds to the specified Berkeley DB error code.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public static void ThrowException(
    int err
)

Visual Basic (Declaration)

Public Shared Sub ThrowException ( _
    err As Integer _
)

Visual C++

public:
static void ThrowException(
    int err
)

Parameters

err

Type: System::::Int32
The Berkeley DB error code
See Also

DatabaseException Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `DatabaseException` type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data</strong></td>
<td>Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td><strong>HelpLink</strong></td>
<td>Gets or sets a link to the help file associated with this exception. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td><strong>InnerException</strong></td>
<td>Gets the <code>Exception</code> instance that caused the current exception. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td><strong>Message</strong></td>
<td>Gets a message that describes the current exception. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td><strong>Source</strong></td>
<td>Gets or sets the name of the application or the object that causes the error. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td><strong>StackTrace</strong></td>
<td>Gets a string representation of the frames on the call stack at the time the current exception was thrown. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td><strong>TargetSite</strong></td>
<td>Gets the method that throws the current exception. (Inherited from <code>Exception</code>.)</td>
</tr>
</tbody>
</table>
See Also

DatabaseException Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
DatabaseFeedbackDelegate Delegate

See Also

The application-specified feedback function called to report Berkeley DB operation progress.

Namespace: BerkeleyDB
Syntax

C#

public delegate void DatabaseFeedbackDelegate(
    DatabaseFeedbackEvent opcode,
    int percent
)

Visual Basic (Declaration)

Public Delegate Sub DatabaseFeedbackDelegate (_
    opcode As DatabaseFeedbackEvent, _
    percent As Integer _
)

Visual C++

public delegate void DatabaseFeedbackDelegate(
    DatabaseFeedbackEvent opcode,
    int percent
)

Parameters

opcode
    Type: BerkeleyDB::DatabaseFeedbackEvent
    An operation code specifying the Berkley DB operation

percent
    Type: System::Int32
    The percent of the operation that has been completed, specified as an integer value between 0 and 100.
See Also

BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Specifies the database operation whose progress is being reported

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public enum DatabaseFeedbackEvent

Visual Basic (Declaration)

Public Enumeration DatabaseFeedbackEvent

Visual C++

public enum class DatabaseFeedbackEvent
<table>
<thead>
<tr>
<th>Member name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UPGRADE</td>
<td>The underlying database is being upgraded.</td>
</tr>
<tr>
<td>VERIFY</td>
<td>The underlying database is being verified.</td>
</tr>
</tbody>
</table>
See Also

BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
A class representing the supported Berkeley DB access methods.

**Namespace:**  [BerkeleyDB](#)

**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public class DatabaseType

Visual Basic (Declaration)

Public Class DatabaseType

Visual C++

public ref class DatabaseType
Inheritance Hierarchy

System::Object
BerkeleyDB::DatabaseType
See Also

DatabaseType Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The **DatabaseType** type exposes the following members.
# Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Convert this instance of DatabaseType to its string representation. (Overrrides <a href="#">Object..::.ToString()</a>.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>------------------------------</td>
</tr>
<tr>
<td>BTREE</td>
<td>BTree access method</td>
</tr>
<tr>
<td>HASH</td>
<td>Hash access method</td>
</tr>
<tr>
<td>QUEUE</td>
<td>Queue access method</td>
</tr>
<tr>
<td>RECNO</td>
<td>Recno access method</td>
</tr>
<tr>
<td>UNKNOWN</td>
<td>Unknown access method</td>
</tr>
</tbody>
</table>
See Also

DatabaseType Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `DatabaseType` type exposes the following members.
## Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BTREE</td>
<td>BTree access method</td>
</tr>
<tr>
<td>HASH</td>
<td>Hash access method</td>
</tr>
<tr>
<td>QUEUE</td>
<td>Queue access method</td>
</tr>
<tr>
<td>RECNO</td>
<td>Recno access method</td>
</tr>
<tr>
<td>UNKNOWN</td>
<td>Unknown access method</td>
</tr>
</tbody>
</table>
See Also

DatabaseType Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation

DatabaseType:::BTREE Field

**DatabaseType Class**  **See Also**

BTree access method

**Namespace:**  BerkeleyDB

**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public static readonly DatabaseType BTREE

Visual Basic (Declaration)

Public Shared ReadOnly BTREE As DatabaseType

Visual C++

public:
static initonly DatabaseType^ BTREE
See Also

DatabaseType Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
DatabaseType.HASH Field

See Also

Hash access method

Namespace: BerkeleyDB
Syntax

C#

public static readonly DatabaseType HASH

Visual Basic (Declaration)

Public Shared ReadOnly HASH As DatabaseType

Visual C++

public:
static initonly DatabaseType^ HASH
See Also

DatabaseType Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Queue access method

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public static readonly DatabaseType QUEUE

Visual Basic (Declaration)

Public Shared ReadOnly QUEUE As DatabaseType

Visual C++

public:
static initonly DatabaseType^ QUEUE
See Also

DatabaseType Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation

**DatabaseType..:::RECNO Field**

[DatabaseType Class](#) [See Also](#)

Recno access method

**Namespace:** [BerkeleyDB](#)

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#
public static readonly DatabaseType RECNO

Visual Basic (Declaration)
Public Shared ReadOnly RECNO As DatabaseType

Visual C++
public:
static initonly DatabaseType^ RECNO
See Also

DatabaseType Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Unknown access method

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#
public static readonly DatabaseType UNKNOWN

Visual Basic (Declaration)
Public Shared ReadOnly UNKNOWN As DatabaseType

Visual C++
public:
static initonly DatabaseType^ UNKNOWN
See Also

DatabaseType Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The **DatabaseType** type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <strong>Object</strong> is equal to the current <strong>Object</strong>. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <strong>Type</strong> of the current instance. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Convert this instance of DatabaseType to its string representation. (Overrides <strong>Object</strong>::*::ToString()().)</td>
</tr>
</tbody>
</table>
See Also

- DatabaseType Class
- BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Convert this instance of DatabaseType to its string representation.

**Namespace:**  BerkeleyDB  
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
### Syntax

**C#**

```csharp
public override string ToString()
```

**Visual Basic (Declaration)**

```vbnet
Public Overrides Function ToString As String
```

**Visual C++**

```cpp
public:
    virtual String^ ToString() override
```

### Return Value

A string representation of this instance.
See Also

DatabaseType Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
A class representing a unique identifier for a thread of control in a Berkeley DB application.

Namespace: **BerkeleyDB**
Syntax

C#  
public class DbThreadID

Visual Basic (Declaration)  
Public Class DbThreadID

Visual C++

public ref class DbThreadID
Inheritance Hierarchy

System.::.Object
BerkeleyDB.::.DbThreadID
See Also

DbThreadID Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `DbThreadId` type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DbThreadID</td>
<td>Instantiate a new DbThreadID object</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a <a href="#">String</a> that represents the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>--------------------------------------</td>
</tr>
<tr>
<td>processID</td>
<td>The Process ID of the thread of control</td>
</tr>
<tr>
<td>threadID</td>
<td>The Thread ID of the thread of control</td>
</tr>
</tbody>
</table>
See Also

DbThreadID Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation
DbThreadID Constructor

DbThreadID Class  See Also

Instantiate a new DbThreadID object

Namespace:  BerkeleyDB
Syntax

**C#**

```csharp
public DbThreadID(
    int pid,
    uint tid
)
```

**Visual Basic (Declaration)**

```vbnet
Public Sub New (
    pid As Integer, _
    tid As UInteger _
)
```

**Visual C++**

```cpp
public:
DbThreadID(
    int pid,
    unsigned int tid
)
```

**Parameters**

**pid**
- Type: `System::::Int32`
- The Process ID of the thread of control

**tid**
- Type: `System::::UInt32`
- The Thread ID of the thread of control
See Also

DbThreadID Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
DbThreadID Fields

The **DbThreadID** type exposes the following members.
## Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>processID</td>
<td>The Process ID of the thread of control</td>
</tr>
<tr>
<td>threadID</td>
<td>The Thread ID of the thread of control</td>
</tr>
</tbody>
</table>
See Also

DbThreadID Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
DbThreadID Class See Also

The Process ID of the thread of control

Namespace: BerkeleyDB
Syntax

C#

```csharp
public int processID
```

Visual Basic (Declaration)

```vbnet
Public processID As Integer
```

Visual C++

```cpp
public:
    int processID
```
See Also

DbThreadID Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
DbThreadID Class  See Also

The Thread ID of the thread of control

Namespace:  BerkeleyDB
Syntax

C#

public uint threadID

Visual Basic (Declaration)

Public threadID As UInteger

Visual C++

public:
unsigned int threadID
See Also

DbThreadID Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The **DbThreadId** type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a <a href="#">String</a> that represents the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
</tbody>
</table>
See Also

DbThreadID Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
When multiple threads of control are modifying the database, there is normally the potential for deadlock. In Berkeley DB, deadlock is signified by a DeadlockException thrown from the Berkeley DB function. Whenever a Berkeley DB function throws a DeadlockException, the enclosing transaction should be aborted.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
### Syntax

**C#**

```csharp
public class DeadlockException : DatabaseException
```

**Visual Basic (Declaration)**

```vbnet
Public Class DeadlockException
    Inherits DatabaseException
```

**Visual C++**

```cpp
public ref class DeadlockException : public DatabaseException
```
Inheritance Hierarchy

System...:::Object
  System...:::Exception
    BerkeleyDB...:::DatabaseException
      BerkeleyDB...:::DeadlockException
See Also

DeadlockException Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation

DeadlockException Members

[DeadlockException Class, Constructors, Methods, Fields, Properties, See Also]

The DeadlockException type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeadlockException</td>
<td>Initialize a new instance of the DeadlockException</td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| **Equals**          | Determines whether the specified [Object](https://docs.microsoft.com/en-us/dotnet/api/system.object) is equal to the current [Object](https://docs.microsoft.com/en-us/dotnet/api/system.object).  (Inherited from [Object](https://docs.microsoft.com/en-us/dotnet/api/system.object).)  
When overridden in a derived class, returns the [Exception](https://docs.microsoft.com/en-us/dotnet/api/system.exception) that is the root cause of one or more subsequent exceptions.  (Inherited from [Exception](https://docs.microsoft.com/en-us/dotnet/api/system.exception).) |
| **GetBaseException**| Serves as a hash function for a particular type.  (Inherited from [Object](https://docs.microsoft.com/en-us/dotnet/api/system.object).)  
When overridden in a derived class, sets the [SerializationInfo](https://docs.microsoft.com/en-us/dotnet/api/system.serialization.serializationinfo) with information about the exception.  (Inherited from [Exception](https://docs.microsoft.com/en-us/dotnet/api/system.exception).) |
| **GetHashCode**     | Gets the runtime type of the current instance.  (Inherited from [Exception](https://docs.microsoft.com/en-us/dotnet/api/system.exception).)  
Creates and returns a string representation of the current exception.  (Inherited from [Exception](https://docs.microsoft.com/en-us/dotnet/api/system.exception).) |
<p>| <strong>GetObjectData</strong>   |                                                                              |
| <strong>GetType</strong>         |                                                                              |
| <strong>ToString</strong>        |                                                                              |</p>
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ErrorCode</td>
<td>The underlying error code from the Berkeley DB C library.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">DatabaseException</a>.)</td>
</tr>
</tbody>
</table>
# Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data</strong></td>
<td>Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from <strong>Exception</strong>.)</td>
</tr>
<tr>
<td><strong>HelpLink</strong></td>
<td>Gets or sets a link to the help file associated with this exception. (Inherited from <strong>Exception</strong>.)</td>
</tr>
<tr>
<td><strong>InnerException</strong></td>
<td>Gets the <strong>Exception</strong> instance that caused the current exception. (Inherited from <strong>Exception</strong>.)</td>
</tr>
<tr>
<td><strong>Message</strong></td>
<td>Gets a message that describes the current exception. (Inherited from <strong>Exception</strong>.)</td>
</tr>
<tr>
<td><strong>Source</strong></td>
<td>Gets or sets the name of the application or the object that causes the error. (Inherited from <strong>Exception</strong>.)</td>
</tr>
<tr>
<td><strong>StackTrace</strong></td>
<td>Gets a string representation of the frames on the call stack at the time the current exception was thrown. (Inherited from <strong>Exception</strong>.)</td>
</tr>
<tr>
<td><strong>TargetSite</strong></td>
<td>Gets the method that throws the current exception. (Inherited from <strong>Exception</strong>.)</td>
</tr>
</tbody>
</table>
See Also

DeadlockException Class  
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Initialize a new instance of the DeadlockException

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public DeadlockException()

Visual Basic (Declaration)

Public Sub New

Visual C++

public:
DeadlockException()
See Also

DeadlockException Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `DeadlockException` type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ErrorCode</td>
<td>The underlying error code from the Berkeley DB C library. (Inherited from DatabaseException.)</td>
</tr>
</tbody>
</table>
See Also

DeadlockException Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `DeadlockException` type exposes the following members.
### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified object is equal to the current object. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>GetBaseException</strong></td>
<td>When overridden in a derived class, returns the <code>Exception</code> that is the root cause of one or more subsequent exceptions. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>GetObjectData</strong></td>
<td>When overridden in a derived class, sets the <code>SerializationInfo</code> with information about the exception. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the runtime type of the current instance. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Creates and returns a string representation of the current exception. (Inherited from <code>Exception</code>.)</td>
</tr>
</tbody>
</table>
See Also

DeadlockException Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The **DeadlockException** type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td>HelpLink</td>
<td>Gets or sets a link to the help file associated with this exception. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td>InnerException</td>
<td>Gets the <code>Exception</code> instance that caused the current exception. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td>Message</td>
<td>Gets a message that describes the current exception. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td>Source</td>
<td>Gets or sets the name of the application or the object that causes the error. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td>StackTrace</td>
<td>Gets a string representation of the frames on the call stack at the time the current exception was thrown. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td>TargetSite</td>
<td>Gets the method that throws the current exception. (Inherited from <code>Exception</code>.)</td>
</tr>
</tbody>
</table>
See Also

DeadlockException Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation
DeadlockPolicy Class

A class to represent what lock request(s) should be rejected during deadlock resolution.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public class DeadlockPolicy

Visual Basic (Declaration)

Public Class DeadlockPolicy

Visual C++

public ref class DeadlockPolicy
Inheritance Hierarchy

System..:::Object
BerkeleyDB..:::DeadlockPolicy
See Also

DeadlockPolicy Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The **DeadlockPolicy** type exposes the following members.
### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a> is equal to the current <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <a href="https://docs.microsoft.com/en-us/dotnet/api/system.type">Type</a> of the current instance. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a <a href="https://docs.microsoft.com/en-us/dotnet/api/system.string">String</a> that represents the current <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
</tbody>
</table>
### Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>s</code> DEFAULT</td>
<td>If no DeadlockPolicy has yet been specified, use <strong>RANDOM</strong>.</td>
</tr>
<tr>
<td><code>s</code> EXPIRE</td>
<td>Reject lock requests which have timed out. No other deadlock detection is performed.</td>
</tr>
<tr>
<td><code>s</code> MAX_LOCKS</td>
<td>Reject the lock request for the locker ID with the most locks.</td>
</tr>
<tr>
<td><code>s</code> MAX_WRITE</td>
<td>Reject the lock request for the locker ID with the most write locks.</td>
</tr>
<tr>
<td><code>s</code> MIN_LOCKS</td>
<td>Reject the lock request for the locker ID with the fewest locks.</td>
</tr>
<tr>
<td><code>s</code> MIN_WRITE</td>
<td>Reject the lock request for the locker ID with the fewest write locks.</td>
</tr>
<tr>
<td><code>s</code> OLDEST</td>
<td>Reject the lock request for the locker ID with the oldest lock.</td>
</tr>
<tr>
<td><code>s</code> RANDOM</td>
<td>Reject the lock request for a random locker ID.</td>
</tr>
<tr>
<td><code>s</code> YOUNGEST</td>
<td>Reject the lock request for the locker ID with the youngest lock.</td>
</tr>
</tbody>
</table>
See Also

DeadlockPolicy Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
DeadlockPolicy Fields

The `DeadlockPolicy` type exposes the following members.
### Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEFAULT</td>
<td>If no DeadlockPolicy has yet been specified, use <strong>RANDOM</strong>.</td>
</tr>
<tr>
<td>EXPIRE</td>
<td>Reject lock requests which have timed out. No other deadlock detection is performed.</td>
</tr>
<tr>
<td>MAX_LOCKS</td>
<td>Reject the lock request for the locker ID with the most locks.</td>
</tr>
<tr>
<td>MAX_WRITE</td>
<td>Reject the lock request for the locker ID with the most write locks.</td>
</tr>
<tr>
<td>MIN_LOCKS</td>
<td>Reject the lock request for the locker ID with the fewest locks.</td>
</tr>
<tr>
<td>MIN_WRITE</td>
<td>Reject the lock request for the locker ID with the fewest write locks.</td>
</tr>
<tr>
<td>OLDEST</td>
<td>Reject the lock request for the locker ID with the oldest lock.</td>
</tr>
<tr>
<td>RANDOM</td>
<td>Reject the lock request for a random locker ID.</td>
</tr>
<tr>
<td>YOUNGEST</td>
<td>Reject the lock request for the locker ID with the youngest lock.</td>
</tr>
</tbody>
</table>
See Also

DeadlockPolicy Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If no DeadlockPolicy has yet been specified, use **RANDOM**.

**Namespace:**  BerkeleyDB  
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public static DeadlockPolicy DEFAULT

Visual Basic (Declaration)

Public Shared DEFAULT As DeadlockPolicy

Visual C++

public:
static DeadlockPolicy^ DEFAULT
See Also

DeadlockPolicy Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
DeadlockPolicy Class  See Also

Reject lock requests which have timed out. No other deadlock detection is performed.

Namespace:  BerkeleyDB
Syntax

C#

public static DeadlockPolicy EXPIRE

Visual Basic (Declaration)

Public Shared EXPIRE As DeadlockPolicy

Visual C++

public:
static DeadlockPolicy^ EXPIRE
See Also

DeadlockPolicy Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
DeadlockPolicy...::MAX_LOCKS Field

See Also

Reject the lock request for the locker ID with the most locks.

Namespace: BerkeleyDB
Syntax

C#

public static DeadlockPolicy MAX_LOCKS

Visual Basic (Declaration)

Public Shared MAX_LOCKS As DeadlockPolicy

Visual C++

public:
static DeadlockPolicy^ MAX_LOCKS
See Also

- DeadlockPolicy Class
- BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
DeadlockPolicy...:::MAX_WRITE Field

DeadlockPolicy Class  See Also

Reject the lock request for the locker ID with the most write locks.

Namespace:  BerkeleyDB
Syntax

C#
public static DeadlockPolicy MAX_WRITE

Visual Basic (Declaration)
Public Shared MAX_WRITE As DeadlockPolicy

Visual C++
public:
static DeadlockPolicy^ MAX_WRITE
See Also

DeadlockPolicy Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
DeadlockPolicy::MIN_LOCKS Field

Reject the lock request for the locker ID with the fewest locks.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

class DeadlockPolicy

Visual Basic (Declaration)

Public Shared MIN_LOCKS As DeadlockPolicy

Visual C++

public:
static DeadlockPolicy^ MIN_LOCKS
See Also

DeadlockPolicy Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
DeadlockPolicy...::MIN_WRITE Field

DeadlockPolicy Class  See Also

Reject the lock request for the locker ID with the fewest write locks.

Namespace:  BerkeleyDB
Syntax

C#

public static DeadlockPolicy MIN_WRITE

Visual Basic (Declaration)

Public Shared MIN_WRITE As DeadlockPolicy

Visual C++

public:
static DeadlockPolicy^ MIN_WRITE
See Also

DeadlockPolicy Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
DeadlockPolicy.:::OLDEST Field

DeadlockPolicy Class  See Also

Reject the lock request for the locker ID with the oldest lock.

Namespace:  BerkeleyDB
Syntax

**C#**

```csharp
public static DeadlockPolicy OLDEST
```

**Visual Basic (Declaration)**

```vbnet
Public Shared OLDEST As DeadlockPolicy
```

**Visual C++**

```cpp
public:
static DeadlockPolicy^ OLDEST
```
See Also

DeadlockPolicy Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
DeadlockPolicy::RANDOM Field

Reject the lock request for a random locker ID.

Namespace: BerkeleyDB
Syntax

C#

class DeadlockPolicy
{
    public static DeadlockPolicy RANDOM
}

Visual Basic (Declaration)

Public Shared RANDOM As DeadlockPolicy

Visual C++

public:
static DeadlockPolicy^ RANDOM
See Also

DeadlockPolicy Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
DeadlockPolicy.YOUNGEST Field

Reject the lock request for the locker ID with the youngest lock.

Namespace:  BerkeleyDB
Syntax

C#

public static DeadlockPolicy YOUNGEST

Visual Basic (Declaration)

Public Shared YOUNGEST As DeadlockPolicy

Visual C++

public:
static DeadlockPolicy^ YOUNGEST
See Also

DeadlockPolicy Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The **DeadlockPolicy** type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="#">Object</a>)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a>)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a <a href="#">String</a> that represents the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>)</td>
</tr>
</tbody>
</table>
See Also

DeadlockPolicy Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
DuplicatesPolicy Enumeration

See Also

Policy for duplicate data items in the database; that is, whether insertion when the key of the key/data pair being inserted already exists in the database will be successful.

Namespace: BerkeleyDB
Syntax

C#

public enum DuplicatesPolicy

Visual Basic (Declaration)

Public Enumeration DuplicatesPolicy

Visual C++

public enum class DuplicatesPolicy
## Members

<table>
<thead>
<tr>
<th>Member name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NONE</td>
<td>Insertion when the key of the key/data pair being inserted already exists in the database will fail.</td>
</tr>
<tr>
<td>SORTED</td>
<td>Duplicates are allowed and maintained in sorted order, as determined by the duplicate comparison function. Duplicates are allowed and ordered in the database by the order of insertion, unless the ordering is otherwise specified by use of a cursor operation or a duplicate sort function.</td>
</tr>
<tr>
<td>UNSORTED</td>
<td></td>
</tr>
</tbody>
</table>
See Also

BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
EncryptionAlgorithm Enumeration

See Also

Specifies an algorithm used for encryption and decryption

Namespace: BerkeleyDB
Syntax

C#

public enum EncryptionAlgorithm

Visual Basic (Declaration)

Public Enumeration EncryptionAlgorithm

Visual C++

public enum class EncryptionAlgorithm
## Members

<table>
<thead>
<tr>
<th>Member name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEFAULT</td>
<td>The default algorithm, or the algorithm previously used in an existing environment</td>
</tr>
<tr>
<td>AES</td>
<td>The Rijndael/AES algorithm</td>
</tr>
</tbody>
</table>
See Also

BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
An application-specified comparison function.

**Namespace:** [BerkeleyDB](https://berkeleydb.com/

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public delegate int EntryComparisonDelegate(
    DatabaseEntry dbt1,
    DatabaseEntry dbt2
)

Visual Basic (Declaration)

Public Delegate Function EntryComparisonDelegate ( _
    dbt1 As DatabaseEntry, _
    dbt2 As DatabaseEntry _
) As Integer

Visual C++

public delegate int EntryComparisonDelegate(
    DatabaseEntry^ dbt1,
    DatabaseEntry^ dbt2
)

Parameters

dbt1
    Type: BerkeleyDB::DatabaseEntry
    The application supplied key.

dbt2
    Type: BerkeleyDB::DatabaseEntry
    The current tree's key.

Return Value

An integer value less than, equal to, or greater than zero if the first key parameter is considered to be respectively less than, equal to, or greater than the second key parameter.
See Also

BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The application-specified feedback function called to report Berkeley DB operation progress.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
### Syntax

#### C#

```csharp
public delegate void EnvironmentFeedbackDelegate(
    EnvironmentFeedbackEvent opcode,
    int percent
)
```

#### Visual Basic (Declaration)

```vbnet
Public Delegate Sub EnvironmentFeedbackDelegate ( _
    opcode As EnvironmentFeedbackEvent, _
    percent As Integer _
)
```

#### Visual C++

```c++
public delegate void EnvironmentFeedbackDelegate(  
    EnvironmentFeedbackEvent opcode,
    int percent
)
```

### Parameters

**opcode**

Type: `BerkeleyDB::EnvironmentFeedbackEvent`
An operation code specifying the Berkley DB operation

**percent**

Type: `System::Int32`
The percent of the operation that has been completed, specified as an integer value between 0 and 100.
See Also

BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
See Also

Specifies the environment operation whose progress is being reported

Namespace: BerkeleyDB
Syntax

**C#**

```csharp
public enum EnvironmentFeedbackEvent
```

**Visual Basic (Declaration)**

```vbnet
Public Enumeration EnvironmentFeedbackEvent
```

**Visual C++**

```cpp
public enum class EnvironmentFeedbackEvent
```
<table>
<thead>
<tr>
<th>Member name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RECOVERY</td>
<td>The environment is being recovered.</td>
</tr>
</tbody>
</table>
See Also

BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Constants representing error codes returned by the Berkeley DB library.

**Namespace:** BerkeleyDB
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#
public class ErrorCodes

Visual Basic (Declaration)
Public Class ErrorCodes

Visual C++
public ref class ErrorCodes
Inheritance Hierarchy

System..::..Object
BerkeleyDB..::..ErrorCodes
See Also

ErrorCodes Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The **ErrorCodes** type exposes the following members.
# Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ErrorCodes</td>
<td>Initializes a new instance of the ErrorCodes class</td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a> is equal to the current <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <a href="https://docs.microsoft.com/en-us/dotnet/api/system.type">Type</a> of the current instance. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a <a href="https://docs.microsoft.com/en-us/dotnet/api/system.string">String</a> that represents the current <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td>Fields</td>
<td></td>
</tr>
<tr>
<td>---</td>
<td></td>
</tr>
<tr>
<td><strong>Name</strong></td>
<td></td>
</tr>
<tr>
<td>DB_BUFFER_SMALL</td>
<td></td>
</tr>
<tr>
<td>DB_DONOTINDEX</td>
<td></td>
</tr>
<tr>
<td>DB_FOREIGN_CONFLICT</td>
<td></td>
</tr>
<tr>
<td>DB_KEYEMPTY</td>
<td></td>
</tr>
<tr>
<td>DB_KEYEXIST</td>
<td></td>
</tr>
<tr>
<td>DB_LOCK_DEADLOCK</td>
<td></td>
</tr>
<tr>
<td>DB_LOCK_NOTGRANTED</td>
<td></td>
</tr>
<tr>
<td>DB_LOG_BUFFER_FULL</td>
<td></td>
</tr>
<tr>
<td>DB_NOSERVER</td>
<td></td>
</tr>
<tr>
<td>DB_NOSERVER_HOME</td>
<td></td>
</tr>
<tr>
<td>DB_NOSERVER_ID</td>
<td></td>
</tr>
<tr>
<td>DB_NOTFOUND</td>
<td></td>
</tr>
<tr>
<td>DB_OLD_VERSION</td>
<td></td>
</tr>
<tr>
<td>DB_PAGE_NOTFOUND</td>
<td></td>
</tr>
<tr>
<td>DB_REP_DUPMASTER</td>
<td></td>
</tr>
<tr>
<td>DB_REP_HANDLE_DEAD</td>
<td></td>
</tr>
<tr>
<td>DB_REP_HOLDELECTION</td>
<td></td>
</tr>
<tr>
<td>DB_REP_IGNORE</td>
<td></td>
</tr>
<tr>
<td>DB_REP_ISPERM</td>
<td></td>
</tr>
<tr>
<td>DB_REP_JOIN_FAILURE</td>
<td></td>
</tr>
<tr>
<td>DB_REP_LEASE_EXPIRED</td>
<td></td>
</tr>
<tr>
<td>DB_REP_LOCKOUT</td>
<td></td>
</tr>
<tr>
<td>DB_REP_NEWSITE</td>
<td></td>
</tr>
<tr>
<td>DB_REP_NOTPERM</td>
<td></td>
</tr>
<tr>
<td>DB_REP_UNAVAIL</td>
<td></td>
</tr>
<tr>
<td>DB_RUNRECOVERY</td>
<td></td>
</tr>
<tr>
<td>DB_SECONDARY_BAD</td>
<td></td>
</tr>
<tr>
<td>DB_VERIFY_BAD</td>
<td></td>
</tr>
<tr>
<td><strong>Description</strong></td>
<td></td>
</tr>
<tr>
<td>User memory too small for return.</td>
<td></td>
</tr>
<tr>
<td>&quot;Null&quot; return from 2ndary callbk.</td>
<td></td>
</tr>
<tr>
<td>A foreign db constraint triggered.</td>
<td></td>
</tr>
<tr>
<td>Key/data deleted or never created.</td>
<td></td>
</tr>
<tr>
<td>The key/data pair already exists.</td>
<td></td>
</tr>
<tr>
<td>Deadlock.</td>
<td></td>
</tr>
<tr>
<td>Lock unavailable.</td>
<td></td>
</tr>
<tr>
<td>In-memory log buffer full.</td>
<td></td>
</tr>
<tr>
<td>Server panic return.</td>
<td></td>
</tr>
<tr>
<td>Bad home sent to server.</td>
<td></td>
</tr>
<tr>
<td>Bad ID sent to server.</td>
<td></td>
</tr>
<tr>
<td>Key/data pair not found (EOF).</td>
<td></td>
</tr>
<tr>
<td>Out-of-date version.</td>
<td></td>
</tr>
<tr>
<td>Requested page not found.</td>
<td></td>
</tr>
<tr>
<td>There are two masters.</td>
<td></td>
</tr>
<tr>
<td>Rolled back a commit.</td>
<td></td>
</tr>
<tr>
<td>Time to hold an election.</td>
<td></td>
</tr>
<tr>
<td>This msg should be ignored.</td>
<td></td>
</tr>
<tr>
<td>Cached not written perm written.</td>
<td></td>
</tr>
<tr>
<td>Unable to join replication group.</td>
<td></td>
</tr>
<tr>
<td>Master lease has expired.</td>
<td></td>
</tr>
<tr>
<td>API/Replication lockout now.</td>
<td></td>
</tr>
<tr>
<td>New site entered system.</td>
<td></td>
</tr>
<tr>
<td>Permanent log record not written.</td>
<td></td>
</tr>
<tr>
<td>Site cannot currently be reached.</td>
<td></td>
</tr>
<tr>
<td>Panic return.</td>
<td></td>
</tr>
<tr>
<td>Secondary index corrupt.</td>
<td></td>
</tr>
<tr>
<td>Verify failed; bad format.</td>
<td></td>
</tr>
</tbody>
</table>
DB_VERSION_MISMATCH Environment version mismatch.
See Also

ErrorCodes Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
ErrorCodes Constructor

Initializes a new instance of the ErrorCodes class

Namespace: BerkeleyDB
Syntax

C#

public ErrorCodes()

Visual Basic (Declaration)

Public Sub New

Visual C++

public:
ErrorCodes()
See Also

ErrorCodes Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The **ErrorCodes** type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB_BUFFER_SMALL</td>
<td>User memory too small for return.</td>
</tr>
<tr>
<td>DB_DONOTINDEX</td>
<td>&quot;Null&quot; return from 2ndary callbk.</td>
</tr>
<tr>
<td>DB_FOREIGN_CONFLICT</td>
<td>A foreign db constraint triggered.</td>
</tr>
<tr>
<td>DB_KEYEMPTY</td>
<td>Key/data deleted or never created.</td>
</tr>
<tr>
<td>DB_KEYEXIST</td>
<td>The key/data pair already exists.</td>
</tr>
<tr>
<td>DB_LOCK_DEADLOCK</td>
<td>Deadlock.</td>
</tr>
<tr>
<td>DB_LOCK_NOTGRANTED</td>
<td>Lock unavailable.</td>
</tr>
<tr>
<td>DB_LOG_BUFFER_FULL</td>
<td>In-memory log buffer full.</td>
</tr>
<tr>
<td>DB_NOSERVER</td>
<td>Server panic return.</td>
</tr>
<tr>
<td>DB_NOSERVER_HOME</td>
<td>Bad home sent to server.</td>
</tr>
<tr>
<td>DB_NOSERVER_ID</td>
<td>Bad ID sent to server.</td>
</tr>
<tr>
<td>DB_NOTFOUND</td>
<td>Key/data pair not found (EOF).</td>
</tr>
<tr>
<td>DB_OLD_VERSION</td>
<td>Out-of-date version.</td>
</tr>
<tr>
<td>DB_PAGE_NOTFOUND</td>
<td>Requested page not found.</td>
</tr>
<tr>
<td>DB_REP_DUPMASTER</td>
<td>There are two masters.</td>
</tr>
<tr>
<td>DB_REP_HANDLE_DEAD</td>
<td>Rolled back a commit.</td>
</tr>
<tr>
<td>DB_REP_HOLDELECTION</td>
<td>Time to hold an election.</td>
</tr>
<tr>
<td>DB_REP_IGNORE</td>
<td>This msg should be ignored.</td>
</tr>
<tr>
<td>DB_REP_ISPERM</td>
<td>Cached not written perm written.</td>
</tr>
<tr>
<td>DB_REP_JOIN_FAILURE</td>
<td>Unable to join replication group.</td>
</tr>
<tr>
<td>DB_REP_LEASE_EXPIRED</td>
<td>Master lease has expired.</td>
</tr>
<tr>
<td>DB_REP_LOCKOUT</td>
<td>API/Replication lockout now.</td>
</tr>
<tr>
<td>DB_REP_NEWSITE</td>
<td>New site entered system.</td>
</tr>
<tr>
<td>DB_REP_NOTPERM</td>
<td>Permanent log record not written.</td>
</tr>
<tr>
<td>DB_REP_UNAVAIL</td>
<td>Site cannot currently be reached.</td>
</tr>
<tr>
<td>DB_RUNRECOVERY</td>
<td>Panic return.</td>
</tr>
<tr>
<td>DB_SECONDARY_BAD</td>
<td>Secondary index corrupt.</td>
</tr>
<tr>
<td>DBVERIFY_BAD</td>
<td>Verify failed; bad format.</td>
</tr>
</tbody>
</table>
DB_VERSION_MISMATCH Environment version mismatch.
See Also

ErrorCodes Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
User memory too small for return.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll)  
Version: 4.8.24.0
**Syntax**

**C#**

public const int DB_BUFFER_SMALL

**Visual Basic (Declaration)**

Public Const DB_BUFFER_SMALL As Integer

**Visual C++**

public:
literal int DB_BUFFER_SMALL
See Also

ErrorCodes Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
"Null" return from 2ndary callbk.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

**C#**

public const int DB_DONOTINDEX

**Visual Basic (Declaration)**

Public Const DB_DONOTINDEX As Integer

**Visual C++**

public:
    literal int DB_DONOTINDEX
See Also

ErrorCodes Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
A foreign db constraint triggered.

**Namespace:**  BerkeleyDB  
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll)  
Version:  4.8.24.0
### Syntax

**C#**

```csharp
public const int DB_FOREIGN_CONFLICT
```

**Visual Basic (Declaration)**

```vbnet
Public Const DB_FOREIGN_CONFLICT As Integer
```

**Visual C++**

```cpp
public:
    literal int DB_FOREIGN_CONFLICT
```
See Also

ErrorCodes Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
ErrorCodes Class  See Also

Key/data deleted or never created.

Namespace:  BerkeleyDB
Syntax

C#

public const int DB_KEYEMPTY

Visual Basic (Declaration)

Public Const DB_KEYEMPTY As Integer

Visual C++

public:
literal int DB_KEYEMPTY
See Also

ErrorsCodes Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The key/data pair already exists.

Namespace: BerkeleyDB
Syntax

C#

public const int DB_KEYEXIST

Visual Basic (Declaration)

Public Const DB_KEYEXIST As Integer

Visual C++

public:
literal int DB_KEYEXIST
See Also

ErrorCodes Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Deadlock.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public const int DB_LOCK_DEADLOCK

Visual Basic (Declaration)

Public Const DB_LOCK_DEADLOCK As Integer

Visual C++

public:
    literal int DB_LOCK_DEADLOCK
See Also

ErrorCodes Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Lock unavailable.

Namespace:  BerkeleyDB
Syntax

C#

public const int DB_LOCK_NOTGRANTED

Visual Basic (Declaration)

Public Const DB_LOCK_NOTGRANTED As Integer

Visual C++

public:
literal int DB_LOCK_NOTGRANTED
See Also

ErrorCodes Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
In-memory log buffer full.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public const int DB_LOG_BUFFER_FULL

Visual Basic (Declaration)

Public Const DB_LOG_BUFFER_FULL As Integer

Visual C++

public:
literal int DB_LOG_BUFFER_FULL
See Also

ErrorCodes Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Server panic return.

**Namespace:** [BerkeleyDB](https://www.bsd.org/berkeleydb/)

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public const int DB_NOSERVER

Visual Basic (Declaration)

Public Const DB_NOSERVER As Integer

Visual C++

public:
    literal int DB_NOSERVER
See Also

ErrorCodes Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Bad home sent to server.

**Namespace:**  [BerkeleyDB](https://www.berkeleydb.com/)

**Assembly:**  `libdb_dotnet48` (in `libdb_dotnet48.dll`) Version: 4.8.24.0
Syntax

C#

public const int DB_NOSERVER_HOME

Visual Basic (Declaration)

Public Const DB_NOSERVER_HOME As Integer

Visual C++

public:
literal int DB_NOSERVER_HOME
See Also

ErrorCodes Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Bad ID sent to server.

Namespace: BerkeleyDB
Syntax

C#

public const int DB_NOSERVER_ID

Visual Basic (Declaration)

Public Const DB_NOSERVER_ID As Integer

Visual C++

public:

literal int DB_NOSERVER_ID
See Also

ErrorCodes Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Key/data pair not found (EOF).

**Namespace:**  BerkeleyDB  
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

**C#**

public const int DB_NOTFOUND

**Visual Basic (Declaration)**

Public Const DB_NOTFOUND As Integer

**Visual C++**

public:
    literal int DB_NOTFOUND
See Also

ErrorCodes Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation
ErrorCodes...:::DB_OLD_VERSION Field

ErrorCodes Class  See Also

Out-of-date version.

Namespace:  BerkeleyDB
Syntax

C#

public const int DB_OLD_VERSION

Visual Basic (Declaration)

Public Const DB_OLD_VERSION As Integer

Visual C++

public:
literal int DB_OLD_VERSION
See Also

ErrorCodes Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Requested page not found.

**Namespace:** BerkeleyDB
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public const int DB_PAGE_NOTFOUND

Visual Basic (Declaration)

Public Const DB_PAGE_NOTFOUND As Integer

Visual C++

public:
literal int DB_PAGE_NOTFOUND
See Also

ErrorCodes Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
There are two masters.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#
public const int DB_REP_DUPMASTER

Visual Basic (Declaration)
Public Const DB_REP_DUPMASTER As Integer

Visual C++
public:
    literal int DB_REP_DUPMASTER
See Also

ErrorCodes Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Rolled back a commit.

**Namespace:**  BerkeleyDB  
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public const int DB_REP_HANDLE_DEAD

Visual Basic (Declaration)

Public Const DB_REP_HANDLE_DEAD As Integer

Visual C++

public:
literal int DB_REP_HANDLE_DEAD
See Also

ErrorCodes Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation

ErrorCodes..:::DB_REP_HOLDELECTION Field

ErrorCodes Class  See Also

Time to hold an election.

Namespace:  BerkeleyDB
Syntax

C#

public const int DB_REP_HOLDELECTION

Visual Basic (Declaration)

Public Const DB_REP_HOLDELECTION As Integer

Visual C++

public:
literal int DB_REP_HOLDELECTION
See Also

ErrorCodes Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation

ErrorCodes DB_REP_IGNORE Field

See Also

This msg should be ignored.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public const int DB_REP_IGNORE

Visual Basic (Declaration)

Public Const DB_REP_IGNORE As Integer

Visual C++

public:
    literal int DB_REP_IGNORE
See Also

ErrorCodes Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
ErrorCodes..::..DB_REP_ISPERM Field

Cached not written perm written.

Namespace: BerkeleyDB
Syntax

C#
public const int DB_REP_ISPERM

Visual Basic (Declaration)
Public Const DB_REP_ISPERM As Integer

Visual C++
public:
literal int DB_REP_ISPERM
See Also

ErrrorCodes Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Unable to join replication group.

**Namespace:** [BerkeleyDB](http://berkeleydb.net)

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public const int DB_REP_JOIN_FAILURE

Visual Basic (Declaration)

Public Const DB_REP_JOIN_FAILURE As Integer

Visual C++

public:
literal int DB_REP_JOIN_FAILURE
See Also

ErrorCodes Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Master lease has expired.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
### Syntax

**C#**

```csharp
public const int DB_REP_LEASE_EXPIRED
```

**Visual Basic (Declaration)**

```vbnet
Public Const DB_REP_LEASE_EXPIRED As Integer
```

**Visual C++**

```cpp
public:
  literal int DB_REP_LEASE_EXPIRED
```
See Also

ErrorCodes Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
API/Replication lockout now.

**Namespace:** [BerkeleyDB](https://www.berkeleydb.net)
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public const int DB_REP_LOCKOUT

Visual Basic (Declaration)

Public Const DB_REP_LOCKOUT As Integer

Visual C++

public:
    literal int DB_REP_LOCKOUT
See Also

ErrorCodes Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
New site entered system.

**Namespace:**  BerkeleyDB
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

**C#**

```csharp
public const int DB_REP_NEWSITE
```

**Visual Basic (Declaration)**

```vbnet
Public Const DB_REP_NEWSITE As Integer
```

**Visual C++**

```cpp
public:
literal int DB_REP_NEWSITE
```
See Also

ErrorCodes Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Permanent log record not written.

**Namespace:**  BerkeleyDB  
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public const int DB_REP_NOTPERM

Visual Basic (Declaration)

Public Const DB_REP_NOTPERM As Integer

Visual C++

public:
literal int DB_REP_NOTPERM
See Also

ErrorCodes Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Site cannot currently be reached.

**Namespace:**  BerkeleyDB  
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#
public const int DB_REP_UNAVAIL

Visual Basic (Declaration)
Public Const DB_REP_UNAVAIL As Integer

Visual C++

public:
literal int DB_REP_UNAVAIL
See Also

ErrorCodes Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Panic return.

Namespace: BerkeleyDB
Syntax

**C#**

```csharp
public const int DB_RUNRECOVERY
```

**Visual Basic (Declaration)**

```vbnet
Public Const DB_RUNRECOVERY As Integer
```

**Visual C++**

```c++
public:
    literal int DB_RUNRECOVERY
```
See Also

ErrorCodes Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
ErrorCodes..:::DB_SECONDARY_BAD Field

Secondary index corrupt.

Namespace: BerkeleyDB
Syntax

C#

public const int DB_SECONDARY_BAD

Visual Basic (Declaration)

Public Const DB_SECONDARY_BAD As Integer

Visual C++

public:
literal int DB_SECONDARY_BAD
See Also

ErrorCodes Class  
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Verify failed; bad format.

**Namespace:**  BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll)  
**Version:** 4.8.24.0
Syntax

C#

public const int DB_VERIFY_BAD

Visual Basic (Declaration)

Public Const DB_VERIFY_BAD As Integer

Visual C++

public:
    literal int DB_VERIFY_BAD
See Also

ErrorCodes Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
ErrorCodes..::..DB_VERSION_MISMATCH Field

Environment version mismatch.

Namespace: BerkeleyDB
Syntax

C#

public const int DB_VERSION_MISMATCH

Visual Basic (Declaration)

Public Const DB_VERSION_MISMATCH As Integer

Visual C++

public:
    literal int DB_VERSION_MISMATCH
See Also

ErrorCodes Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `ErrorCodes` type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified <a href="#">Object</a> is equal to the current Object. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as a hash function for a particular type.  (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the <a href="#">Type</a> of the current instance.  (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a <a href="#">String</a> that represents the current Object.  (Inherited from <a href="#">Object</a>.)</td>
</tr>
</tbody>
</table>
See Also

ErrorCodes Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The application-specified error reporting function.

Namespace:  BerkeleyDB
Syntax

C#

public delegate void ErrorFeedbackDelegate(
    string errPrefix,
    string errorMessage
)

Visual Basic (Declaration)

Public Delegate Sub ErrorFeedbackDelegate ( _
    errPrefix As String, _
    errMessage As String _
)

Visual C++

public delegate void ErrorFeedbackDelegate(
    String^ errPrefix,
    String^ errMessage
)

Parameters

errPrefix
    Type: System:::String
    The prefix string

errorMessage
    Type: System:::String
    The error message string
See Also

BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
See Also

The application's event notification function.

Namespace: BerkeleyDB
Syntax

C#

public delegate void EventNotifyDelegate(
    NotificationEvent eventcode,
    byte[] event_info
)

Visual Basic (Declaration)

Public Delegate Sub EventNotifyDelegate ( _
    eventcode As NotificationEvent, _
    event_info As Byte() _
)

Visual C++

public delegate void EventNotifyDelegate(
    NotificationEvent eventcode,
    array<unsigned char>^ event_info
)

Parameters

eventcode
  Type: BerkeleyDB::::NotificationEvent
  An even code specifying the Berkeley DB event

event_info
  Type: array< System::::Byte >[](0]
  Additional information describing an event. By default, event_info is null; specific events may pass non-null values, in which case the event will also describe the information's structure.
See Also

BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
ForeignConflictException Class

Namespace: BerkeleyDB
Syntax

C#

public class ForeignConflictException : DatabaseException

Visual Basic (Declaration)

Public Class ForeignConflictException _
    Inherits DatabaseException

Visual C++

public ref class ForeignConflictException : public DatabaseException
Inheritance Hierarchy

System...:::Object
  System...:::Exception
    BerkeleyDB...:::DatabaseException
      BerkeleyDB...:::ForeignConflictException
See Also

ForeignConflictException Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `ForeignConflictException` type exposes the following members.
### Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ForeignConflictException</code></td>
<td>Initialize a new instance of the <code>ForeignConflictException</code></td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Equals</code></td>
<td>Determines whether the specified <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a> is equal to the current <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.) When overridden in a derived class, returns the <a href="https://docs.microsoft.com/en-us/dotnet/api/system.exception">Exception</a> that is the root cause of one or more subsequent exceptions. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.exception">Exception</a>.)</td>
</tr>
<tr>
<td><code>GetBaseException</code></td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.) When overridden in a derived class, sets the <a href="https://docs.microsoft.com/en-us/dotnet/api/system.serialization.serializationinfo">SerializationInfo</a> with information about the exception. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.exception">Exception</a>.)</td>
</tr>
<tr>
<td><code>GetHashCode</code></td>
<td>Gets the runtime type of the current instance. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.exception">Exception</a>.)</td>
</tr>
<tr>
<td><code>GetObjectData</code></td>
<td>Creates and returns a string representation of the current exception. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.exception">Exception</a>.)</td>
</tr>
<tr>
<td><code>GetType</code></td>
<td></td>
</tr>
<tr>
<td><code>ToString</code></td>
<td></td>
</tr>
</tbody>
</table>
## Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ErrorCode</td>
<td>The underlying error code from the Berkeley DB C library. (Inherited from <a href="#">DatabaseException</a>.)</td>
</tr>
</tbody>
</table>
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data</strong></td>
<td>Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td><strong>HelpLink</strong></td>
<td>Gets or sets a link to the help file associated with this exception. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td><strong>InnerException</strong></td>
<td>Gets the <code>Exception</code> instance that caused the current exception. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td><strong>Message</strong></td>
<td>Gets a message that describes the current exception. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td><strong>Source</strong></td>
<td>Gets or sets the name of the application or the object that causes the error. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td><strong>StackTrace</strong></td>
<td>Gets a string representation of the frames on the call stack at the time the current exception was thrown. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td><strong>TargetSite</strong></td>
<td>Gets the method that throws the current exception. (Inherited from <code>Exception</code>.)</td>
</tr>
</tbody>
</table>
See Also

ForeignConflictException Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Initialize a new instance of the ForeignConflictException

Namespace: BerkeleyDB
Syntax

C#

public ForeignConflictException()

Visual Basic (Declaration)

Public Sub New

Visual C++

public:
ForeignConflictException()
See Also

ForeignConflictException Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The **ForeignConflictException** type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ErrorCode</td>
<td>The underlying error code from the Berkeley DB C library. (Inherited from DatabaseException.)</td>
</tr>
</tbody>
</table>
See Also

ForeignConflictException Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The *ForeignConflictException* type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a> is equal to the current <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.) When overridden in a derived class, returns the <a href="https://docs.microsoft.com/en-us/dotnet/api/system.exception">Exception</a> that is the root cause of one or more subsequent exceptions. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.exception">Exception</a>.)</td>
</tr>
<tr>
<td>GetBaseException</td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.) When overridden in a derived class, sets the <a href="https://docs.microsoft.com/en-us/dotnet/api/system.serialization.serializationinfo">SerializationInfo</a> with information about the exception. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.exception">Exception</a>.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Gets the runtime type of the current instance. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.exception">Exception</a>.)</td>
</tr>
<tr>
<td>GetObjectData</td>
<td>Creates and returns a string representation of the current exception. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.exception">Exception</a>.)</td>
</tr>
<tr>
<td>GetType</td>
<td></td>
</tr>
<tr>
<td>ToString</td>
<td></td>
</tr>
</tbody>
</table>
See Also

ForeignConflictException Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `ForeignConflictException` type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| **Data**      | Gets a collection of key/value pairs that provide additional user-defined information about the exception.  
(Inherited from [Exception](#).) |
| **HelpLink**  | Gets or sets a link to the help file associated with this exception.  
(Inherited from [Exception](#).) |
| **InnerException** | Gets the [Exception](#) instance that caused the current exception.  
(Inherited from [Exception](#).) |
| **Message**   | Gets a message that describes the current exception.  
(Inherited from [Exception](#).) |
| **Source**    | Gets or sets the name of the application or the object that causes the error.  
(Inherited from [Exception](#).) |
| **StackTrace** | Gets a string representation of the frames on the call stack at the time the current exception was thrown.  
(Inherited from [Exception](#).) |
| **TargetSite** | Gets the method that throws the current exception.  
(Inherited from [Exception](#).) |
See Also

ForeignConflictException Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
See Also

Specifies the action to take when deleting a foreign key

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public enum ForeignKeyDeleteAction

Visual Basic (Declaration)

Public Enumeration ForeignKeyDeleteAction

Visual C++

public enum class ForeignKeyDeleteAction
## Members

<table>
<thead>
<tr>
<th>Member name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABORT</td>
<td>Abort the deletion.</td>
</tr>
<tr>
<td>CASCADE</td>
<td>Delete records that refer to the foreign key</td>
</tr>
<tr>
<td>NULLIFY</td>
<td>Nullify records that refer to the foreign key</td>
</tr>
</tbody>
</table>
See Also

BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Namespace: BerkeleyDB
Syntax

C#

public delegate DatabaseEntry ForeignKeyNullifyDelegate(
    DatabaseEntry key,
    DatabaseEntry data,
    DatabaseEntry foreignkey
)

Visual Basic (Declaration)

Public Delegate Function ForeignKeyNullifyDelegate ( _
    key As DatabaseEntry, _
    data As DatabaseEntry, _
    foreignkey As DatabaseEntry _
) As DatabaseEntry

Visual C++

public delegate DatabaseEntry^ ForeignKeyNullifyDelegate(
    DatabaseEntry^ key,
    DatabaseEntry^ data,
    DatabaseEntry^ foreignkey
)

Parameters

key
Type: BerkeleyDB::DatabaseEntry

data
Type: BerkeleyDB::DatabaseEntry

foreignkey
Type: BerkeleyDB::DatabaseEntry

Return Value
See Also

BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
In-memory logs are configured and no more log buffer space is available.

Namespace:  BerkeleyDB
Syntax

C#

public class FullLogBufferException : DatabaseException

Visual Basic (Declaration)

Public Class FullLogBufferException
    Inherits DatabaseException

Visual C++

public ref class FullLogBufferException : public DatabaseException
Inheritance Hierarchy

System::Object
  System::Exception
    BerkeleyDB::DatabaseException
    BerkeleyDB::FullLogBufferException
See Also

FullLogBufferException Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The **FullLogBufferException** type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FullLogBufferException</td>
<td>Initialize a new instance of the FullLogBufferException</td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a> is equal to the current <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.) When overridden in a derived class, returns the <a href="https://docs.microsoft.com/en-us/dotnet/api/system.exception">Exception</a> that is the root cause of one or more subsequent exceptions. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.exception">Exception</a>.)</td>
</tr>
<tr>
<td><strong>GetBaseException</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.) When overridden in a derived class, sets the <a href="https://docs.microsoft.com/en-us/dotnet/api/system.serialization.serializationinfo">SerializationInfo</a> with information about the exception. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.exception">Exception</a>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Gets the runtime type of the current instance. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.exception">Exception</a>.)</td>
</tr>
<tr>
<td><strong>GetObjectData</strong></td>
<td>Creates and returns a string representation of the current exception. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.exception">Exception</a>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td></td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ErrorCode</td>
<td>The underlying error code from the Berkeley DB C library. (Inherited from DatabaseException.)</td>
</tr>
</tbody>
</table>
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>HelpLink</td>
<td>Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>InnerException</td>
<td>Gets the Exception instance that caused the current exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>Message</td>
<td>Gets a message that describes the current exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>Source</td>
<td>Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)</td>
</tr>
<tr>
<td>StackTrace</td>
<td>Gets a string representation of the frames on the call stack at the time the current exception was thrown. (Inherited from Exception.)</td>
</tr>
<tr>
<td>TargetSite</td>
<td>Gets the method that throws the current exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>
See Also

FullLogBufferException Class  
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation
FullLogBufferException Constructor

FullLogBufferException Class  See Also

Initialize a new instance of the FullLogBufferException

Namespace:  BerkeleyDB
Syntax

C#

public FullLogBufferException()

Visual Basic (Declaration)

Public Sub New

Visual C++

public:
FullLogBufferException()
See Also

FullLogBufferException Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `FullLogBufferException` type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ErrorCode</td>
<td>The underlying error code from the Berkeley DB C library. (Inherited from DatabaseException.)</td>
</tr>
</tbody>
</table>
See Also

FullLogBufferException Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `FullLogBufferException` type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a> is equal to the current <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.) When overridden in a derived class, returns the <a href="https://docs.microsoft.com/en-us/dotnet/api/system.exception">Exception</a> that is the root cause of one or more subsequent exceptions. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.exception">Exception</a>.)</td>
</tr>
<tr>
<td><strong>GetBaseException</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.) When overridden in a derived class, sets the <a href="https://docs.microsoft.com/en-us/dotnet/api/system.serialization.serializationinfo">SerializationInfo</a> with information about the exception. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.exception">Exception</a>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Gets the runtime type of the current instance. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.exception">Exception</a>.) Creates and returns a string representation of the current exception. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.exception">Exception</a>.)</td>
</tr>
<tr>
<td><strong>GetObjectData</strong></td>
<td></td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td></td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td></td>
</tr>
</tbody>
</table>
See Also

FullLogBufferException Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The **FullLogBufferException** type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Inherited from</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>Gets a collection of key/value pairs that provide additional user-defined information about the exception.</td>
<td>Exception</td>
</tr>
<tr>
<td>HelpLink</td>
<td>Gets or sets a link to the help file associated with this exception.</td>
<td>Exception</td>
</tr>
<tr>
<td>InnerException</td>
<td>Gets the Exception instance that caused the current exception.</td>
<td>Exception</td>
</tr>
<tr>
<td>Message</td>
<td>Gets a message that describes the current exception.</td>
<td>Exception</td>
</tr>
<tr>
<td>Source</td>
<td>Gets or sets the name of the application or the object that causes the error.</td>
<td>Exception</td>
</tr>
<tr>
<td>StackTrace</td>
<td>Gets a string representation of the frames on the call stack at the time the current exception was thrown.</td>
<td>Exception</td>
</tr>
<tr>
<td>TargetSite</td>
<td>Gets the method that throws the current exception.</td>
<td>Exception</td>
</tr>
</tbody>
</table>
See Also

FullLogBufferException Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
A class for traversing the records of a HashDatabase

Namespace: BerkeleyDB
Syntax

C#
public class HashCursor : Cursor

Visual Basic (Declaration)
Public Class HashCursor _
    Inherits Cursor

Visual C++
public ref class HashCursor : public Cursor
Inheritance Hierarchy

`System::Object`
`BerkeleyDB::BaseCursor`
`BerkeleyDB::Cursor`
`BerkeleyDB::HashCursor`
See Also

HashCursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The **HashCursor** type exposes the following members.
### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Add</strong></td>
<td>Overloaded. Insert the specified key/data pair into the database, unless a key/data pair comparing equally to it already exists in the database.</td>
</tr>
<tr>
<td><strong>AddUnique</strong></td>
<td>Discard the cursor. It is possible for the Close() method to throw a DeadlockException, signaling that any enclosing transaction should be aborted. If the application is already intending to abort the transaction, this error should be ignored, and the application should proceed.</td>
</tr>
<tr>
<td><strong>Close</strong></td>
<td>After Close has been called, regardless of its result, the object may not be used again.</td>
</tr>
<tr>
<td><strong>Compare</strong></td>
<td>Compare this cursor's position to another's. (Inherited from BaseCursor.)</td>
</tr>
<tr>
<td><strong>Count</strong></td>
<td>Returns a count of the number of data items for the key to which the cursor refers. (Inherited from BaseCursor.)</td>
</tr>
<tr>
<td><strong>Delete</strong></td>
<td>Delete the key/data pair to which the cursor refers. (Inherited from Cursor.)</td>
</tr>
</tbody>
</table>
Dispose

Release the resources held by this object, and close the cursor if it's still open.
(Inherited from BaseCursor.)

Duplicate

Create a new cursor that uses the same transaction and locker ID as the original cursor.
Determines whether the specified Object is equal to the current Object.
(Inherited from Object.)

Equals

Returns an enumerator that iterates through the Cursor.
(Inherited from Cursor.)

GetEnumerator

Serves as a hash function for a particular type.
(Inherited from Object.)

GetHashCode

Gets the Type of the current instance.
(Inherited from Object.)

GetType

Insert

Insert the data element as a duplicate element of the key to which the cursor refers.

Move

Overloaded.

MoveFirst

Overloaded.

MoveFirstMultiple

Overloaded.

If positioning the cursor fails, CurrentMultiple will contain an empty KeyValuePair(Of TKey, TValue>).

MoveFirstMultipleKey

Overloaded.

MoveLast

Overloaded.

MoveMultiple

Overloaded.

MoveMultipleKey

Overloaded.

MoveNext

Overloaded.

MoveNextDuplicate

Overloaded.

MoveNextDuplicateMultiple

Overloaded.

MoveNextDuplicateMultipleKey

Overloaded.
- **MoveNextMultiple** Overloaded.
- **MoveNextMultipleKey** Overloaded.
- **MoveNextUnique** Overloaded.
- **MoveNextUniqueMultiple** Overloaded.
- **MoveNextUniqueMultipleKey** Overloaded.
- **MovePrev** Overloaded.
- **MovePrevDuplicate** Overloaded.
- **MovePrevUnique** Overloaded.

- **Overwrite**
  Overwrite the data of the key/data pair to which the cursor refers with the specified data item.
  (Inherited from **Cursor**.)

- **Refresh** Overloaded.
- **RefreshMultiple** Overloaded.
- **RefreshMultipleKey** Overloaded.

- **ToString** Returns a **String** that represents the current **Object**.
  (Inherited from **Object**.)
### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current</td>
<td>The key/data pair at which the cursor currently points. (Inherited from <strong>Cursor</strong>.)</td>
</tr>
<tr>
<td>CurrentMultiple</td>
<td>The key and multiple data items at which the cursor currently points. (Inherited from <strong>Cursor</strong>.)</td>
</tr>
<tr>
<td>CurrentMultipleKey</td>
<td>The multiple key and data items at which the cursor currently points. (Inherited from <strong>Cursor</strong>.)</td>
</tr>
<tr>
<td>Priority</td>
<td>The cache priority for pages referenced by the cursor. (Inherited from <strong>Cursor</strong>.)</td>
</tr>
</tbody>
</table>
See Also

HashCursor Class
BerkeleyDB.Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `HashCursor` type exposes the following members.
Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add</td>
<td>Overloaded. Insert the specified key/data pair into the database, unless a key/data pair comparing equally to it already exists in the database.</td>
</tr>
<tr>
<td>AddUnique</td>
<td>Discard the cursor.</td>
</tr>
<tr>
<td>Close</td>
<td>It is possible for the Close() method to throw a DeadlockException, signaling that any enclosing transaction should be aborted. If the application is already intending to abort the transaction, this error should be ignored, and the application should proceed.</td>
</tr>
<tr>
<td>Compare</td>
<td>After Close has been called, regardless of its result, the object may not be used again.</td>
</tr>
<tr>
<td>Count</td>
<td>(Inherited from BaseCursor.) Compare this cursor's position to another's. (Inherited from BaseCursor.)</td>
</tr>
<tr>
<td>Delete</td>
<td>Returns a count of the number of data items for the key to which the cursor refers. (Inherited from BaseCursor.) Delete the key/data pair to which the cursor refers. (Inherited from Cursor.)</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Dispose</strong></td>
<td>Release the resources held by this object, and close the cursor if it's still open.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <strong>BaseCursor</strong>.)</td>
</tr>
<tr>
<td><strong>Duplicate</strong></td>
<td>Create a new cursor that uses the same transaction and locker ID as the original cursor.</td>
</tr>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <strong>Object</strong> is equal to the current <strong>Object</strong>.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>GetEnumerator</strong></td>
<td>Returns an enumerator that iterates through the <strong>Cursor</strong>.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <strong>Cursor</strong>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <strong>Type</strong> of the current instance.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>Insert</strong></td>
<td>Insert the data element as a duplicate element of the key to which the cursor refers.</td>
</tr>
<tr>
<td><strong>Move</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>MoveFirst</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>MoveFirstMultiple</strong></td>
<td>Overloaded. If positioning the cursor fails, <strong>CurrentMultiple</strong> will contain an empty <strong>KeyValuePair</strong>&lt;<strong>(Of <em>(TKey, TValue)</em>)&gt;</strong>.</td>
</tr>
<tr>
<td><strong>MoveFirstMultipleKey</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>MoveLast</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>MoveMultiple</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>MoveMultipleKey</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>MoveNext</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>MoveNextDuplicate</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>MoveNextDuplicateMultiple</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>MoveNextDuplicateMultipleKey</strong></td>
<td>Overloaded.</td>
</tr>
</tbody>
</table>
- **MoveNextMultiple** Overloaded.
- **MoveNextMultipleKey** Overloaded.
- **MoveNextUnique** Overloaded.
- **MoveNextUniqueMultiple** Overloaded.
- **MoveNextUniqueMultipleKey** Overloaded.
- **MovePrev** Overloaded.
- **MovePrevDuplicate** Overloaded.
- **MovePrevUnique** Overloaded.

**Overwrite**

Overwrite the data of the key/data pair to which the cursor refers with the specified data item.

(Inherited from **Cursor**.)

- **Refresh** Overloaded.
- **RefreshMultiple** Overloaded.
- **RefreshMultipleKey** Overloaded.

**ToString**

Returns a **String** that represents the current **Object**.

(Inherited from **Object**.)
See Also

HashCursor Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
HashCursor

::

Add Method

HashCursor Class  See Also
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add(KeyValuePair&lt;(Of &lt;(DatabaseEntry, DatabaseEntry)&gt;)&gt;))</td>
<td>Stores the key/data pair in the database. (Inherited from Cursor.)</td>
</tr>
<tr>
<td>Add(KeyValuePair&lt;(Of &lt;(DatabaseEntry, DatabaseEntry)&gt;)), Cursor...:::InsertLocation)</td>
<td>Insert the specified key/data pair into the database.</td>
</tr>
</tbody>
</table>
See Also

HashCursor Class
HashCursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Insert the specified key/data pair into the database.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

```csharp
public void Add(
    KeyValuePair<DatabaseEntry, DatabaseEntry> pair,
    Cursor:::InsertLocation loc
)
```

Visual Basic (Declaration)

```vbnet
Public Sub Add ( _
    pair As KeyValuePair(Of DatabaseEntry, DatabaseEntry), _
    loc As Cursor:::InsertLocation _
)
```

Visual C++

```cpp
public:
void Add(
    KeyValuePair<DatabaseEntry^, DatabaseEntry^> pair,
    Cursor:::InsertLocation loc
)
```

Parameters

pair

Type: `System.Collections.Generic:::KeyValuePair<Of <(DatabaseEntry, DatabaseEntry>>>`

The key/data pair to be inserted

loc

Type: `BerkeleyDB:::Cursor:::InsertLocation`

If the key already exists in the database and no duplicate sort function has been specified, specify whether the inserted data item is added as the first or the last of the data items for that key.
See Also

HashCursor Class
Add Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Insert the specified key/data pair into the database, unless a key/data pair comparing equally to it already exists in the database.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public void AddUnique(  
    KeyValuePair<DatabaseEntry, DatabaseEntry> pair
)

Visual Basic (Declaration)

Public Sub AddUnique (_
    pair As KeyValuePair(Of DatabaseEntry, DatabaseEntry) _
)

Visual C++

public:
void AddUnique(  
    KeyValuePair<DatabaseEntry^, DatabaseEntry^> pair
)

Parameters

pair
    Type: System.Collections.Generic:::KeyValuePair<Of (DatabaseEntry, DatabaseEntry)>)
    The key/data pair to be inserted
<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>BerkeleyDB::KeyExistException</td>
<td>Thrown if a matching key/data pair already exists in the database.</td>
</tr>
</tbody>
</table>
See Also

HashCursor Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Create a new cursor that uses the same transaction and locker ID as the original cursor.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public HashCursor Duplicate(
    bool keepPosition
)

Visual Basic (Declaration)

Public Function Duplicate ( _
    keepPosition As Boolean _
) As HashCursor

Visual C++

public:
    HashCursor^ Duplicate(
        bool keepPosition
    )

Parameters

keepPosition
Type: System::Boolean
If true, the newly created cursor is initialized to refer to the same position in
the database as the original cursor (if any) and hold the same locks (if any).
If false, or the original cursor does not hold a database position and locks,
the created cursor is uninitialized and will behave like a cursor newly
created by Cursor().

Return Value

A newly created cursor
Remarks

This is useful when an application is using locking and requires two or more cursors in the same thread of control.
See Also

HashCursor Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Insert the data element as a duplicate element of the key to which the cursor refers.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public void Insert(
    DatabaseEntry data,
    Cursor:::InsertLocation loc
)

Visual Basic (Declaration)

Public Sub Insert ( _
    data As DatabaseEntry, _
    loc As Cursor:::InsertLocation _
)

Visual C++

public:
void Insert(
    DatabaseEntry^ data,
    Cursor:::InsertLocation loc
)

Parameters

data
Type: BerkeleyDB:::DatabaseEntry
The data element to insert

loc
Type: BerkeleyDB:::Cursor:::InsertLocation
Specify whether to insert the data item immediately before or immediately after the cursor's current position.
See Also

HashCursor Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation
HashCursor...:::Move Method

HashCursor Class  See Also
### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Move</strong>(DatabaseEntry, Boolean)</td>
<td>Set the cursor to refer to key, and store the datum associated with the given key in Current. In the presence of duplicate key values, the first data item in the set of duplicates is stored in Current. (Inherited from Cursor.)</td>
</tr>
<tr>
<td><strong>Move</strong>(KeyValuePair&lt;Of &lt;(DatabaseEntry, DatabaseEntry)&gt;&gt;, Boolean)</td>
<td>Move the cursor to the specified key/data pair of the database. The cursor is positioned to a key/data pair if both the key and data match the values provided on the key and data parameters. (Inherited from Cursor.)</td>
</tr>
<tr>
<td><strong>Move</strong>(DatabaseEntry, Boolean, LockingInfo)</td>
<td>Set the cursor to refer to key, and store the datum associated with the given key in Current. In the presence of duplicate key values, the first data item in the set of duplicates is stored in Current. (Inherited from Cursor.)</td>
</tr>
<tr>
<td><strong>Move</strong>(KeyValuePair&lt;Of &lt;(DatabaseEntry, DatabaseEntry)&gt;&gt;, Boolean, LockingInfo)</td>
<td>Move the cursor to the specified key/data pair of the database. The cursor is positioned to a key/data pair if both the key and data match the values provided on the key and data parameters. (Inherited from Cursor.)</td>
</tr>
</tbody>
</table>
See Also

HashCursor Class
HashCursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
HashCursor::MoveFirst Method

HashCursor Class  See Also
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MoveFirst()()()</td>
<td>Set the cursor to refer to the first key/data pair of the database, and store that pair in Current. If the first key has duplicate values, the first data item in the set of duplicates is stored in Current. (Inherited from Cursor.)</td>
</tr>
<tr>
<td>MoveFirst(LockingInfo)</td>
<td>Set the cursor to refer to the first key/data pair of the database, and store that pair in Current. If the first key has duplicate values, the first data item in the set of duplicates is stored in Current. (Inherited from Cursor.)</td>
</tr>
</tbody>
</table>
See Also

HashCursor Class
HashCursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If positioning the cursor fails, `CurrentMultiple` will contain an empty `KeyValuePair<(Of <(TKey, TValue)>)>`. 
### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MoveFirstMultiple()()</td>
<td>Set the cursor to refer to the first key/data pair of the database, and store that key and as many duplicate data items that can fit in a buffer the size of one database page in <code>CurrentMultiple</code>. (Inherited from <code>Cursor</code>.)</td>
</tr>
<tr>
<td>MoveFirstMultiple(LockingInfo)</td>
<td>Set the cursor to refer to the first key/data pair of the database, and store that key and as many duplicate data items that can fit in a buffer the size of one database page in <code>CurrentMultiple</code>. (Inherited from <code>Cursor</code>.)</td>
</tr>
<tr>
<td>MoveFirstMultiple(Int32)</td>
<td>Set the cursor to refer to the first key/data pair of the database, and store that key and as many duplicate data items that can fit in a buffer the size of <code>BufferSize</code> in <code>CurrentMultiple</code>. (Inherited from <code>Cursor</code>.)</td>
</tr>
<tr>
<td>MoveFirstMultiple(Int32, LockingInfo)</td>
<td>Set the cursor to refer to the first key/data pair of the database, and store that key and as many duplicate data items that can fit in a buffer the size of <code>BufferSize</code> in <code>CurrentMultiple</code>. (Inherited from <code>Cursor</code>.)</td>
</tr>
</tbody>
</table>
See Also

HashCursor Class
HashCursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
C#  Visual Basic
Visual C++
Include Protected Members
Include Inherited Members
Berkeley DB .NET API Documentation
HashCursor...:::MoveFirstMultipleKey Method
HashCursor Class  See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MoveFirstMultipleKey()</strong>()</td>
<td>Set the cursor to refer to the first key/data pair of the database, and store that pair and as many ensuing key/data pairs that can fit in a buffer the size of one database page in <code>CurrentMultipleKey</code>. (Inherited from <code>Cursor</code>.)*</td>
</tr>
<tr>
<td><strong>MoveFirstMultipleKey(LockingInfo)</strong></td>
<td>Set the cursor to refer to the first key/data pair of the database, and store that pair and as many ensuing key/data pairs that can fit in a buffer the size of one database page in <code>CurrentMultipleKey</code>. (Inherited from <code>Cursor</code>.)*</td>
</tr>
<tr>
<td><strong>MoveFirstMultipleKey(Int32)</strong></td>
<td>Set the cursor to refer to the first key/data pair of the database, and store that pair and as many ensuing key/data pairs that can fit in a buffer the size of <code>BufferSize</code> in <code>CurrentMultipleKey</code>. (Inherited from <code>Cursor</code>.)*</td>
</tr>
<tr>
<td><strong>MoveFirstMultipleKey(Int32, LockingInfo)</strong></td>
<td>Set the cursor to refer to the first key/data pair of the database, and store that pair and as many ensuing key/data pairs that can fit in a buffer the size of <code>BufferSize</code> in <code>CurrentMultipleKey</code>. (Inherited from <code>Cursor</code>.)*</td>
</tr>
</tbody>
</table>
See Also

HashCursor Class
HashCursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
MoveLast Method

HashCursor Class  See Also
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MoveLast()()</td>
<td>Set the cursor to refer to the last key/data pair of the database, and store that pair in Current. If the last key has duplicate values, the last data item in the set of duplicates is stored in Current. (Inherited from Cursor.)</td>
</tr>
<tr>
<td>MoveLast(LockingInfo)</td>
<td>Set the cursor to refer to the last key/data pair of the database, and store that pair in Current. If the last key has duplicate values, the last data item in the set of duplicates is stored in Current. (Inherited from Cursor.)</td>
</tr>
</tbody>
</table>
See Also

HashCursor Class
HashCursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation
HashCursor..:::MoveMultiple Method

HashCursor Class  See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MoveMultiple</strong>(DatabaseEntry, Boolean)</td>
<td>Set the cursor to refer to key, and store that key and as many duplicate data items associated with the given key that can fit in a buffer the size of one database page in CurrentMultiple. (Inherited from Cursor.)</td>
</tr>
<tr>
<td><strong>MoveMultiple</strong>(KeyValuePair&lt;Of &lt;(DatabaseEntry, DatabaseEntry)&gt;, Boolean)</td>
<td>Move the cursor to the specified key/data pair of the database, and store that key/data pair and as many duplicate data items associated with the given key that can fit in a buffer the size of one database page in CurrentMultiple. The cursor is positioned to a key/data pair if both the key and data match the values provided on the key and data parameters. (Inherited from Cursor.)</td>
</tr>
<tr>
<td><strong>MoveMultiple</strong>(DatabaseEntry, Boolean, LockingInfo)</td>
<td>Set the cursor to refer to key, and store that key and as many duplicate data items associated with the given key that can fit in a buffer the size of one database page in CurrentMultiple. (Inherited from Cursor.)</td>
</tr>
<tr>
<td><strong>MoveMultiple</strong>(DatabaseEntry, Boolean, Int32)</td>
<td>Set the cursor to refer to key, and store that key and as many duplicate data items associated with the given key that can fit in a buffer the size of BufferSize in CurrentMultiple. (Inherited from Cursor.)</td>
</tr>
</tbody>
</table>
MoveMultiple(KeyValuePair<(Of (DatabaseEntry, DatabaseEntry)>, Boolean, LockingInfo))

Move the cursor to the specified key/data pair of the database, and store that key/data pair and as many duplicate data items associated with the given key that can fit in a buffer the size of one database page in CurrentMultiple. The cursor is positioned to a key/data pair if both the key and data match the values provided on the key and data parameters. (Inherited from Cursor.)

MoveMultiple(KeyValuePair<(Of (DatabaseEntry, DatabaseEntry)>, Boolean, Int32))

Move the cursor to the specified key/data pair of the database, and store that key/data pair and as many duplicate data items associated with the given key that can fit in a buffer the size of BufferSize in CurrentMultiple. The cursor is positioned to a key/data pair if both the key and data match the values provided on the key and data parameters. (Inherited from Cursor.)

MoveMultiple(DatabaseEntry, Boolean, Int32, LockingInfo)

Set the cursor to refer to key, and store that key and as many duplicate data items associated with the given key that can fit in a buffer the size of BufferSize in CurrentMultiple. (Inherited from Cursor.)

MoveMultiple(KeyValuePair<(Of (DatabaseEntry, DatabaseEntry)>, Boolean, Int32, LockingInfo))

Move the cursor to the specified key/data pair of the database, and store that key/data pair and as many duplicate data items associated with the given key that can fit in a buffer the size of BufferSize in CurrentMultiple. The cursor is positioned to a key/data pair if both
the key and data match the values provided on the key and data parameters.
(Inherited from Cursor.)
See Also

HashCursor Class
HashCursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation
HashCursor...:.MoveMultipleKey Method

HashCursor Class  See Also
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MoveMultipleKey</strong> <em>(DatabaseEntry, Boolean)</em></td>
<td>Set the cursor to refer to key, and store that key and as many ensuing key/data pairs that can fit in a buffer the size of one database page in <em>CurrentMultipleKey</em>. *(Inherited from <em>Cursor</em>.)</td>
</tr>
<tr>
<td><strong>MoveMultipleKey</strong> <em>(KeyValuePair&lt;Of &lt;(DatabaseEntry, DatabaseEntry)&gt;, Boolean)&gt;</em></td>
<td>Move the cursor to the specified key/data pair of the database, and store that key/data pair and as many ensuing key/data pairs that can fit in a buffer the size of one database page in <em>CurrentMultipleKey</em>. The cursor is positioned to a key/data pair if both the key and data match the values provided on the key and data parameters. *(Inherited from <em>Cursor</em>.)</td>
</tr>
<tr>
<td><strong>MoveMultipleKey</strong> <em>(DatabaseEntry, Boolean, LockingInfo)</em></td>
<td>Set the cursor to refer to key, and store that key and as many ensuing key/data pairs that can fit in a buffer the size of one database page in <em>CurrentMultipleKey</em>. *(Inherited from <em>Cursor</em>.)</td>
</tr>
<tr>
<td><strong>MoveMultipleKey</strong> <em>(DatabaseEntry, Boolean, Int32)</em></td>
<td>Set the cursor to refer to key, and store that key and as many ensuing key/data pairs that can fit in a buffer the size of BufferSize in <em>CurrentMultipleKey</em>.</td>
</tr>
</tbody>
</table>
MoveMultipleKey(KeyValuePair(Of (DatabaseEntry, DatabaseEntry>), Boolean, LockingInfo)

Move the cursor to the specified key/data pair of the database, and store that key/data pair and as many ensuing key/data pairs that can fit in a buffer the size of one database page in CurrentMultipleKey. The cursor is positioned to a key/data pair if both the key and data match the values provided on the key and data parameters. (Inherited from Cursor.)

MoveMultipleKey(KeyValuePair(Of (DatabaseEntry, DatabaseEntry>), Boolean, Int32)

Move the cursor to the specified key/data pair of the database, and store that key/data pair and as many ensuing key/data pairs that can fit in a buffer the size of BufferSize in CurrentMultipleKey. The cursor is positioned to a key/data pair if both the key and data match the values provided on the key and data parameters. (Inherited from Cursor.)

MoveMultipleKey(DatabaseEntry, Boolean, Int32, LockingInfo)

Set the cursor to refer to key, and store that key and as many ensuing key/data pairs that can fit in a buffer the size of BufferSize in CurrentMultipleKey. (Inherited from Cursor.)

MoveMultipleKey(KeyValuePair(Of (DatabaseEntry, DatabaseEntry>), Int32, LockingInfo)

Move the cursor to the specified key/data pair of the database, and store that key/data pair and as many ensuing key/data pairs that can fit in a buffer the size of BufferSize in
CurrentMultipleKey. The cursor is positioned to a key/data pair if both the key and data match the values provided on the key and data parameters. (Inherited from Cursor.)
See Also

HashCursor Class
HashCursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
HashCursor

MoveNext Method

HashCursor Class  See Also
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MoveNext()()</td>
<td>If the cursor is not yet initialized, MoveNext is identical to MoveFirst(). Otherwise, move the cursor to the next key/data pair of the database, and store that pair in Current. In the presence of duplicate key values, the value of Current.Key may not change. (Inherited from Cursor.)</td>
</tr>
<tr>
<td>MoveNext(LockingInfo)</td>
<td>If the cursor is not yet initialized, MoveNext is identical to MoveFirst(LockingInfo). Otherwise, move the cursor to the next key/data pair of the database, and store that pair in Current. In the presence of duplicate key values, the value of Current.Key may not change. (Inherited from Cursor.)</td>
</tr>
</tbody>
</table>
See Also

HashCursor Class
HashCursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation
HashCursor::MoveNextDuplicate Method

HashCursor Class  See Also
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MoveNextDuplicate()</strong></td>
<td>If the next key/data pair of the database is a duplicate data record for the current key/data pair, move the cursor to the next key/data pair in the database, and store that pair in <code>Current</code>. MoveNextDuplicate will return false if the next key/data pair of the database is not a duplicate data record for the current key/data pair. (Inherited from <code>Cursor</code>.)</td>
</tr>
<tr>
<td><strong>MoveNextDuplicate(LockingInfo)</strong></td>
<td>If the next key/data pair of the database is a duplicate data record for the current key/data pair, move the cursor to the next key/data pair in the database, and store that pair in <code>Current</code>. MoveNextDuplicate will return false if the next key/data pair of the database is not a duplicate data record for the current key/data pair. (Inherited from <code>Cursor</code>.)</td>
</tr>
</tbody>
</table>
See Also

HashCursor Class
HashCursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation

HashCursor::MoveNextDuplicateMultiple Method

HashCursor Class  See Also
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MoveNextDuplicateMultiple()()()</td>
<td>If the next key/data pair of the database is a duplicate data record for the current key/data pair, move the cursor to the next key/data pair in the database, and store that pair and as many duplicate data items that can fit in a buffer the size of one database page in <code>CurrentMultiple</code>. MoveNextDuplicateMultiple will return false if the next key/data pair of the database is not a duplicate data record for the current key/data pair. (Inherited from <code>Cursor</code>.)</td>
</tr>
<tr>
<td>MoveNextDuplicateMultiple(LockingInfo)</td>
<td>If the next key/data pair of the database is a duplicate data record for the current key/data pair, move the cursor to the next key/data pair in the database, and store that pair and as many duplicate data items that can fit in a buffer the size of one database page in <code>CurrentMultiple</code>. MoveNextDuplicateMultiple will return false if the next key/data pair of the database is not a duplicate data record for the current key/data pair.</td>
</tr>
</tbody>
</table>
MoveNextDuplicateMultiple(Int32)

MoveNextDuplicateMultiple(Int32, LockingInfo)

(Inherited from Cursor.)
If the next key/data pair of the database is a duplicate data record for the current key/data pair, then move cursor to the next key/data pair in the database, and store that pair and as many duplicate data items that can fit in a buffer the size of BufferSize in CurrentMultiple.
MoveNextDuplicateMultiple will return false if the next key/data pair of the database is not a duplicate data record for the current key/data pair.
(Inherited from Cursor.)
If the next key/data pair of the database is a duplicate data record for the current key/data pair, move the cursor to the next key/data pair in the database, and store that pair and as many duplicate data items that can fit in a buffer the size of BufferSize in CurrentMultiple.
MoveNextDuplicateMultiple will return false if the next key/data pair of the database is not a duplicate data record for the current key/data pair.
See Also

HashCursor Class
HashCursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MoveNextDuplicateMultipleKey()()</td>
<td>If the next key/data pair of the database is a duplicate data record for the current key/data pair, move the cursor to the next key/data pair in the database, and store that pair and as many duplicate data items that can fit in a buffer the size of one database page in <code>CurrentMultipleKey</code>. <code>MoveNextDuplicateMultipleKey</code> will return false if the next key/data pair of the database is not a duplicate data record for the current key/data pair. (Inherited from <code>Cursor</code>.)</td>
</tr>
<tr>
<td>MoveNextDuplicateMultipleKey(LockingInfo)</td>
<td>If the next key/data pair of the database is a duplicate data record for the current key/data pair, move the cursor to the next key/data pair in the database, and store that pair and as many duplicate data items that can fit in a buffer the size of one database page in <code>CurrentMultipleKey</code>. <code>MoveNextDuplicateMultipleKey</code> will return false if the next key/data pair of the database is not a duplicate data record for the current key/data pair. (Inherited from <code>Cursor</code>.)</td>
</tr>
</tbody>
</table>

MoveNextDuplicateMultipleKey(Int32)

database is a duplicate data record for the current key/data pair, move the cursor to the next key/data pair in the database, and store that pair and as many duplicate data items that can fit in a buffer the size of BufferSize in CurrentMultipleKey. MoveNextDuplicateMultipleKey will return false if the next key/data pair of the database is not a duplicate data record for the current key/data pair. (Inherited from Cursor.)

MoveNextDuplicateMultipleKey(Int32, LockingInfo)

If the next key/data pair of the database is a duplicate data record for the current key/data pair, move the cursor to the next key/data pair in the database, and store that pair and as many duplicate data items that can fit in a buffer the size of BufferSize in CurrentMultipleKey. MoveNextDuplicateMultipleKey will return false if the next key/data pair of the database is not a duplicate data record for the current key/data pair. (Inherited from Cursor.)
See Also

HashCursor Class
HashCursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
HashCursor Class  See Also
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MoveNextMultiple()</strong></td>
<td>If the cursor is not yet initialized, MoveNextMultiple is identical to MoveFirstMultiple(). Otherwise, move the cursor to the next key/data pair of the database, and store that pair and as many duplicate data items that can fit in a buffer the size of one database page in CurrentMultiple. In the presence of duplicate key values, the value of CurrentMultiple.Key may not change. (Inherited from Cursor.)</td>
</tr>
<tr>
<td><strong>MoveNextMultiple(LockingInfo)</strong></td>
<td>If the cursor is not yet initialized, MoveNextMultiple is identical to MoveFirstMultiple(LockingInfo). Otherwise, move the cursor to the next key/data pair of the database, and store that pair and as many duplicate data items that can fit in a buffer the size of one database page in CurrentMultiple. In the presence of duplicate key values, the value of CurrentMultiple.Key may not change. (Inherited from Cursor.)</td>
</tr>
<tr>
<td><strong>MoveNextMultiple(Int32)</strong></td>
<td>If the cursor is not yet initialized, MoveNextMultiple is identical to MoveFirstMultiple(Int32). Otherwise, move the cursor to the next key/data pair of the database, and store that pair and as many duplicate data items that can fit in a buffer the size of BufferSize in CurrentMultiple. In the presence of duplicate key values, the</td>
</tr>
</tbody>
</table>
value of `CurrentMultiple.Key` may not change.
(Inherited from `Cursor`.)
If the cursor is not yet initialized, `MoveNextMultiple` is identical to
`MoveFirstMultiple(Int32, LockingInfo)`. Otherwise, move the
cursor to the next key/data pair of the database, and store that pair and as
many duplicate data items that can fit in a buffer the size of BufferSize in
`CurrentMultiple`. In the presence of
duplicate key values, the value of
`CurrentMultiple.Key` may not change.
(Inherited from `Cursor`.)
See Also

HashCursor Class
HashCursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation
HashCursor...:::MoveNextMultipleKey Method

HashCursor Class  See Also
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MoveNextMultipleKey()</strong></td>
<td>If the cursor is not yet initialized, MoveNextMultipleKey is identical to MoveFirstMultipleKey(). Otherwise, move the cursor to the next key/data pair of the database, and store that pair and as many ensuing key/data pairs that can fit in a buffer the size of one database page in CurrentMultipleKey. In the presence of duplicate key values, the keys of CurrentMultipleKey may not change. (Inherited from Cursor.)</td>
</tr>
<tr>
<td><strong>MoveNextMultipleKey(LockingInfo)</strong></td>
<td>If the cursor is not yet initialized, MoveNextMultipleKey is identical to MoveFirstMultipleKey(LockingInfo). Otherwise, move the cursor to the next key/data pair of the database, and store that pair and as many ensuing key/data pairs that can fit in a buffer the size of one database page in CurrentMultipleKey. In the presence of duplicate key values, the keys of CurrentMultipleKey may not change. (Inherited from Cursor.)</td>
</tr>
<tr>
<td><strong>MoveNextMultipleKey(Int32)</strong></td>
<td>If the cursor is not yet initialized, MoveNextMultipleKey is identical to MoveFirstMultipleKey(Int32). Otherwise, move the cursor to the next key/data pair of the database, and store that pair and as many ensuing key/data pairs that can fit in a buffer the size of one database page in CurrentMultipleKey. In the presence of duplicate key values, the keys of CurrentMultipleKey may not change. (Inherited from Cursor.)</td>
</tr>
</tbody>
</table>
buffer the size of BufferSize in CurrentMultipleKey. In the presence of duplicate key values, the keys of CurrentMultipleKey may not change. (Inherited from Cursor.)

If the cursor is not yet initialized, MoveNextMultipleKey is identical to MoveFirstMultipleKey(Int32, LockingInfo). Otherwise, move the cursor to the next key/data pair of the database, and store that pair and as many ensuing key/data pairs that can fit in a buffer the size of BufferSize in CurrentMultipleKey. In the presence of duplicate key values, the keys of CurrentMultipleKey may not change. (Inherited from Cursor.)
See Also

HashCursor Class
HashCursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation
HashCursor...::MoveNextUnique Method

HashCursor Class  See Also
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MoveNextUnique()()()</td>
<td>If the cursor is not yet initialized, MoveNextUnique is identical to MoveFirst(). Otherwise, move the cursor to the next non-duplicate key in the database, and store that key and associated datum in Current. MoveNextUnique will return false if no non-duplicate key/data pairs exist after the cursor position in the database. (Inherited from Cursor.)</td>
</tr>
<tr>
<td>MoveNextUnique(LockingInfo)</td>
<td>MoveNextUnique will return false if no non-duplicate key/data pairs exist after the cursor position in the database. (Inherited from Cursor.)</td>
</tr>
</tbody>
</table>
See Also

HashCursor Class
HashCursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
HashCursor Class  See Also
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MoveNextUniqueMultiple()()</strong></td>
<td>If the cursor is not yet initialized, MoveNextUniqueMultiple is identical to <strong>MoveFirstMultiple()()</strong>. Otherwise, move the cursor to the next non-duplicate key in the database, and store that key and associated datum and as many duplicate data items that can fit in a buffer the size of one database page in <code>CurrentMultiple</code>. MoveNextUniqueMultiple will return false if no non-duplicate key/data pairs exist after the cursor position in the database. (Inherited from <code>Cursor</code>.) If the cursor is not yet initialized, MoveNextUniqueMultiple is identical to <strong>MoveFirstMultiple(LockingInfo)</strong>. Otherwise, move the cursor to the next non-duplicate key in the database, and store that key and associated datum and as many duplicate data items that can fit in a buffer the size of one database page in <code>CurrentMultiple</code>. MoveNextUniqueMultiple will return false if no non-duplicate key/data pairs exist after the cursor position in the database. (Inherited from <code>Cursor</code>.)</td>
</tr>
<tr>
<td><strong>MoveNextUniqueMultiple(LockingInfo)</strong></td>
<td></td>
</tr>
</tbody>
</table>
MoveNextUniqueMultiple is identical to MoveFirstMultiple(Int32).
Otherwise, move the cursor to the next non-duplicate key in the
database, and store that key and associated datum and as many
duplicate data items that can fit in a buffer the size of BufferSize in
CurrentMultiple.
MoveNextUniqueMultiple will return false if no non-duplicate
key/data pairs exist after the cursor position in the database.
(Inherited from Cursor.)
If the cursor is not yet initialized,
MoveNextUniqueMultiple is identical to
MoveFirstMultiple(Int32,
LockingInfo). Otherwise, move
the cursor to the next non-
duplicate key in the database, and
store that key and associated
datum and as many duplicate
data items that can fit in a buffer
the size of BufferSize in
CurrentMultiple.
MoveNextUniqueMultiple will return false if no non-duplicate
key/data pairs exist after the
cursor position in the database.
(Inherited from Cursor.)
See Also

HashCursor Class
HashCursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
C# □ Visual Basic
□ Visual C++
□ Include Protected Members
□ Include Inherited Members
Berkeley DB .NET API Documentation
HashCursor...:::MoveNextUniqueMultipleKey Method
HashCursor Class  See Also
Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MoveNextUniqueMultipleKey()</td>
<td>If the cursor is not yet initialized, MoveNextUniqueMultipleKey is identical to MoveFirstMultipleKey(). Otherwise, move the cursor to the next non-duplicate key in the database, and store that key and associated datum and as many ensuing key/data pairs that can fit in a buffer the size of one database page in CurrentMultipleKey. MoveNextUniqueMultipleKey will return false if no non-duplicate key/data pairs exist after the cursor position in the database. (Inherited from Cursor.)</td>
</tr>
<tr>
<td>MoveNextUniqueMultipleKey(LockingInfo)</td>
<td>If the cursor is not yet initialized, MoveNextUniqueMultipleKey is identical to MoveFirstMultipleKey(LockingInfo). Otherwise, move the cursor to the next non-duplicate key in the database, and store that key and associated datum and as many ensuing key/data pairs that can fit in a buffer the size of one database page in CurrentMultipleKey. MoveNextUniqueMultipleKey will return false if no non-duplicate key/data pairs exist after the cursor position in the database. (Inherited from Cursor.)</td>
</tr>
</tbody>
</table>
**MoveNextUniqueMultipleKey(Int32)**

identical to **MoveFirstMultipleKey(Int32)**. Otherwise, move the cursor to the next non-duplicate key in the database, and store that key and associated datum and as many ensuing key/data pairs that can fit in a buffer of size BufferSize in **CurrentMultipleKey**.

MoveNextUniqueMultipleKey will return false if no non-duplicate key/data pairs exist after the cursor position in the database.
(Inherited from **Cursor**.)

If the cursor is not yet initialized, **MoveNextUniqueMultipleKey** is identical to **MoveFirstMultipleKey(Int32, LockingInfo)**. Otherwise, move the cursor to the next non-duplicate key in the database, and store that key and associated datum and as many ensuing key/data pairs that can fit in a buffer of size BufferSize in **CurrentMultipleKey**.

MoveNextUniqueMultipleKey will return false if no non-duplicate key/data pairs exist after the cursor position in the database.
(Inherited from **Cursor**.)

**MoveNextUniqueMultipleKey(Int32, LockingInfo)**
See Also

HashCursor Class
HashCursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
HashCursor Class  See Also
# Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MovePrev()()</strong></td>
<td>If the cursor is not yet initialized, MovePrev is identical to <code>MoveLast()()</code>. Otherwise, move the cursor to the previous key/data pair of the database, and store that pair in <code>Current</code>. In the presence of duplicate key values, the value of <code>Current.Key</code> may not change. (Inherited from <code>Cursor</code>.)</td>
</tr>
<tr>
<td><strong>MovePrev(LockingInfo)</strong></td>
<td>If the cursor is not yet initialized, MovePrev is identical to <code>MoveLast(LockingInfo)</code>. Otherwise, move the cursor to the previous key/data pair of the database, and store that pair in <code>Current</code>. In the presence of duplicate key values, the value of <code>Current.Key</code> may not change. (Inherited from <code>Cursor</code>.)</td>
</tr>
</tbody>
</table>
See Also

HashCursor Class
HashCursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
HashCursor Class  See Also
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MovePrevDuplicate()</td>
<td>If the previous key/data pair of the database is a duplicate data record for the current key/data pair, the cursor is moved to the previous key/data pair of the database, and that pair is stored in Current. MovePrevDuplicate will return false if the previous key/data pair of the database is not a duplicate data record for the current key/data pair. (Inherited from Cursor.)</td>
</tr>
<tr>
<td>MovePrevDuplicate(LockingInfo)</td>
<td>If the previous key/data pair of the database is a duplicate data record for the current key/data pair, the cursor is moved to the previous key/data pair of the database, and that pair is stored in Current. MovePrevDuplicate will return false if the previous key/data pair of the database is not a duplicate data record for the current key/data pair. (Inherited from Cursor.)</td>
</tr>
</tbody>
</table>
See Also

HashCursor Class
HashCursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
HashCursor Class  See Also
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MovePrevUnique()()</strong></td>
<td>If the cursor is not yet initialized, MovePrevUnique is identical to MoveLast(). Otherwise, move the cursor to the previous non-duplicate key in the database, and store that key and associated datum in Current. MovePrevUnique will return false if no non-duplicate key/data pairs exist after the cursor position in the database. (Inherited from Cursor.)</td>
</tr>
<tr>
<td><strong>MovePrevUnique(LockingInfo)</strong></td>
<td>If the cursor is not yet initialized, MovePrevUnique is identical to MoveLast(LockingInfo). Otherwise, move the cursor to the previous non-duplicate key in the database, and store that key and associated datum in Current. MovePrevUnique will return false if no non-duplicate key/data pairs exist after the cursor position in the database. (Inherited from Cursor.)</td>
</tr>
</tbody>
</table>

Overload List
See Also

HashCursor Class
HashCursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
HashCursor::Refresh Method

HashCursor Class  See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Refresh()()</td>
<td>Store the key/data pair to which the cursor refers in <strong>Current</strong>.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <strong>Cursor</strong>.)</td>
</tr>
<tr>
<td>Refresh(LockingInfo)</td>
<td>Store the key/data pair to which the cursor refers in <strong>Current</strong>.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <strong>Cursor</strong>.)</td>
</tr>
</tbody>
</table>
See Also

HashCursor Class
HashCursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
HashCursor Class  See Also
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RefreshMultiple()()()</td>
<td>Store the key/data pair to which the cursor refers and as many duplicate data items that can fit in a buffer the size of one database page in CurrentMultiple. (Inherited from Cursor.)</td>
</tr>
<tr>
<td>RefreshMultiple(LockingInfo)</td>
<td>Store the key/data pair to which the cursor refers and as many duplicate data items that can fit in a buffer the size of one database page in CurrentMultiple. (Inherited from Cursor.)</td>
</tr>
<tr>
<td>RefreshMultiple(Int32)</td>
<td>Store the key/data pair to which the cursor refers and as many duplicate data items that can fit in a buffer the size of BufferSize in CurrentMultiple. (Inherited from Cursor.)</td>
</tr>
<tr>
<td>RefreshMultiple(Int32, LockingInfo)</td>
<td>Store the key/data pair to which the cursor refers and as many duplicate data items that can fit in a buffer the size of BufferSize in CurrentMultiple. (Inherited from Cursor.)</td>
</tr>
</tbody>
</table>
See Also

HashTable Class
HashTable Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
C#  Visual Basic  Visual C++  Include Protected Members  Include Inherited Members  Berkeley DB .NET API Documentation  HashCursor:::.RefreshMultipleKey Method  HashCursor Class  See Also
### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="#">RefreshMultipleKey()()()</a></td>
<td>Store the key/data pair to which the cursor refers and as many ensuing key/data pairs that can fit in a buffer the size of one database page in CurrentMultipleKey. (Inherited from Cursor.)</td>
</tr>
<tr>
<td><a href="#">RefreshMultipleKey(LockingInfo)</a></td>
<td>Store the key/data pair to which the cursor refers and as many ensuing key/data pairs that can fit in a buffer the size of one database page in CurrentMultipleKey. (Inherited from Cursor.)</td>
</tr>
<tr>
<td><a href="#">RefreshMultipleKey(Int32)</a></td>
<td>Store the key/data pair to which the cursor refers and as many ensuing key/data pairs that can fit in a buffer the size of BufferSize in CurrentMultipleKey. (Inherited from Cursor.)</td>
</tr>
<tr>
<td><a href="#">RefreshMultipleKey(Int32, LockingInfo)</a></td>
<td>Store the key/data pair to which the cursor refers and as many ensuing key/data pairs that can fit in a buffer the size of BufferSize in CurrentMultipleKey. (Inherited from Cursor.)</td>
</tr>
</tbody>
</table>
See Also

HashCursor Class
HashCursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `HashCursor` type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| Current            | The key/data pair at which the cursor currently points.  
(Inherited from [Cursor](#).)                                                                                                                   |
| CurrentMultiple    | The key and multiple data items at which the cursor currently points.  
(Inherited from [Cursor](#).)                                                                                                                   |
| CurrentMultipleKey | The multiple key and data items at which the cursor currently points.  
(Inherited from [Cursor](#).)                                                                                                                   |
| Priority           | The cache priority for pages referenced by the cursor.  
(Inherited from [Cursor](#).)                                                                                                                   |
See Also

HashCursor Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
A class representing a HashDatabase. The Hash format is an extensible, dynamic hashing scheme.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public class HashDatabase : Database

Visual Basic (Declaration)

Public Class HashDatabase Inherits Database

Visual C++

public ref class HashDatabase : public Database
Inheritance Hierarchy

System::Object
   BerkeleyDB::BaseDatabase
      BerkeleyDB::Database
         BerkeleyDB::HashDatabase
See Also

HashDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `HashDatabase` type exposes the following members.
Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Close</td>
<td>Overloaded.</td>
</tr>
</tbody>
</table>

Although closing a database will close any open cursors, it is recommended that applications explicitly close all their Cursor objects before closing the database. The reason why is that when the cursor is explicitly closed, the memory allocated for it is reclaimed; however, this will not happen if you close a database while cursors are still opened.

The same rule, for the same reasons, hold true for Transaction objects. Simply make sure you resolve all your transaction objects before closing your database handle.

Because key/data pairs are cached in memory, applications should make a point to always either close database handles or sync their data to disk (using `Sync()` before exiting, to ensure that any data cached in main memory are reflected in the underlying file system.

When called on a database that is the primary database for a secondary index, the primary database should be closed only after all secondary indices referencing it have been closed.

When multiple threads are using the object concurrently, only a single thread may call the Close method.

The object may not be accessed again after
Close is called, regardless of its outcome.

- **Cursor**
  Overloaded.
- **Delete**
  Overloaded.
- **Dispose**
  Release the resources held by this object, and close the database if it's still open.
  (Inherited from `BaseDatabase`.)
- **Equals**
  Determines whether the specified `Object` is equal to the current `Object`.
  (Inherited from `Object`.)
- **Exists**
  Overloaded.
- **FastStats**
  Among other things, this method makes it possible for applications to request key and record counts without incurring the performance penalty of traversing the entire database.

  The statistical information is described by the `BTreeStats`, `HashStats`, `QueueStats`, and `RecnoStats` classes.

- **Get**
  Overloaded.
- **GetBoth**
  Overloaded.
- **GetBothMultiple**
  Overloaded.
- **GetHashCode**
  Serves as a hash function for a particular type.
  (Inherited from `Object`.)
- **GetMultiple**
  Overloaded.
- **GetType**
  Gets the `Type` of the current instance.
  (Inherited from `Object`.)
- **Join**
  Create a specialized join cursor for use in performing equality or natural joins on secondary indices.
  (Inherited from `Database`.)
- **Open**
  Overloaded.
  Overloaded.
  The statistical information is described by the
PrintFastStats

PrintFastStats

BTreeStats, HashStats, QueueStats, and RecnoStats classes.

Overloaded.

The statistical information is described by the BTreeStats, HashStats, QueueStats, and RecnoStats classes.

Overloaded.

Put

Put

If the database supports duplicates, add the new data value at the end of the duplicate set. If the database supports sorted duplicates, the new data value is inserted at the correct sorted location.

PutNoDuplicate

PutNoDuplicate

Overloaded.

PutNoOverwrite

PutNoOverwrite

Overloaded.

Stats

Stats

The statistical information is described by BTreeStats.

Sync

Sync

Flush any cached information to disk.

(Inherited from BaseDatabase.)

Returns a String that represents the current Object.

(Inherited from Object.)

Overloaded.

Truncate

Truncate

When called on a database configured with secondary indices, Truncate will truncate the primary database and all secondary indices. A count of the records discarded from the primary database is returned.

TruncateUnusedPages

TruncateUnusedPages

Overloaded.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AutoCommit</strong></td>
<td>If true, all database modification operations based on this object will be transactionally protected. (Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
<tr>
<td><strong>CacheSize</strong></td>
<td>The size of the shared memory buffer pool -- that is, the cache. (Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
<tr>
<td><strong>Compare</strong></td>
<td>The Hash key comparison function. The comparison function is called whenever it is necessary to compare a key specified by the application with a key currently stored in the tree.</td>
</tr>
<tr>
<td><strong>Creation</strong></td>
<td>The CreatePolicy with which this database was opened. (Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
<tr>
<td><strong>DatabaseName</strong></td>
<td>The name of this database, if it has one. (Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
<tr>
<td><strong>DoChecksum</strong></td>
<td>If true, do checksum verification of pages read into the cache from the backing filestore. (Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
<tr>
<td><strong>DupCompare</strong></td>
<td>The duplicate data item comparison function.</td>
</tr>
<tr>
<td><strong>Duplicates</strong></td>
<td>Whether the insertion of duplicate data items in the database is permitted, and whether duplicates items are sorted.</td>
</tr>
<tr>
<td><strong>EncryptAlgorithm</strong></td>
<td>The algorithm used by the Berkeley DB library to perform encryption and decryption. (Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
<tr>
<td><strong>Encrypted</strong></td>
<td>If true, encrypt all data stored in the database. (Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
<tr>
<td><strong>Endianness</strong></td>
<td>The database byte order. (Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
<tr>
<td><strong>ErrorFeedback</strong></td>
<td>The mechanism for reporting detailed error messages to the application.</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-----------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ErrorPrefix</td>
<td>The prefix string that appears before error messages issued by Berkeley DB.</td>
</tr>
<tr>
<td>Feedback</td>
<td>Monitor progress within long running operations.</td>
</tr>
<tr>
<td>FileName</td>
<td>The filename of this database, if it has one.</td>
</tr>
<tr>
<td>FillFactor</td>
<td>The desired density within the hash table. If true, the object is free-threaded; that is,</td>
</tr>
<tr>
<td></td>
<td>concurrently usable by multiple threads in the address space.</td>
</tr>
<tr>
<td>FreeThreaded</td>
<td>A user-defined hash function; if no hash function is specified, a default hash function is</td>
</tr>
<tr>
<td></td>
<td>used. If true, the object references a physical file supporting multiple databases.</td>
</tr>
<tr>
<td>InHostOrder</td>
<td>If true, the underlying database files were created on an architecture of the same byte order</td>
</tr>
<tr>
<td></td>
<td>as the current one. This information may be used to determine whether application data</td>
</tr>
<tr>
<td></td>
<td>needs to be adjusted for this architecture or not.</td>
</tr>
<tr>
<td>NoMMap</td>
<td>If true, this database is not mapped into process memory.</td>
</tr>
<tr>
<td>NonDurableTxns</td>
<td>If true, Berkeley DB will not write log records for this database.</td>
</tr>
<tr>
<td>Pagesize</td>
<td>The database's current page size.</td>
</tr>
</tbody>
</table>

(Inherited from [BaseDatabase](#)).
| **Priority** | object. (Inherited from BaseDatabase.) |
| **ReadOnly** | If true, this database has been opened for reading only. Any attempt to modify items in the database will fail, regardless of the actual permissions of any underlying files. (Inherited from BaseDatabase.) |
| **ReadUncommitted** | If true, this database supports transactional read operations with degree 1 isolation. Read operations on the database may request the return of modified but not yet committed data. (Inherited from BaseDatabase.) |
| **TableSize** | An estimate of the final size of the hash table. |
| **Transactional** | If true, this database has been opened in a transactional mode. (Inherited from BaseDatabase.) |
| **Truncated** | If true, the underlying file was physically truncated upon open, discarding all previous databases it might have held. (Inherited from BaseDatabase.) |
| **Type** | The type of the underlying access method (and file format). This value may be used to determine the type of the database after an Open(String, DatabaseConfig). (Inherited from BaseDatabase.) |
| **UseMVCC** | If true, the database was opened with support for multiversion concurrency control. (Inherited from BaseDatabase.) |
See Also

HashDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `HashDatabase` type exposes the following members.
**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Close</td>
<td>Overloaded. Although closing a database will close any open cursors, it is recommended that applications explicitly close all their Cursor objects before closing the database. The reason why is that when the cursor is explicitly closed, the memory allocated for it is reclaimed; however, this will not happen if you close a database while cursors are still opened. The same rule, for the same reasons, hold true for Transaction objects. Simply make sure you resolve all your transaction objects before closing your database handle. Because key/data pairs are cached in memory, applications should make a point to always either close database handles or sync their data to disk (using Sync()()) before exiting, to ensure that any data cached in main memory are reflected in the underlying file system. When called on a database that is the primary database for a secondary index, the primary database should be closed only after all secondary indices referencing it have been closed. When multiple threads are using the object concurrently, only a single thread may call the Close method. The object may not be accessed again after</td>
</tr>
</tbody>
</table>
Close is called, regardless of its outcome.

- **Cursor** Overloaded.
- **Delete** Overloaded.
- **Dispose**
  Release the resources held by this object, and close the database if it's still open.
  (Inherited from **BaseDatabase**.)
- **Equals**
  Determines whether the specified **Object** is equal to the current **Object**.
  (Inherited from **Object**.)
- **Exists** Overloaded.
- **FastStats**
  Among other things, this method makes it possible for applications to request key and record counts without incurring the performance penalty of traversing the entire database.
  The statistical information is described by the **BTreeStats**, **HashStats**, **QueueStats**, and **RecnoStats** classes.
- **Get** Overloaded.
- **GetBoth** Overloaded.
- **GetBothMultiple** Overloaded.
- **GetHashCode**
  Serves as a hash function for a particular type.
  (Inherited from **Object**.)
- **GetMultiple** Overloaded.
- **GetType**
  Gets the **Type** of the current instance.
  (Inherited from **Object**.)
- **Join**
  Create a specialized join cursor for use in performing equality or natural joins on secondary indices.
  (Inherited from **Database**.)
- **Open** Overloaded.
  Overloaded.
  The statistical information is described by the
PrintFastStats

BTreeStats, HashStats, QueueStats, and RecnoStats classes.

Overloaded.

PrintStats

The statistical information is described by the BTreeStats, HashStats, QueueStats, and RecnoStats classes.

Overloaded.

Put

If the database supports duplicates, add the new data value at the end of the duplicate set. If the database supports sorted duplicates, the new data value is inserted at the correct sorted location.

PutNoDuplicate

Overloaded.

PutNoOverwrite

Overloaded.

Stats

The statistical information is described by BTreeStats.

Sync

Flush any cached information to disk. (Inherited from BaseDatabase.)

Returns a String that represents the current Object.

ToString

(Inherited from Object.)

Overloaded.

When called on a database configured with secondary indices, Truncate will truncate the primary database and all secondary indices. A count of the records discarded from the primary database is returned.

Truncate

TruncateUnusedPages

Overloaded.
See Also

HashDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Although closing a database will close any open cursors, it is recommended that applications explicitly close all their Cursor objects before closing the database. The reason why is that when the cursor is explicitly closed, the memory allocated for it is reclaimed; however, this will not happen if you close a database while cursors are still opened.

The same rule, for the same reasons, hold true for Transaction objects. Simply make sure you resolve all your transaction objects before closing your database handle.

Because key/data pairs are cached in memory, applications should make a point to always either close database handles or sync their data to disk (using Sync() before exiting, to ensure that any data cached in main memory are reflected in the underlying file system.

When called on a database that is the primary database for a secondary index, the primary database should be closed only after all secondary indices referencing it have been closed.

When multiple threads are using the object concurrently, only a single thread may call the Close method.

The object may not be accessed again after Close is called, regardless of its outcome.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Close()</td>
<td>Flush any cached database information to disk, close any open Cursor objects, free any allocated resources, and close any underlying files.</td>
</tr>
<tr>
<td></td>
<td><em>(Inherited from BaseDatabase.)</em></td>
</tr>
<tr>
<td>Close(Boolean)</td>
<td>Optionally flush any cached database information to disk, close any open Cursor objects, free any allocated resources, and close any underlying files.</td>
</tr>
<tr>
<td></td>
<td><em>(Inherited from BaseDatabase.)</em></td>
</tr>
</tbody>
</table>
See Also

HashDatabase Class
HashDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
HashDatabase.....Cursor Method

HashDatabase Class   See Also
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cursor()()</td>
<td>Create a database cursor.</td>
</tr>
<tr>
<td>Cursor(CursorConfig)</td>
<td>Create a database cursor with the given configuration.</td>
</tr>
<tr>
<td>Cursor(Transaction)</td>
<td>Create a transactionally protected database cursor.</td>
</tr>
<tr>
<td>Cursor(CursorConfig, Transaction)</td>
<td>Create a transactionally protected database cursor with the given configuration.</td>
</tr>
</tbody>
</table>
See Also

HashDatabase Class
HashDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Create a database cursor.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public HashCursor Cursor()

Visual Basic (Declaration)

Public Function Cursor As HashCursor

Visual C++

public: HashCursor^ Cursor()

Return Value

A newly created cursor
See Also

HashDatabase Class
Cursor Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Create a database cursor with the given configuration.

**Namespace:** [BerkeleyDB](https://www.berkleydb.org/)

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public HashCursor Cursor(
    CursorConfig cfg
)

Visual Basic (Declaration)

Public Function Cursor (_
    cfg As CursorConfig _
) As HashCursor

Visual C++

public:
HashCursor^ Cursor(
    CursorConfig^ cfg
)

Parameters

cfg
    Type: BerkeleyDB::CursorConfig
    The configuration properties for the cursor.

Return Value

A newly created cursor
See Also

HashDatabase Class
Cursor Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Create a transactionally protected database cursor.

**Namespace:** [BerkeleyDB](https://example.com/BerkeleyDB)

**Assembly:** `libdb_dotnet48` (in `libdb_dotnet48.dll`) Version: 4.8.24.0
Syntax

**C#**

```csharp
public HashCursor Cursor(
    Transaction txn
)
```

**Visual Basic (Declaration)**

```vbnet
Public Function Cursor (
    _
    txn As Transaction _
) As HashCursor
```

**Visual C++**

```cpp
public:
    HashCursor^ Cursor(
        Transaction^ txn
    )
```

**Parameters**

txn

Type: **BerkeleyDB::Transaction**

The transaction context in which the cursor may be used.

**Return Value**

A newly created cursor
See Also

HashDatabase Class
Cursor Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Create a transactionally protected database cursor with the given configuration.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

```csharp
public HashCursor Cursor(
    CursorConfig cfg,
    Transaction txn
)
```

Visual Basic (Declaration)

```vbnet
Public Function Cursor (_
    cfg As CursorConfig, _
    txn As Transaction _
) As HashCursor
```

Visual C++

```cpp
public: 
    HashCursor^ Cursor(
        CursorConfig^ cfg,
        Transaction^ txn
    )
```

Parameters

cfg
Type: BerkeleyDB::CursorConfig
The configuration properties for the cursor.

txn
Type: BerkeleyDB::Transaction
The transaction context in which the cursor may be used.

Return Value

A newly created cursor
See Also

HashDatabase Class
Cursor Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation

HashDatabase Class  See Also
### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delete(DatabaseEntry)</td>
<td>Remove key/data pairs from the database. The key/data pair associated with key is discarded from the database. In the presence of duplicate key values, all records associated with the designated key will be discarded. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>Delete(DatabaseEntry, Transaction)</td>
<td>Remove key/data pairs from the database. The key/data pair associated with key is discarded from the database. In the presence of duplicate key values, all records associated with the designated key will be discarded. (Inherited from BaseDatabase.)</td>
</tr>
</tbody>
</table>
See Also

HashDatabase Class
HashDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation

HashDatabase Exists Method

HashDatabase Class  See Also
**Overload List**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exists(DatabaseEntry)</td>
<td>Check whether key appears in the database. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>Exists(DatabaseEntry, Transaction)</td>
<td>Check whether key appears in the database. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>Exists(DatabaseEntry, Transaction, LockingInfo)</td>
<td>Check whether key appears in the database. (Inherited from BaseDatabase.)</td>
</tr>
</tbody>
</table>
See Also

HashDatabase Class
HashDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Among other things, this method makes it possible for applications to request key and record counts without incurring the performance penalty of traversing the entire database.

The statistical information is described by the `BTreeStats`, `HashStats`, `QueueStats`, and `RecnoStats` classes.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>FastStats()()()</td>
<td>Return the database statistical information which does not require traversal of the database.</td>
<td></td>
</tr>
<tr>
<td>FastStats(Transaction)</td>
<td>Return the database statistical information which does not require traversal of the database.</td>
<td></td>
</tr>
<tr>
<td>FastStats(Transaction, Isolation)</td>
<td>Return the database statistical information which does not require traversal of the database.</td>
<td></td>
</tr>
</tbody>
</table>
See Also

HashDatabase Class
HashDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Return the database statistical information which does not require traversal of the database.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public HashStats FastStats()

Visual Basic (Declaration)

Public Function FastStats As HashStats

Visual C++

public:
HashStats^ FastStats()

Return Value

The database statistical information which does not require traversal of the database.
See Also

HashDatabase Class
FastStats Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Return the database statistical information which does not require traversal of the database.

**Namespace:**  [BerkeleyDB](#)  
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll)  
**Version:**  4.8.24.0
Syntax

C#

```csharp
public HashStats FastStats(
    Transaction txn
)
```

Visual Basic (Declaration)

```vbnet
Public Function FastStats ( _
    txn As Transaction _
) As HashStats
```

Visual C++

```cpp
public: 
HashStats^ FastStats(
    Transaction^ txn
)
```

Parameters

txn

Type: `BerkeleyDB::Transaction`

If the operation is part of an application-specified transaction, txn is a Transaction object returned from `BeginTransaction()`; if the operation is part of a Berkeley DB Concurrent Data Store group, txn is a handle returned from `BeginCDSGroup()`; otherwise null.

Return Value

The database statistical information which does not require traversal of the database.
See Also

HashDatabase Class
FastStats Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
HashDatabase Class  See Also

Return the database statistical information which does not require traversal of the database.

Namespace:  BerkeleyDB
Syntax

C#

public HashStats FastStats(
    Transaction txn,
    Isolation isoDegree
)

Visual Basic (Declaration)

Public Function FastStats (_
    txn As Transaction,
    isoDegree As Isolation _
) As HashStats

Visual C++

public:
    HashStats^ FastStats(
        Transaction^ txn,
        Isolation isoDegree
    )

Parameters

txn
    Type: BerkeleyDB::Transaction
    If the operation is part of an application-specified transaction, txn is a Transaction object returned from BeginTransaction(); if the operation is part of a Berkeley DB Concurrent Data Store group, txn is a handle returned from BeginCDSGroup(); otherwise null.

isoDegree
    Type: BerkeleyDB::Isolation
    The level of isolation for database reads. DEGREE_ONE will be silently ignored for databases which did not specify ReadUncommitted.
**Return Value**

The database statistical information which does not require traversal of the database.
See Also

HashDatabase Class
FastStats Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
HashDatabase

See Also

HashDatabase Class
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Get(DatabaseEntry)</td>
<td>Retrieve a key/data pair from the database. In the presence of duplicate key values, Get will return the first data item for key. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>Get(DatabaseEntry, Transaction)</td>
<td>Retrieve a key/data pair from the database. In the presence of duplicate key values, Get will return the first data item for key. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>Get(DatabaseEntry, Transaction, LockingInfo)</td>
<td>Retrieve a key/data pair from the database. In the presence of duplicate key values, Get will return the first data item for key. (Inherited from BaseDatabase.)</td>
</tr>
</tbody>
</table>
See Also

HashDatabase Class
HashDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
HashDatabase Class  See Also
### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetBoth(DatabaseEntry, DatabaseEntry)</td>
<td>Retrieve a key/data pair from the database which matches key and data.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
<tr>
<td>GetBoth(DatabaseEntry, DatabaseEntry, Transaction)</td>
<td>Retrieve a key/data pair from the database which matches key and data.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
<tr>
<td>GetBoth(DatabaseEntry, DatabaseEntry, Transaction, LockingInfo)</td>
<td>Retrieve a key/data pair from the database which matches key and data.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
</tbody>
</table>
See Also

HashDatabase Class
HashDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
HashDatabase Class  See Also
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetBothMultiple(DatabaseEntry, DatabaseEntry)</td>
<td>If a key/data pair in the database matches key and data, return the key and all duplicate data items. (Inherited from Database.)</td>
</tr>
<tr>
<td>GetBothMultiple(DatabaseEntry, DatabaseEntry, Int32)</td>
<td>If a key/data pair in the database matches key and data, return the key and all duplicate data items. (Inherited from Database.)</td>
</tr>
<tr>
<td>GetBothMultiple(DatabaseEntry, DatabaseEntry, Int32, Transaction)</td>
<td>If a key/data pair in the database matches key and data, return the key and all duplicate data items. (Inherited from Database.)</td>
</tr>
<tr>
<td>GetBothMultiple(DatabaseEntry, DatabaseEntry, Int32, Transaction, LockingInfo)</td>
<td>If a key/data pair in the database matches key and data, return the key and all duplicate data items. (Inherited from Database.)</td>
</tr>
</tbody>
</table>
See Also

HashDatabase Class
HashDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation
HashDatabase.....GetMultiple Method

HashDatabase Class  See Also
# Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>GetMultiple(DatabaseEntry)</code></td>
<td>Retrieve a key and all duplicate data items from the database. (Inherited from <a href="#">Database</a>.)</td>
</tr>
<tr>
<td><code>GetMultiple(DatabaseEntry, Int32)</code></td>
<td>Retrieve a key and all duplicate data items from the database. (Inherited from <a href="#">Database</a>.)</td>
</tr>
<tr>
<td><code>GetMultiple(DatabaseEntry, Int32, Transaction)</code></td>
<td>Retrieve a key and all duplicate data items from the database. (Inherited from <a href="#">Database</a>.)</td>
</tr>
<tr>
<td><code>GetMultiple(DatabaseEntry, Int32, Transaction, LockingInfo)</code></td>
<td>Retrieve a key and all duplicate data items from the database. (Inherited from <a href="#">Database</a>.)</td>
</tr>
</tbody>
</table>
See Also

HashDatabase Class
HashDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Open(String, HashDatabaseConfig)</code></td>
<td>Instantiate a new HashDatabase object and open the database represented by Filename.</td>
</tr>
<tr>
<td><code>Open(String, HashDatabaseConfig, Transaction)</code></td>
<td>Instantiate a new HashDatabase object and open the database represented by Filename.</td>
</tr>
<tr>
<td><code>Open(String, String, HashDatabaseConfig)</code></td>
<td>Instantiate a new HashDatabase object and open the database represented by Filename and DatabaseName.</td>
</tr>
<tr>
<td><code>Open(String, String, HashDatabaseConfig, Transaction)</code></td>
<td>Instantiate a new HashDatabase object and open the database represented by Filename and DatabaseName.</td>
</tr>
</tbody>
</table>
See Also

HashDatabase Class
HashDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
HashDatabase Class  See Also

Instantiate a new HashDatabase object and open the database represented by Filename.

Namespace: BerkeleyDB
## Syntax

### C#

```csharp
public static HashDatabase Open(
    string Filename,
    HashDatabaseConfig cfg)
```

### Visual Basic (Declaration)

```vbnet
Public Shared Function Open (_
    Filename As String, _
    cfg As HashDatabaseConfig _
) As HashDatabase
```

### Visual C++

```cpp
public:
static HashDatabase^ Open(
    String^ Filename, 
    HashDatabaseConfig^ cfg)
```

## Parameters

**Filename**
- Type: `System::::String`
- The name of an underlying file that will be used to back the database. In-memory databases never intended to be preserved on disk may be created by setting this parameter to null.

**cfg**
- Type: `BerkeleyDB::::HashDatabaseConfig`
- The database's configuration

## Return Value
A new, open database object
Remarks

If Filename is null, the database is strictly temporary and cannot be opened by any other thread of control, thus the database can only be accessed by sharing the single database object that created it, in circumstances where doing so is safe.

If AutoCommit is set, the operation will be implicitly transaction protected. Note that transactionally protected operations on a database object requires the object itself be transactionally protected during its open.
See Also

HashDatabase Class
Open Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
HashDatabase Class  See Also

Instantiate a new HashDatabase object and open the database represented by Filename.

Namespace: BerkeleyDB
### Syntax

#### C#

```csharp
public static HashDatabase Open(
    string Filename,
    HashDatabaseConfig cfg,
    Transaction txn
)
```

#### Visual Basic (Declaration)

```vbnet
Public Shared Function Open (_
    Filename As String, _
    cfg As HashDatabaseConfig, _
    txn As Transaction _
) As HashDatabase
```

#### Visual C++

```cpp
public:
static HashDatabase^ Open(
    String^ Filename, 
    HashDatabaseConfig^ cfg, 
    Transaction^ txn 
)
```

### Parameters

**Filename**

Type: `System::String`

The name of an underlying file that will be used to back the database. In-memory databases never intended to be preserved on disk may be created by setting this parameter to null.

**cfg**

Type: `BerkeleyDB::HashDatabaseConfig`

The database's configuration
txn

Type: BerkeleyDB::::Transaction
If the operation is part of an application-specified transaction, txn is a Transaction object returned from BeginTransaction(); if the operation is part of a Berkeley DB Concurrent Data Store group, txn is a handle returned from BeginCDSGroup(); otherwise null.

**Return Value**

A new, open database object
Remarks

If Filename is null, the database is strictly temporary and cannot be opened by any other thread of control, thus the database can only be accessed by sharing the single database object that created it, in circumstances where doing so is safe.

If txn is null, but AutoCommit is set, the operation will be implicitly transaction protected. Note that transactionally protected operations on a database object requires the object itself be transactionally protected during its open. Also note that the transaction must be committed before the object is closed.
See Also

HashDatabase Class
Open Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Instantiate a new HashDatabase object and open the database represented by Filename and DatabaseName.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
### Syntax

#### C#

```
public static HashDatabase Open(
    string Filename,
    string DatabaseName,
    HashDatabaseConfig cfg
)
```

#### Visual Basic (Declaration)

```
Public Shared Function Open (_
    Filename As String, _
    DatabaseName As String, _
    cfg As HashDatabaseConfig _
) As HashDatabase
```

#### Visual C++

```
public:
    static HashDatabase^ Open(
        String^ Filename, _
        String^ DatabaseName, _
        HashDatabaseConfig^ cfg
    )
```

### Parameters

**Filename**

Type: `System::::String`

The name of an underlying file that will be used to back the database. In-memory databases never intended to be preserved on disk may be created by setting this parameter to null.

**DatabaseName**

Type: `System::::String`

This parameter allows applications to have multiple databases in a single file. Although no DatabaseName needs to be specified, it is an error to
attempt to open a second database in a file that was not initially created using a database name.

cfg
  Type: BerkeleyDB::::HashDatabaseConfig
  The database's configuration

**Return Value**

A new, open database object
Remarks

If both Filename and DatabaseName are null, the database is strictly temporary and cannot be opened by any other thread of control, thus the database can only be accessed by sharing the single database object that created it, in circumstances where doing so is safe. If Filename is null and DatabaseName is non-null, the database can be opened by other threads of control and will be replicated to client sites in any replication group.

If AutoCommit is set, the operation will be implicitly transaction protected. Note that transactionally protected operations on a database object requires the object itself be transactionally protected during its open.
See Also

HashDatabase Class
Open Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation

HashDatabase...::Open Method (String, String, HashDatabaseConfig, Transaction)

HashDatabase Class  See Also

Instantiate a new HashDatabase object and open the database represented by Filename and DatabaseName.

Namespace:  BerkeleyDB
Syntax

C#

```csharp
public static HashDatabase Open(
    string Filename,
    string DatabaseName,
    HashDatabaseConfig cfg,
    Transaction txn
)
```

Visual Basic (Declaration)

```vbnet
Public Shared Function Open ( _
    Filename As String, _
    DatabaseName As String, _
    cfg As HashDatabaseConfig, _
    txn As Transaction _
) As HashDatabase
```

Visual C++

```cpp
public:
    static HashDatabase^ Open(
        String^ Filename,
        String^ DatabaseName,
        HashDatabaseConfig^ cfg,
        Transaction^ txn
    )
```

Parameters

Filename
Type: `System::String`
The name of an underlying file that will be used to back the database. In-memory databases never intended to be preserved on disk may be created by setting this parameter to null.

DatabaseName
Type: `System::String`
This parameter allows applications to have multiple databases in a single file. Although no DatabaseName needs to be specified, it is an error to attempt to open a second database in a file that was not initially created using a database name.

cfg
Type: BerkeleyDB::HashDatabaseConfig
The database's configuration

txn
Type: BerkeleyDB::Transaction
If the operation is part of an application-specified transaction, txn is a Transaction object returned from BeginTransaction(); if the operation is part of a Berkeley DB Concurrent Data Store group, txn is a handle returned from BeginCDSGroup(); otherwise null.

**Return Value**

A new, open database object
Remarks

If both Filename and DatabaseName are null, the database is strictly temporary and cannot be opened by any other thread of control, thus the database can only be accessed by sharing the single database object that created it, in circumstances where doing so is safe. If Filename is null and DatabaseName is non-null, the database can be opened by other threads of control and will be replicated to client sites in any replication group.

If txn is null, but AutoCommit is set, the operation will be implicitly transaction protected. Note that transactionally protected operations on a database object requires the object itself be transactionally protected during its open. Also note that the transaction must be committed before the object is closed.
See Also

HashDatabase Class
Open Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
HashDatabase Class  See Also

The statistical information is described by the BTreeStats, HashStats, QueueStats, and RecnoStats classes.
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>PrintFastStats()</code></td>
<td>Display the database statistical information which does not require traversal of the database. (Inherited from <code>BaseDatabase</code>.)</td>
</tr>
<tr>
<td><code>PrintFastStats(Boolean)</code></td>
<td>Display the database statistical information which does not require traversal of the database. (Inherited from <code>BaseDatabase</code>.)</td>
</tr>
</tbody>
</table>
See Also

HashDatabase Class
HashDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The statistical information is described by the `BTreeStats`, `HashStats`, `QueueStats`, and `RecnoStats` classes.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PrintStats()()</td>
<td>Display the database statistical information.</td>
</tr>
<tr>
<td>(Inherited from BaseDatabase.)</td>
<td></td>
</tr>
<tr>
<td>PrintStats(Boolean)</td>
<td>Display the database statistical information.</td>
</tr>
<tr>
<td>(Inherited from BaseDatabase.)</td>
<td></td>
</tr>
</tbody>
</table>
See Also

HashDatabase Class
HashDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If the database supports duplicates, add the new data value at the end of the duplicate set. If the database supports sorted duplicates, the new data value is inserted at the correct sorted location.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Put(DatabaseEntry, DatabaseEntry)</td>
<td>Store the key/data pair in the database, replacing any previously existing key if duplicates are disallowed, or adding a duplicate data item if duplicates are allowed. (Inherited from Database.)</td>
</tr>
<tr>
<td>Put(DatabaseEntry, DatabaseEntry, Transaction)</td>
<td>Store the key/data pair in the database, replacing any previously existing key if duplicates are disallowed, or adding a duplicate data item if duplicates are allowed. (Inherited from Database.)</td>
</tr>
</tbody>
</table>
See Also

HashDatabase Class
HashDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation
HashDatabase::PutNoDuplicate Method

HashDatabase Class  See Also
# Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PutNoDuplicate(DatabaseEntry, DatabaseEntry)</td>
<td>Store the key/data pair in the database only if it does not already appear in the database.</td>
</tr>
<tr>
<td>PutNoDuplicate(DatabaseEntry, DatabaseEntry, Transaction)</td>
<td>Store the key/data pair in the database only if it does not already appear in the database.</td>
</tr>
</tbody>
</table>
See Also

HashDatabase Class
HashDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Store the key/data pair in the database only if it does not already appear in the database.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#  
public void PutNoDuplicate(
    DatabaseEntry key,
    DatabaseEntry data
)

Visual Basic (Declaration)  
Public Sub PutNoDuplicate (_
    key As DatabaseEntry, _
    data As DatabaseEntry _
)

Visual C++  
public:
void PutNoDuplicate(
    DatabaseEntry^ key,
    DatabaseEntry^ data
)

Parameters

key  
Type: BerkeleyDB::DatabaseEntry  
The key to store in the database

data  
Type: BerkeleyDB::DatabaseEntry  
The data item to store in the database
See Also

HashDatabase Class
PutNoDuplicate Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
C#  Visual Basic  Visual C++  
Berkeley DB .NET API Documentation  
HashDatabase...::PutNoDuplicate Method (DatabaseEntry, DatabaseEntry, Transaction)  
HashDatabase Class  See Also  

Store the key/data pair in the database only if it does not already appear in the database.  

Namespace:  BerkeleyDB  
**Syntax**

**C#**

```csharp
public void PutNoDuplicate(
    DatabaseEntry key,
    DatabaseEntry data,
    Transaction txn
)
```

**Visual Basic (Declaration)**

```vbnet
Public Sub PutNoDuplicate (_
    key As DatabaseEntry, _
    data As DatabaseEntry, _
    txn As Transaction _
)
```

**Visual C++**

```cpp
public:
void PutNoDuplicate(
    DatabaseEntry^ key,
    DatabaseEntry^ data,
    Transaction^ txn
)
```

**Parameters**

- **key**
  
  Type: `BerkeleyDB::DatabaseEntry`
  
  The key to store in the database

- **data**

  Type: `BerkeleyDB::DatabaseEntry`
  
  The data item to store in the database

- **txn**

  Type: `BerkeleyDB::Transaction`
If the operation is part of an application-specified transaction, txn is a Transaction object returned from `BeginTransaction()`; if the operation is part of a Berkeley DB Concurrent Data Store group, txn is a handle returned from `BeginCDSGroup()`; otherwise null.
See Also

HashDatabase Class
PutNoDuplicate Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation
HashDatabase....: PutNoOverwrite Method

HashDatabase Class  See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>PutNoOverwrite(DatabaseEntry, DatabaseEntry)</code></td>
<td>Store the key/data pair in the database, only if the key does not already appear in the database. (Inherited from <code>Database</code>.)</td>
</tr>
<tr>
<td><code>PutNoOverwrite(DatabaseEntry, DatabaseEntry, Transaction)</code></td>
<td>Store the key/data pair in the database, only if the key does not already appear in the database. (Inherited from <code>Database</code>.)</td>
</tr>
</tbody>
</table>
See Also

HashDatabase Class
HashDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The statistical information is described by `BTreeStats`.
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stats()()()</td>
<td>Return the database statistical information for this database.</td>
</tr>
<tr>
<td>Stats(Transaction)</td>
<td>Return the database statistical information for this database.</td>
</tr>
<tr>
<td>Stats(Transaction, Isolation)</td>
<td>Return the database statistical information for this database.</td>
</tr>
</tbody>
</table>
See Also

HashDatabase Class
HashDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
HashDatabase Class  See Also

Return the database statistical information for this database.

**Namespace:**  BerkeleyDB

**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public HashStats Stats()

Visual Basic (Declaration)

Public Function Stats As HashStats

Visual C++

public:
HashStats^ Stats()

Return Value

Database statistical information.
See Also

HashDatabase Class
Stats Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Return the database statistical information for this database.

**Namespace:**  [BerkeleyDB](#)

**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
## Syntax

### C#

```csharp
public HashStats Stats(
    Transaction txn
)
```

### Visual Basic (Declaration)

```vbnet
Public Function Stats (_
    txn As Transaction _
) As HashStats
```

### Visual C++

```cpp
public:
    HashStats^ Stats(
        Transaction^ txn
    )
```

## Parameters

- **txn**
  
  Type: `BerkeleyDB..::.Transaction`
  
  If the operation is part of an application-specified transaction, `txn` is a `Transaction` object returned from `BeginTransaction()`, if the operation is part of a Berkeley DB Concurrent Data Store group, `txn` is a handle returned from `BeginCDSGroup()`, otherwise null.

## Return Value

Database statistical information.
See Also

HashDatabase Class
Stats Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
HashDatabase Class  See Also

Return the database statistical information for this database.

Namespace:  BerkeleyDB
Syntax

C#

public HashStats Stats(
    Transaction txn,
    Isolation isoDegree
)

Visual Basic (Declaration)

Public Function Stats ( _
    txn As Transaction, _
    isoDegree As Isolation _
) As HashStats

Visual C++

public:
    HashStats Stats(
        Transaction^ txn,
        Isolation isoDegree
    )

Parameters

txn
Type: BerkeleyDB:::::Transaction
If the operation is part of an application-specified transaction, txn is a Transaction object returned from BeginTransaction(); if the operation is part of a Berkeley DB Concurrent Data Store group, txn is a handle returned from BeginCDSGroup(); otherwise null.

isoDegree
Type: BerkeleyDB:::::Isolation
The level of isolation for database reads. DEGREE_ONE will be silently ignored for databases which did not specify ReadUncommitted.
Return Value

Database statistical information.
See Also

HashDatabase Class
Stats Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
When called on a database configured with secondary indices, Truncate will truncate the primary database and all secondary indices. A count of the records discarded from the primary database is returned.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Empty the database, discarding all records it contains.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
<tr>
<td>Truncate()()</td>
<td>Empty the database, discarding all records it contains.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
<tr>
<td>Truncate(Transaction)</td>
<td></td>
</tr>
</tbody>
</table>
See Also

HashDatabase Class
HashDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation
HashDatabase....:TruncateUnusedPages Method

HashDatabase Class  See Also
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TruncateUnusedPages()()()</td>
<td>Return pages to the filesystem that are already free and at the end of the file.</td>
</tr>
<tr>
<td>TruncateUnusedPages(Transaction)</td>
<td>Return pages to the filesystem that are already free and at the end of the file.</td>
</tr>
</tbody>
</table>
See Also

HashDatabase Class
HashDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
HashDatabase Class  See Also

Return pages to the filesystem that are already free and at the end of the file.

Namespace:  BerkeleyDB
### Syntax

**C#**

```csharp
public uint TruncateUnusedPages()
```

**Visual Basic (Declaration)**

```vbnet
Public Function TruncateUnusedPages As UInteger
```

**Visual C++**

```cpp
public:
unsigned int TruncateUnusedPages()
```

### Return Value

The number of database pages returned to the filesystem
See Also

HashDatabase Class
TruncateUnusedPages Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Return pages to the filesystem that are already free and at the end of the file.

**Namespace:** BerkeleyDB
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

**C#**

```csharp
public uint TruncateUnusedPages(
    Transaction txn
)
```

**Visual Basic (Declaration)**

```vbnet
Public Function TruncateUnusedPages ( _
    txn As Transaction _
) As UInteger
```

**Visual C++**

```cpp
public:
    unsigned int TruncateUnusedPages(
        Transaction^ txn
    )
```

**Parameters**

txn

Type: BerkeleyDB::Transaction

If the operation is part of an application-specified transaction, txn is a Transaction object returned from BeginTransaction(); if the operation is part of a Berkeley DB Concurrent Data Store group, txn is a handle returned from BeginCDSGroup(); otherwise null.

**Return Value**

The number of database pages returned to the filesystem
See Also

HashDatabase Class
TruncateUnusedPages Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The HashDatabase type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AutoCommit</strong></td>
<td>If true, all database modification operations based on this object will be transactionally protected. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>CacheSize</strong></td>
<td>The size of the shared memory buffer pool -- that is, the cache. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>Compare</strong></td>
<td>The Hash key comparison function. The comparison function is called whenever it is necessary to compare a key specified by the application with a key currently stored in the tree. The CreatePolicy with which this database was opened. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>Creation</strong></td>
<td>The CreatePolicy with which this database was opened. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>DatabaseName</strong></td>
<td>The name of this database, if it has one. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>DoChecksum</strong></td>
<td>If true, do checksum verification of pages read into the cache from the backing filestore. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>DupCompare</strong></td>
<td>The duplicate data item comparison function.</td>
</tr>
<tr>
<td><strong>Duplicates</strong></td>
<td>Whether the insertion of duplicate data items in the database is permitted, and whether duplicates items are sorted.</td>
</tr>
<tr>
<td><strong>EncryptAlgorithm</strong></td>
<td>The algorithm used by the Berkeley DB library to perform encryption and decryption. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>Encrypted</strong></td>
<td>If true, encrypt all data stored in the database. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>Endianness</strong></td>
<td>The database byte order. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>ErrorFeedback</strong></td>
<td>The mechanism for reporting detailed error messages to the application.</td>
</tr>
</tbody>
</table>
ErrorPrefix

The prefix string that appears before error messages issued by Berkeley DB.
(Inherited from BaseDatabase.)

Feedback

Monitor progress within long running operations.
(Inherited from BaseDatabase.)

FileName

The filename of this database, if it has one.
(Inherited from BaseDatabase.)

FillFactor

The desired density within the hash table.
If true, the object is free-threaded; that is, concurrently usable by multiple threads in the address space.
(Inherited from BaseDatabase.)

FreeThreaded

A user-defined hash function; if no hash function is specified, a default hash function is used.
If true, the object references a physical file supporting multiple databases.
(Inherited from BaseDatabase.)

HasMultiple

If true, the underlying database files were created on an architecture of the same byte order as the current one. This information may be used to determine whether application data needs to be adjusted for this architecture or not.
(Inherited from BaseDatabase.)

InHostOrder

If true, this database is not mapped into process memory.

NoMMap

See MMapSize for further information.
(Inherited from BaseDatabase.)

NonDurableTxns

If true, Berkeley DB will not write log records for this database.
(Inherited from BaseDatabase.)

Pagesize

The database's current page size.
(Inherited from BaseDatabase.)

The cache priority for pages referenced by this
<table>
<thead>
<tr>
<th>Priority</th>
<th>object. (Inherited from BaseDatabase.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ReadOnly</td>
<td>If true, this database has been opened for reading only. Any attempt to modify items in the database will fail, regardless of the actual permissions of any underlying files. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>ReadUncommitted</td>
<td>If true, this database supports transactional read operations with degree 1 isolation. Read operations on the database may request the return of modified but not yet committed data. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>TableSize</td>
<td>An estimate of the final size of the hash table.</td>
</tr>
<tr>
<td>Transactional</td>
<td>If true, this database has been opened in a transactional mode. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>Truncated</td>
<td>If true, the underlying file was physically truncated upon open, discarding all previous databases it might have held. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>Type</td>
<td>The type of the underlying access method (and file format). This value may be used to determine the type of the database after an Open(String, DatabaseConfig). (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>UseMVCC</td>
<td>If true, the database was opened with support for multiversion concurrency control. (Inherited from BaseDatabase.)</td>
</tr>
</tbody>
</table>
See Also

HashDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The Hash key comparison function. The comparison function is called whenever it is necessary to compare a key specified by the application with a key currently stored in the tree.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public EntryComparisonDelegate Compare { get; private set; }

Visual Basic (Declaration)

Public Property Compare As EntryComparisonDelegate

Visual C++

public:
property EntryComparisonDelegate^ Compare {
    EntryComparisonDelegate^ get ();
    void set (EntryComparisonDelegate^ value);
}
See Also

HashDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The duplicate data item comparison function.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public EntryComparisonDelegate DupCompare { get; private set; }

Visual Basic (Declaration)

Public Property DupCompare As EntryComparisonDelegate

Visual C++

public:
property EntryComparisonDelegate^ DupCompare {
    EntryComparisonDelegate^ get ()
    void set (EntryComparisonDelegate^ value);
}
See Also

HashDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Whether the insertion of duplicate data items in the database is permitted, and whether duplicates items are sorted.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public DuplicatesPolicy Duplicates { get; }

Visual Basic (Declaration)

Public ReadOnly Property Duplicates As DuplicatesPolicy

Visual C++

public:
property DuplicatesPolicy Duplicates {
    DuplicatesPolicy get ();
}


See Also

HashDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The desired density within the hash table.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public uint FillFactor { get; }

Visual Basic (Declaration)

Public ReadOnly Property FillFactor AsUInteger

Visual C++

public:
property unsigned int FillFactor {
    unsigned int get ();
}

See Also

HashDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
A user-defined hash function; if no hash function is specified, a default hash function is used.

Namespace:  BerkeleyDB
Syntax

C#

public HashFunctionDelegate HashFunction { get; private set; }

Visual Basic (Declaration)

Public Property HashFunction As HashFunctionDelegate

Visual C++

public:
property HashFunctionDelegate^ HashFunction {
    HashFunctionDelegate^ get ();
    void set (HashFunctionDelegate^ value);
}
See Also

HashDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
An estimate of the final size of the hash table.

**Namespace:** [BerkeleyDB](https://www自觉.doc)
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public uint TableSize { get; }

Visual Basic (Declaration)

Public ReadOnly Property TableSize AsUInteger

Visual C++

public:
property unsigned int TableSize {
    unsigned int get ();
}
See Also

HashDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
A class representing configuration parameters for HashDatabase

**Namespace:** BerkeleyDB
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public class HashDatabaseConfig : DatabaseConfig

Visual Basic (Declaration)

Public Class HashDatabaseConfig
    Inherits DatabaseConfig

Visual C++

public ref class HashDatabaseConfig : public DatabaseConfig
Inheritance Hierarchy

System::Object
BerkeleyDB::DatabaseConfig
   BerkeleyDB::HashDatabaseConfig
See Also

HashDatabaseConfig Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `HashDatabaseConfig` type exposes the following members.
# Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HashDatabaseConfig</td>
<td>Instantiate a new HashDatabaseConfig object</td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <strong>Object</strong> is equal to the current <strong>Object</strong>. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <strong>Object</strong>)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <strong>Type</strong> of the current instance. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>SetEncryption</strong></td>
<td>Set the password and algorithm used by the Berkeley DB library to perform encryption and decryption. (Inherited from <strong>DatabaseConfig</strong>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a <strong>String</strong> that represents the current <strong>Object</strong>. (Inherited from <strong>Object</strong>.)</td>
</tr>
</tbody>
</table>
### Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AutoCommit</strong></td>
<td>Enclose the open call within a transaction. If the call succeeds, the open operation will be recoverable and all subsequent database modification operations based on this handle will be transactionally protected. If the call fails, no database will have been created. (Inherited from <code>DatabaseConfig</code>)</td>
</tr>
<tr>
<td><strong>ByteOrder</strong></td>
<td>The byte order for integers in the stored database metadata. The host byte order of the machine where the Berkeley DB library was compiled is the default value. (Inherited from <code>DatabaseConfig</code>)</td>
</tr>
<tr>
<td><strong>CacheSize</strong></td>
<td>The size of the shared memory buffer pool -- that is, the cache. (Inherited from <code>DatabaseConfig</code>)</td>
</tr>
<tr>
<td><strong>Creation</strong></td>
<td>The policy for how to handle database creation.</td>
</tr>
<tr>
<td><strong>DoChecksum</strong></td>
<td>If true, do checksum verification of pages read into the cache from the backing filestore. (Inherited from <code>DatabaseConfig</code>)</td>
</tr>
<tr>
<td><strong>DuplicateCompare</strong></td>
<td>The duplicate data item comparison function.</td>
</tr>
<tr>
<td><strong>Duplicates</strong></td>
<td>Policy for duplicate data items in the database; that is, insertion when the key of the key/data pair being inserted already exists in the database will be successful. The Berkeley DB environment within which to create a database. If null, the database will be created stand-alone; that is, it is not part of any Berkeley DB environment. (Inherited from <code>DatabaseConfig</code>)</td>
</tr>
<tr>
<td><strong>Env</strong></td>
<td>The Berkeley DB environment within which to create a database. If null, the database will be created stand-alone; that is, it is not part of any Berkeley DB environment. (Inherited from <code>DatabaseConfig</code>)</td>
</tr>
<tr>
<td><strong>ErrorFeedback</strong></td>
<td>The mechanism for reporting error messages to the application. (Inherited from <code>DatabaseConfig</code>)</td>
</tr>
<tr>
<td>Setting</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>ErrorPrefix</strong></td>
<td>The prefix string that appears before error messages issued by Berkeley DB.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td><strong>Feedback</strong></td>
<td>Cause the database object to be free-threaded; that is, concurrently usable by multiple threads in the address space.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td><strong>FreeThreaded</strong></td>
<td>The Hash key comparison function.</td>
</tr>
<tr>
<td><strong>HashComparison</strong></td>
<td>A user-defined hash function; if no hash function is specified, a default hash function is used.</td>
</tr>
<tr>
<td><strong>HashFunction</strong></td>
<td>Do not map this database into process memory.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td><strong>NoMMap</strong></td>
<td>If true, Berkeley DB will not write log records for this database.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td><strong>NonDurableTxns</strong></td>
<td>The cache priority for pages referenced by the database.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td><strong>Priority</strong></td>
<td>Open the database for reading only. Any attempt to modify items in the database will fail, regardless of the actual permissions of any underlying files.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td><strong>ReadOnly</strong></td>
<td>Support transactional read operations with degree 1 isolation.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td><strong>ReadUncommitted</strong></td>
<td>Physically truncate the underlying file, discarding all previous databases it might have held.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td><strong>Truncate</strong></td>
<td>Open the database with support for multiversion concurrency control.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td><strong>UseMVCC</strong></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>EncryptAlgorithm</td>
<td>The algorithm used to perform encryption and decryption.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td>EncryptionPassword</td>
<td>The password used to perform encryption and decryption.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td>FillFactor</td>
<td>The desired density within the hash table. If no value is specified, the</td>
</tr>
<tr>
<td></td>
<td>fill factor will be selected dynamically as pages are filled.</td>
</tr>
<tr>
<td>PageSize</td>
<td>The size of the pages used to hold items in the database, in bytes.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td>TableSize</td>
<td>An estimate of the final size of the hash table.</td>
</tr>
</tbody>
</table>
See Also

HashDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Instantiate a new HashDatabaseConfig object

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public HashDatabaseConfig();

Visual Basic (Declaration)

Public Sub New

Visual C++

public:
HashDatabaseConfig();
See Also

HashDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The **HashDatabaseConfig** type exposes the following members.
## Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AutoCommit</td>
<td>Enclose the open call within a transaction. If the call succeeds, the open operation will be recoverable and all subsequent database modification operations based on this handle will be transactionally protected. If the call fails, no database will have been created. (Inherited from <code>DatabaseConfig</code>.)</td>
</tr>
<tr>
<td>ByteOrder</td>
<td>The byte order for integers in the stored database metadata. The host byte order of the machine where the Berkeley DB library was compiled is the default value. (Inherited from <code>DatabaseConfig</code>.)</td>
</tr>
<tr>
<td>CacheSize</td>
<td>The size of the shared memory buffer pool -- that is, the cache. (Inherited from <code>DatabaseConfig</code>.)</td>
</tr>
<tr>
<td>Creation</td>
<td>The policy for how to handle database creation.</td>
</tr>
<tr>
<td>DoChecksum</td>
<td>If true, do checksum verification of pages read into the cache from the backing filestore. (Inherited from <code>DatabaseConfig</code>.)</td>
</tr>
<tr>
<td>DuplicateCompare</td>
<td>The duplicate data item comparison function.</td>
</tr>
<tr>
<td>Duplicates</td>
<td>Policy for duplicate data items in the database; that is, insertion when the key of the key/data pair being inserted already exists in the database will be successful.</td>
</tr>
<tr>
<td>Env</td>
<td>The Berkeley DB environment within which to create a database. If null, the database will be created stand-alone; that is, it is not part of any Berkeley DB environment. (Inherited from <code>DatabaseConfig</code>.)</td>
</tr>
<tr>
<td>ErrorFeedback</td>
<td>The mechanism for reporting error messages to the application. (Inherited from <code>DatabaseConfig</code>.)</td>
</tr>
<tr>
<td><strong>ErrorPrefix</strong></td>
<td>The prefix string that appears before error messages issued by Berkeley DB. (Inherited from <a href="#">DatabaseConfig</a>).</td>
</tr>
<tr>
<td><strong>Feedback</strong></td>
<td>Cause the database object to be free-threaded; that is, concurrently usable by multiple threads in the address space. (Inherited from <a href="#">DatabaseConfig</a>).</td>
</tr>
<tr>
<td><strong>FreeThreaded</strong></td>
<td>The Hash key comparison function.</td>
</tr>
<tr>
<td><strong>HashComparison</strong></td>
<td>A user-defined hash function; if no hash function is specified, a default hash function is used.</td>
</tr>
<tr>
<td><strong>HashFunction</strong></td>
<td>Do not map this database into process memory. (Inherited from <a href="#">DatabaseConfig</a>).</td>
</tr>
<tr>
<td><strong>NoMMap</strong></td>
<td>If true, Berkeley DB will not write log records for this database. (Inherited from <a href="#">DatabaseConfig</a>).</td>
</tr>
<tr>
<td><strong>NonDurableTxns</strong></td>
<td>The cache priority for pages referenced by the database. (Inherited from <a href="#">DatabaseConfig</a>).</td>
</tr>
<tr>
<td><strong>Priority</strong></td>
<td>Open the database for reading only. Any attempt to modify items in the database will fail, regardless of the actual permissions of any underlying files. (Inherited from <a href="#">DatabaseConfig</a>).</td>
</tr>
<tr>
<td><strong>ReadOnly</strong></td>
<td>Support transactional read operations with degree 1 isolation. (Inherited from <a href="#">DatabaseConfig</a>).</td>
</tr>
<tr>
<td><strong>ReadUncommitted</strong></td>
<td>Physically truncate the underlying file, discarding all previous databases it might have held. (Inherited from <a href="#">DatabaseConfig</a>).</td>
</tr>
<tr>
<td><strong>Truncate</strong></td>
<td>Open the database with support for multiversion concurrency control. (Inherited from <a href="#">DatabaseConfig</a>).</td>
</tr>
<tr>
<td><strong>UseMVCC</strong></td>
<td></td>
</tr>
</tbody>
</table>
See Also

HashDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The policy for how to handle database creation.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public CreatePolicy Creation

Visual Basic (Declaration)

Public Creation As CreatePolicy

Visual C++

public:
CreatePolicy Creation
Remarks

If the database does not already exist and NEVER is set, Open(String, HashDatabaseConfig) will fail.
See Also

HashDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The duplicate data item comparison function.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public EntryComparisonDelegate DuplicateCompare

Visual Basic (Declaration)

Public DuplicateCompare As EntryComparisonDelegate

Visual C++

public:
EntryComparisonDelegate^ DuplicateCompare
Remarks

The comparison function is called whenever it is necessary to compare a data item specified by the application with a data item currently stored in the database. Setting DuplicateCompare implies setting Duplicates to SORTED.

If no comparison function is specified, the data items are compared lexically, with shorter data items collating before longer data items.

If the database already exists when Open(String, HashDatabaseConfig) is called, the delegate must be the same as that historically used to create the database or corruption can occur.
See Also

HashDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Policy for duplicate data items in the database; that is, insertion when the key of the key/data pair being inserted already exists in the database will be successful.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

**C#**

public [DuplicatesPolicy] Duplicates

**Visual Basic (Declaration)**

Public Duplicates As [DuplicatesPolicy]

**Visual C++**

public:
[DuplicatesPolicy] Duplicates
Remarks

The ordering of duplicates in the database for **UNSORTED** is determined by the order of insertion, unless the ordering is otherwise specified by use of a cursor operation or a duplicate sort function. The ordering of duplicates in the database for **SORTED** is determined by the duplicate comparison function. If the application does not specify a comparison function using `DuplicateCompare`, a default lexical comparison will be used.

**SORTED** is preferred to **UNSORTED** for performance reasons. **UNSORTED** should only be used by applications wanting to order duplicate data items manually.

If the database already exists, the value of Duplicates must be the same as the existing database or an error will be returned.
See Also

HashDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The Hash key comparison function.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

**C#**

public EntryComparisonDelegate HashComparison

**Visual Basic (Declaration)**

Public HashComparison As EntryComparisonDelegate

**Visual C++**

public:
EntryComparisonDelegate^ HashComparison
Remarks

The comparison function is called whenever it is necessary to compare a key specified by the application with a key currently stored in the tree.

If no comparison function is specified, the keys are compared lexically, with shorter keys collating before longer keys.

If the database already exists, the comparison function must be the same as that historically used to create the database or corruption can occur.
See Also

HashDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation
HashDatabaseConfig::HashFunction Field
HashDatabaseConfig Class  See Also

A user-defined hash function; if no hash function is specified, a default hash function is used.

Namespace: BerkeleyDB
Syntax

**C#**

public HashFunctionDelegate HashFunction

**Visual Basic (Declaration)**

Public HashFunction As HashFunctionDelegate

**Visual C++**

public: HashFunctionDelegate^ HashFunction
Remarks

Because no hash function performs equally well on all possible data, the user may find that the built-in hash function performs poorly with a particular data set.

If the database already exists, HashFunction must be the same as that historically used to create the database or corruption can occur.
See Also

HashDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The **HashDatabaseConfig** type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>SetEncryption</strong></td>
<td>Set the password and algorithm used by the Berkeley DB library to perform encryption and decryption. (Inherited from <a href="#">DatabaseConfig</a>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a <a href="#">String</a> that represents the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
</tbody>
</table>
See Also

HashDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The **HashDatabaseConfig** type exposes the following members.
# Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EncryptAlgorithm</td>
<td>The algorithm used to perform encryption and decryption.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">DatabaseConfig</a>.)</td>
</tr>
<tr>
<td>EncryptionPassword</td>
<td>The password used to perform encryption and decryption.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">DatabaseConfig</a>.)</td>
</tr>
<tr>
<td>FillFactor</td>
<td>The desired density within the hash table. If no value is specified, the</td>
</tr>
<tr>
<td></td>
<td>fill factor will be selected dynamically as pages are filled.</td>
</tr>
<tr>
<td>PageSize</td>
<td>The size of the pages used to hold items in the database, in bytes.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">DatabaseConfig</a>.)</td>
</tr>
<tr>
<td>TableSize</td>
<td>An estimate of the final size of the hash table.</td>
</tr>
</tbody>
</table>
See Also

HashDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The desired density within the hash table. If no value is specified, the fill factor will be selected dynamically as pages are filled.

**Namespace:** BerkeleyDB
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public uint FillFactor { get; set; }

Visual Basic (Declaration)

Public Property FillFactor AsUInteger

Visual C++

public:
property unsigned int FillFactor {
    unsigned int get();
    void set (unsigned int value);
}
Remarks

The density is an approximation of the number of keys allowed to accumulate in any one bucket, determining when the hash table grows or shrinks. If you know the average sizes of the keys and data in your data set, setting the fill factor can enhance performance. A reasonable rule computing fill factor is to set it to the following:

\[
\frac{\text{pagesize} - 32}{\text{average_key_size} + \text{average_data_size} + 8}
\]

If the database already exists, this setting will be ignored.
See Also

HashDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
An estimate of the final size of the hash table.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public uint TableSize { get; set; }

Visual Basic (Declaration)

Public Property TableSize AsUInteger

Visual C++

public:
property unsigned int TableSize {
    unsigned int get ();
    void set (unsigned int value);
}
Remarks

In order for the estimate to be used when creating the database, FillFactor must also be set. If the estimate or fill factor are not set or are set too low, hash tables will still expand gracefully as keys are entered, although a slight performance degradation may be noticed.

If the database already exists, this setting will be ignored.
See Also

HashDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The application-specified hash function.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public delegate uint HashFunctionDelegate(
    byte[] data
)

Visual Basic (Declaration)

Public Delegate Function HashFunctionDelegate ( _
    data As Byte() _
) AsUInteger

Visual C++

public delegate unsigned int HashFunctionDelegate(
    array<unsigned char>^ data
)

Parameters

data

    Type: array< System::Byte >[](0)
    A byte string representing a key in the database

Return Value

The hashed value of data
See Also

BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Statistical information about a HashDatabase

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public class HashStats

Visual Basic (Declaration)

Public Class HashStats

Visual C++

public ref class HashStats
Inheritance Hierarchy

System..:::Object
BerkeleyDB..:::.HashStats
See Also

HashStats Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The **HashStats** type exposes the following members.
### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a>.</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type.</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <a href="#">Type</a> of the current instance.</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a <a href="#">String</a> that represents the current <a href="#">Object</a>.</td>
</tr>
</tbody>
</table>
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BigPages</td>
<td>Number of big key/data pages.</td>
</tr>
<tr>
<td>BigPagesFreeBytes</td>
<td>Bytes free on big item pages.</td>
</tr>
<tr>
<td>BucketPagesFreeBytes</td>
<td>Bytes free on bucket pages.</td>
</tr>
<tr>
<td>DuplicatePages</td>
<td>Number of dup pages.</td>
</tr>
<tr>
<td>DuplicatePagesFreeBytes</td>
<td>Bytes free on duplicate pages.</td>
</tr>
<tr>
<td>FillFactor</td>
<td>Fill factor specified at create.</td>
</tr>
<tr>
<td>FreePages</td>
<td>Pages on the free list.</td>
</tr>
<tr>
<td>MagicNumber</td>
<td>Magic number.</td>
</tr>
<tr>
<td>MetadataFlags</td>
<td>Metadata flags.</td>
</tr>
<tr>
<td>nData</td>
<td>Number of data items.</td>
</tr>
<tr>
<td>nHashBuckets</td>
<td>Number of hash buckets.</td>
</tr>
<tr>
<td>nKeys</td>
<td>Number of unique keys.</td>
</tr>
<tr>
<td>nPages</td>
<td>Page count.</td>
</tr>
<tr>
<td>OverflowPages</td>
<td>Number of overflow pages.</td>
</tr>
<tr>
<td>OverflowPagesFreeBytes</td>
<td>Bytes free on ovfl pages.</td>
</tr>
<tr>
<td>PageSize</td>
<td>Page size.</td>
</tr>
<tr>
<td>Version</td>
<td>Version number.</td>
</tr>
</tbody>
</table>
See Also

HashStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The HashStats type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a> is equal to the current <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <a href="https://docs.microsoft.com/en-us/dotnet/api/system.type">Type</a> of the current instance. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a <a href="https://docs.microsoft.com/en-us/dotnet/api/system.string">String</a> that represents the current <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
</tbody>
</table>
See Also

HashStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The **HashStats** type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BigPages</td>
<td>Number of big key/data pages.</td>
</tr>
<tr>
<td>BigPagesFreeBytes</td>
<td>Bytes free on big item pages.</td>
</tr>
<tr>
<td>BucketPagesFreeBytes</td>
<td>Bytes free on bucket pages.</td>
</tr>
<tr>
<td>DuplicatePages</td>
<td>Number of dup pages.</td>
</tr>
<tr>
<td>DuplicatePagesFreeBytes</td>
<td>Bytes free on duplicate pages.</td>
</tr>
<tr>
<td>FillFactor</td>
<td>Fill factor specified at create.</td>
</tr>
<tr>
<td>FreePages</td>
<td>Pages on the free list.</td>
</tr>
<tr>
<td>MagicNumber</td>
<td>Magic number.</td>
</tr>
<tr>
<td>MetadataFlags</td>
<td>Metadata flags.</td>
</tr>
<tr>
<td>nData</td>
<td>Number of data items.</td>
</tr>
<tr>
<td>nHashBuckets</td>
<td>Number of hash buckets.</td>
</tr>
<tr>
<td>nKeys</td>
<td>Number of unique keys.</td>
</tr>
<tr>
<td>nPages</td>
<td>Page count.</td>
</tr>
<tr>
<td>OverflowPages</td>
<td>Number of overflow pages.</td>
</tr>
<tr>
<td>OverflowPagesFreeBytes</td>
<td>Bytes free on ovfl pages.</td>
</tr>
<tr>
<td>PageSize</td>
<td>Page size.</td>
</tr>
<tr>
<td>Version</td>
<td>Version number.</td>
</tr>
</tbody>
</table>
See Also

\texttt{HashStats Class}\n\texttt{BerkeleyDB Namespace}

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Number of big key/data pages.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public uint BigPages { get; }

Visual Basic (Declaration)

Public ReadOnly Property BigPages As UInteger

Visual C++

public:
property unsigned int BigPages {
    unsigned int get ();
}
See Also

HashStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Bytes free on big item pages.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
**Syntax**

**C#**

```csharp
public ulong BigPagesFreeBytes { get; }
```

**Visual Basic (Declaration)**

```vbnet
Public Readonly Property BigPagesFreeBytes As ULong
```

**Visual C++**

```cpp
public:
property unsigned long long BigPagesFreeBytes {
    unsigned long long get ();
}
```
See Also

HashStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
BucketPagesFreeBytes Property
HashStats Class  See Also

Bytes free on bucket pages.

Namespace:  BerkeleyDB
Syntax

C#

public ulong BucketPagesFreeBytes { get; }

Visual Basic (Declaration)

Public ReadOnly Property BucketPagesFreeBytes As ULong

Visual C++

public:
property unsigned long long BucketPagesFreeBytes {
    unsigned long long get ();
}

See Also

HashStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
HashStats..:.DuplicatePages Property

See Also

Number of dup pages.

Namespace: BerkeleyDB
Syntax

C#

class

public uint DuplicatePages { get; }  

Visual Basic (Declaration)

Public ReadOnly Property DuplicatePages As UInteger

Visual C++

public:

property unsigned int DuplicatePages {
    unsigned int get ();
}
See Also

HashStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Bytes free on duplicate pages.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

```csharp
public ulong DuplicatePagesFreeBytes { get; }
```

Visual Basic (Declaration)

```vbnet
Public ReadOnly Property DuplicatePagesFreeBytes As ULong
```

Visual C++

```cpp
public:
property unsigned long long DuplicatePagesFreeBytes {
    unsigned long long get ();
}
```
See Also

HashStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Fill factor specified at create.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public uint FillFactor { get; }

Visual Basic (Declaration)

Public ReadOnly Property FillFactor AsUInteger

Visual C++

public:
property unsigned int FillFactor {
    unsigned int get ();
}

See Also

HashStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
HashStats Class  See Also

Pages on the free list.

Namespace:  BerkeleyDB
Syntax

C#

```csharp
public uint FreePages { get; }
```

Visual Basic (Declaration)

```vbnet
Public ReadOnly Property FreePages As UInteger
```

Visual C++

```c++
public:
property unsigned int FreePages {
    unsigned int get ();
}
```
See Also

HashStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
HashStats Class  See Also

Magic number.

**Namespace:**  BerkeleyDB

**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public uint MagicNumber { get; }

Visual Basic (Declaration)

Public ReadOnly Property MagicNumber As UInteger

Visual C++

public:
property unsigned int MagicNumber {
    unsigned int get ();
}

See Also

HashStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Metadata flags.

**Namespace:**  BerkeleyDB  
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll)  
Version: 4.8.24.0
Syntax

C#

public uint MetadataFlags { get; }

Visual Basic (Declaration)

Public ReadOnly Property MetadataFlags AsUInteger

Visual C++

public:
property unsigned int MetadataFlags {
    unsigned int get ();
}

See Also

HashStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
HashStats Class  See Also

Number of data items.

Namespace:  BerkeleyDB
Syntax

C#

public uint nData { get; }

Visual Basic (Declaration)

Public ReadOnly Property nData As UInteger

Visual C++

public:
property unsigned int nData {
    unsigned int get();
}
See Also

HashStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Number of hash buckets.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public uint nHashBuckets { get; }

Visual Basic (Declaration)

Public ReadOnly Property nHashBuckets AsUInteger

Visual C++

public:
property unsigned int nHashBuckets {
    unsigned int get ();
}
See Also

HashStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
HashStats..:::.nKeys Property

Number of unique keys.

Namespace: BerkeleyDB
Syntax

C#

public uint nKeys { get; }

Visual Basic (Declaration)

Public ReadOnly Property nKeys AsUInteger

Visual C++

public:
property unsigned int nKeys {
    unsigned int get ();
}

See Also

HashStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
HashStats Class  See Also

Page count.

Namespace:  BerkeleyDB
Syntax

C#

```csharp
public uint nPages { get; }
```

Visual Basic (Declaration)

```vbnet
Public ReadOnly Property nPages AsUInteger
```

Visual C++

```c++
public:
property unsigned int nPages {
    unsigned int get();
}
```
See Also

HashStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
HashStats...: OverflowPages Property

**HashStats Class**  **See Also**

Number of overflow pages.

**Namespace:**  BerkeleyDB

**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public uint OverflowPages { get; }

Visual Basic (Declaration)

Public ReadOnly Property OverflowPages AsUInteger

Visual C++

public:
property unsigned int OverflowPages {
        unsigned int get ();
}

See Also

HashStats Class  
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Bytes free on ovfl pages.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public ulong OverflowPagesFreeBytes { get; }

Visual Basic (Declaration)

Public ReadOnly Property OverflowPagesFreeBytes As ULong

Visual C++

public:
property unsigned long long OverflowPagesFreeBytes {
    unsigned long long get ();
}

See Also

HashStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Page size.

Namespace: **BerkeleyDB**
Syntax

C#

public uint PageSize { get; }

Visual Basic (Declaration)

Public ReadOnly Property PageSize AsUInteger

Visual C++

public:
property unsigned int PageSize {
    unsigned int get ();
}


See Also

HashStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
HashStats Class  See Also

Version number.

Namespace:  BerkeleyDB
## Syntax

**C#**

```csharp
public uint Version { get; }
```

**Visual Basic (Declaration)**

```vbnet
Public ReadOnly Property Version AsUInteger
```

**Visual C++**

```cpp
public:
property unsigned int Version {
    unsigned int get ();
}
```
See Also

HashStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Specify the degree of isolation for transactional operations

**Namespace:**  [BerkeleyDB](https://berkeleydb.com)

**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#
public enum Isolation

Visual Basic (Declaration)
Public Enumeration Isolation

Visual C++
public enum class Isolation
## Members

<table>
<thead>
<tr>
<th>Member name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEGREE_ONE</td>
<td>Read operations on the database may request the return of modified but not yet committed data. Provide for cursor stability but not repeatable reads. Data items which have been previously read by a transaction may be deleted or modified by other transactions before the original transaction completes. For the life of the transaction, every time a thread of control reads a data item, it will be unchanged from its previous value (assuming, of course, the thread of control does not itself modify the item). This is Berkeley DB’s default degree of isolation.</td>
</tr>
<tr>
<td>DEGREE_TWO</td>
<td></td>
</tr>
<tr>
<td>DEGREE_THREE</td>
<td></td>
</tr>
</tbody>
</table>
See Also

BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
A class representing a join cursor, for use in performing equality or natural joins on secondary indices. For information on how to organize your data to use this functionality, see Equality join in the Programmer’s Reference Guide.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public class JoinCursor : IDisposable, IEnumerable<KeyValuePair<DatabaseEntry>>, IEnumerable

Visual Basic (Declaration)

Public Class JoinCursor
    Implements IDisposable, IEnumerable(Of KeyValuePair(Of DatabaseEntry))
    Implements IEnumerable

Visual C++

public ref class JoinCursor : IDisposable,
    IEnumerable<KeyValuePair<DatabaseEntry^, DatabaseEntry^>>, ]
Remarks

JoinCursor does not support many of the operations offered by Cursor and is not a subclass of Cursor.
Inheritance Hierarchy

System..:::.Object
BerkeleyDB..:::.JoinCursor
See Also

JoinCursor Members
BerkeleyDB Namespace
Database::Join(array<SecondaryCursor>[], Boolean)

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The JoinCursor type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Close</td>
<td>Discard the cursor. It is possible for the Close() method to throw a DeadlockException, signaling that any enclosing transaction should be aborted. If the application is already intending to abort the transaction, this error should be ignored, and the application should proceed. After Close has been called, regardless of its result, the object may not be used again.</td>
</tr>
<tr>
<td>Dispose</td>
<td>Release the resources held by this object, and close the cursor if it's still open. Determines whether the specified Object is equal to the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Equals</td>
<td>Returns an enumerator that iterates through the JoinCursor. Serves as a hash function for a particular type. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetEnumerator</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Overloaded. Serves as a hash function for a particular type. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Overloaded. Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MoveNext</td>
<td>Overloaded. Returns a String that represents the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>MoveNextItem</td>
<td>Returns a String that represents the current Object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current</td>
<td>The key/data pair at which the cursor currently points.</td>
</tr>
</tbody>
</table>
See Also

JoinCursor Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `JoinCursor` type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Close</strong></td>
<td>Discard the cursor. It is possible for the Close() method to throw a <code>DeadlockException</code>, signaling that any enclosing transaction should be aborted. If the application is already intending to abort the transaction, this error should be ignored, and the application should proceed. After Close has been called, regardless of its result, the object may not be used again.</td>
</tr>
<tr>
<td><strong>Dispose</strong></td>
<td>Release the resources held by this object, and close the cursor if it's still open. Determines whether the specified <code>Object</code> is equal to the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>Equals</strong></td>
<td>Returns an enumerator that iterates through the <code>JoinCursor</code>.</td>
</tr>
<tr>
<td><strong>GetEnumerator</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Gets the <code>Type</code> of the current instance. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>MoveNext</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>MoveNextItem</strong></td>
<td>Returns a <code>String</code> that represents the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a <code>String</code> that represents the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
</tbody>
</table>
See Also

JoinCursor Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Discard the cursor.

It is possible for the Close() method to throw a `DeadlockException`, signaling that any enclosing transaction should be aborted. If the application is already intending to abort the transaction, this error should be ignored, and the application should proceed.

After Close has been called, regardless of its result, the object may not be used again.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public void Close()

Visual Basic (Declaration)

Public Sub Close

Visual C++

public:
void Close()
## Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>BerkeleyDB::DeadlockException</td>
<td></td>
</tr>
</tbody>
</table>
See Also

JoinCursor Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Release the resources held by this object, and close the cursor if it's still open.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
**Syntax**

**C#**

```csharp
public void Dispose()
```

**Visual Basic (Declaration)**

```vbnet
Public Sub Dispose
```

**Visual C++**

```cpp
public: 
virtual void Dispose() sealed
```

**Implements**

```csharp
IDisposable:::Dispose()
```
See Also

JoinCursor Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Returns an enumerator that iterates through the `JoinCursor`.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public IEnumerator<KeyValuePair<DatabaseEntry, DatabaseEntry>> GetEnumerator;

Visual Basic (Declaration)

Public Function GetEnumerator As IEnumerator(Of KeyValuePair(Of DatabaseEntry, DatabaseEntry))

Visual C++

public:
virtual IEnumerator<KeyValuePair<DatabaseEntry^, DatabaseEntry^>>^ GetEnumerator;

ReturnValue

An enumerator for the Cursor.

Implements

IEnumerator(Of (T>)... GetEnumerator()()
Remarks

The enumerator will begin at the cursor's current position (or the first record if the cursor has not yet been positioned) and iterate forwards (i.e. in the direction of `MoveNext()`) over the remaining records.
See Also

JoinCursor Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
JoinCursor Class  See Also
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MoveNext()()</td>
<td>Iterate over the values associated with the keys to which each SecondaryCursor passed to Join(array&lt;SecondaryCursor&gt;[][]) Boolean was initialized. Any data value that appears in all SecondaryCursors is then used as a key into the primary, and the key/data pair found in the primary is stored in Current. Iterate over the values associated with the keys to which each SecondaryCursor passed to Join(array&lt;SecondaryCursor&gt;[][]) Boolean was initialized. Any data value that appears in all SecondaryCursors is then used as a key into the primary, and the key/data pair found in the primary is stored in Current.</td>
</tr>
<tr>
<td>MoveNext(LockingInfo)</td>
<td></td>
</tr>
</tbody>
</table>
See Also

JoinCursor Class
JoinCursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation

JoinCursor::MoveNext Method

JoinCursor Class  See Also

Iterate over the values associated with the keys to which each SecondaryCursor passed to Join(array<SecondaryCursor>[][], Boolean) was initialized. Any data value that appears in all SecondaryCursors is then used as a key into the primary, and the key/data pair found in the primary is stored in Current.

Namespace:  BerkeleyDB
### Syntax

**C#**

```csharp
public bool MoveNext()
```

**Visual Basic (Declaration)**

```vbnet
Public Function MoveNext As Boolean
```

**Visual C++**

```cpp
public:
bool MoveNext()
```

### Return Value

True if the cursor was positioned successfully, false otherwise.
See Also

JoinCursor Class
MoveNext Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Iterate over the values associated with the keys to which each `SecondaryCursor` passed to `Join(array<SecondaryCursor>[][], Boolean)` was initialized. Any data value that appears in all `SecondaryCursor`es is then used as a key into the primary, and the key/data pair found in the primary is stored in `Current`.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool MoveNext(
    LockingInfo info
)

Visual Basic (Declaration)

Public Function MoveNext ( _
    info As LockingInfo _
) As Boolean

Visual C++

public:
    bool MoveNext(
        LockingInfo^ info
    )

Parameters

info
    Type: BerkeleyDB:::LockingInfo
    The locking behavior to use.

Return Value

True if the cursor was positioned successfully, false otherwise.
See Also

JoinCursor Class
MoveNext Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
JoinCursor

MoveNextItem Method

JoinCursor Class  See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MoveNextItem()()</td>
<td>Iterate over the values associated with the keys to which each <code>SecondaryCursor</code> passed to <code>Join(array&lt;SecondaryCursor&gt;[][], Boolean)</code> was initialized. Any data value that appears in all <code>SecondaryCursor</code>s is then stored in <code>Current.Key</code>.</td>
</tr>
<tr>
<td>MoveNextItem(LockingInfo)()[]</td>
<td>Iterate over the values associated with the keys to which each <code>SecondaryCursor</code> passed to <code>Join(array&lt;SecondaryCursor&gt;[][], Boolean)</code> was initialized. Any data value that appears in all <code>SecondaryCursor</code>s is then stored in <code>Current.Key</code>.</td>
</tr>
</tbody>
</table>
See Also

JoinCursor Class
JoinCursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Iterate over the values associated with the keys to which each `SecondaryCursor` passed to `Join(array<SecondaryCursor>[][], Boolean)` was initialized. Any data value that appears in all `SecondaryCursor`s is then stored in `Current.Key`.

**Namespace:** BerkeleyDB
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool MoveNextItem()

Visual Basic (Declaration)

Public Function MoveNextItem As Boolean

Visual C++

public:
bool MoveNextItem()

Return Value

True if the cursor was positioned successfully, false otherwise.
Remarks

Current.Value will contain an empty DatabaseEntry.
See Also

JoinCursor Class
MoveNextItem Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
JoinCursor...::MoveNextItem Method (LockingInfo)

Iterate over the values associated with the keys to which each SecondaryCursor passed to Join(array<SecondaryCursor>[][], Boolean) was initialized. Any data value that appears in all SecondaryCursors is then stored in Current.Key.

Namespace: BerkeleyDB
Syntax

C#

public bool MoveNextItem(
    LockingInfo info
)

Visual Basic (Declaration)

Public Function MoveNextItem ( _
    info As LockingInfo _
) As Boolean

Visual C++

public:
    bool MoveNextItem(
        LockingInfo^ info
    )

Parameters

info

    Type: BerkeleyDB::::LockingInfo
    The locking behavior to use.

Return Value

True if the cursor was positioned successfully, false otherwise.
Remarks

Current.Value will contain an empty DatabaseEntry.
See Also

JoinCursor Class
MoveNextItem Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `JoinCursor` type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current</td>
<td>The key/data pair at which the cursor currently points.</td>
</tr>
</tbody>
</table>
See Also

JoinCursor Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
JoinCursor Class  See Also

The key/data pair at which the cursor currently points.

Namespace: BerkeleyDB  
Syntax

C#

public KeyValuePair<DatabaseEntry, DatabaseEntry> Current { get; private set; }

Visual Basic (Declaration)

Public Property Current As KeyValuePair(Of DatabaseEntry, DatabaseEntry)

Visual C++

public:
property KeyValuePair<DatabaseEntry^, DatabaseEntry^> Current {
KeyValuePair<DatabaseEntry^, DatabaseEntry^> get ();
void set (KeyValuePair<DatabaseEntry^, DatabaseEntry^> value);}
See Also

JoinCursor Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The requested key/data pair logically exists but was never explicitly created by the application, or that the requested key/data pair was deleted and never re-created. In addition, the Queue access method will throw a KeyEmptyException for records that were created as part of a transaction that was later aborted and never re-created.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll)  
**Version:** 4.8.24.0
Syntax

C#

public class KeyEmptyException : DatabaseException

Visual Basic (Declaration)

Public Class KeyEmptyException
    Inherits DatabaseException

Visual C++

public ref class KeyEmptyException : public DatabaseException
Remarks

The Recno and Queue access methods will automatically create key/data pairs under some circumstances.
Inheritance Hierarchy

System..::.Object
   System..::.Exception
      BerkeleyDB..::.DatabaseException
         BerkeleyDB..::.KeyEmptyException
See Also

KeyEmptyException Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The KeyEmptyException type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>KeyEmptyException</strong></td>
<td>Initialize a new instance of the KeyEmptyException</td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| **Equals**        | Determines whether the specified [Object](#) is equal to the current [Object](#).  
When overridden in a derived class, returns the [Exception](#) that is the root cause of one or more subsequent exceptions.  
(Inherited from [Object](#).) |
| **GetBaseException** | Serves as a hash function for a particular type.  
When overridden in a derived class, sets the [SerializationInfo](#) with information about the exception.  
(Inherited from [Exception](#).) |
| **GetHashCode**   | Gets the runtime type of the current instance.  
(Inherited from [Exception](#).) |
| **GetObjectData** | Gets and returns a string representation of the current exception.  
(Inherited from [Exception](#).) |
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ErrorCode</td>
<td>The underlying error code from the Berkeley DB C library. (Inherited from <code>DatabaseException</code>.)</td>
</tr>
</tbody>
</table>
### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td>HelpLink</td>
<td>Gets or sets a link to the help file associated with this exception.        (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td>InnerException</td>
<td>Gets the <code>Exception</code> instance that caused the current exception.             (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td>Message</td>
<td>Gets a message that describes the current exception.                        (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td>Source</td>
<td>Gets or sets the name of the application or the object that causes the error. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td>StackTrace</td>
<td>Gets a string representation of the frames on the call stack at the time the current exception was thrown. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td>TargetSite</td>
<td>Gets the method that throws the current exception.                          (Inherited from <code>Exception</code>.)</td>
</tr>
</tbody>
</table>
See Also

KeyEmptyException Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
KeyEmptyException Constructor

Initialize a new instance of the KeyEmptyException

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public KeyEmptyException()

Visual Basic (Declaration)

Public Sub New

Visual C++

public:
KeyEmptyException()
See Also

KeyEmptyException Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The KeyEmptyException type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ErrorCode</td>
<td>The underlying error code from the Berkeley DB C library. (Inherited from DatabaseException.)</td>
</tr>
</tbody>
</table>
See Also

-KeyEmptyException Class
-BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The **KeyEmptyException** type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified Object is equal to the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>GetBaseException</strong></td>
<td>When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions. (Inherited from Exception.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>GetObjectData</strong></td>
<td>When overridden in a derived class, sets the SerializationInfo with information about the exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the runtime type of the current instance. (Inherited from Exception.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Creates and returns a string representation of the current exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>
See Also

KeyEmptyException Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `KeyEmptyException` type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data</strong></td>
<td>Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td><strong>HelpLink</strong></td>
<td>Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td><strong>InnerException</strong></td>
<td>Gets the Exception instance that caused the current exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td><strong>Message</strong></td>
<td>Gets a message that describes the current exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td><strong>Source</strong></td>
<td>Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)</td>
</tr>
<tr>
<td><strong>StackTrace</strong></td>
<td>Gets a string representation of the frames on the call stack at the time the current exception was thrown. (Inherited from Exception.)</td>
</tr>
<tr>
<td><strong>TargetSite</strong></td>
<td>Gets the method that throws the current exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>
See Also

KeyEmptyException Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
A key/data pair was inserted into the database using `PutNoOverwrite(DatabaseEntry, DatabaseEntry)` and the key already exists in the database, or using `PutNoDuplicate(DatabaseEntry, DatabaseEntry)` or `PutNoDuplicate(DatabaseEntry, DatabaseEntry)` and the key/data pair already exists in the database.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

**C#**

```csharp
public class KeyExistException : DatabaseException
```

**Visual Basic (Declaration)**

```vbnet
Public Class KeyExistException
    Inherits DatabaseException
```

**Visual C++**

```cpp
public ref class KeyExistException : public DatabaseException
```
Inheritance Hierarchy

System..:::Object
  System..:::Exception
    BerkeleyDB..:::.DatabaseException
      BerkeleyDB..:::.KeyExistException
See Also

KeyExistException Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
KeyExistException Members

The **KeyExistException** type exposes the following members.
### Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>KeyExistException</code></td>
<td>Initialize a new instance of the KeyExistException</td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a> is equal to the current <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>. When overridden in a derived class, returns the <a href="https://docs.microsoft.com/en-us/dotnet/api/system.exception">Exception</a> that is the root cause of one or more subsequent exceptions. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.exception">Exception</a>).</td>
</tr>
<tr>
<td><strong>GetBaseException</strong></td>
<td>Serves as a hash function for a particular type. When overridden in a derived class, sets the <a href="https://docs.microsoft.com/en-us/dotnet/api/system.serialization.serializationinfo">SerializationInfo</a> with information about the exception. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.exception">Exception</a>).</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td></td>
</tr>
<tr>
<td><strong>GetObjectData</strong></td>
<td></td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the runtime type of the current instance. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.exception">Exception</a>).</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Creates and returns a string representation of the current exception. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.exception">Exception</a>).</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ErrorCode</td>
<td>The underlying error code from the Berkeley DB C library. (Inherited from <a href="#">DatabaseException</a>.)</td>
</tr>
</tbody>
</table>
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>HelpLink</td>
<td>Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>InnerException</td>
<td>Gets the Exception instance that caused the current exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>Message</td>
<td>Gets a message that describes the current exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>Source</td>
<td>Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)</td>
</tr>
<tr>
<td>StackTrace</td>
<td>Gets a string representation of the frames on the call stack at the time the current exception was thrown. (Inherited from Exception.)</td>
</tr>
<tr>
<td>TargetSite</td>
<td>Gets the method that throws the current exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>
See Also

KeyExistException Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
KeyExistException Constructor

Initialize a new instance of the KeyExistException

Namespace: BerkeleyDB
Syntax

**C#**

```csharp
public KeyExistException()
```

**Visual Basic (Declaration)**

```vbnet
Public Sub New
```

**Visual C++**

```cpp
public:
KeyExistException()
```
See Also

KeyExistException Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `KeyExistException` type exposes the following members.
## Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ErrorCode</td>
<td>The underlying error code from the Berkeley DB C library.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">DatabaseException</a>.)</td>
</tr>
</tbody>
</table>
See Also

KeyExistException Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `KeyExistException` type exposes the following members.
# Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <code>Object</code> is equal to the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>GetBaseException</strong></td>
<td>When overridden in a derived class, returns the <code>Exception</code> that is the root cause of one or more subsequent exceptions. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>GetObjectData</strong></td>
<td>When overridden in a derived class, sets the <code>SerializationInfo</code> with information about the exception. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the runtime type of the current instance. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Creates and returns a string representation of the current exception. (Inherited from <code>Exception</code>.)</td>
</tr>
</tbody>
</table>
See Also

KeyExistException Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `KeyExistException` type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data</strong></td>
<td>Gets a collection of key/value pairs that provide additional user-defined information about the exception. <em>(Inherited from Exception.)</em></td>
</tr>
<tr>
<td><strong>HelpLink</strong></td>
<td>Gets or sets a link to the help file associated with this exception.        <em>(Inherited from Exception.)</em></td>
</tr>
<tr>
<td><strong>InnerException</strong></td>
<td>Gets the Exception instance that caused the current exception.            <em>(Inherited from Exception.)</em></td>
</tr>
<tr>
<td><strong>Message</strong></td>
<td>Gets a message that describes the current exception.                       <em>(Inherited from Exception.)</em></td>
</tr>
<tr>
<td><strong>Source</strong></td>
<td>Gets or sets the name of the application or the object that causes the error. <em>(Inherited from Exception.)</em></td>
</tr>
<tr>
<td><strong>StackTrace</strong></td>
<td>Gets a string representation of the frames on the call stack at the time the current exception was thrown. <em>(Inherited from Exception.)</em></td>
</tr>
<tr>
<td><strong>TargetSite</strong></td>
<td>Gets the method that throws the current exception.                          <em>(Inherited from Exception.)</em></td>
</tr>
</tbody>
</table>
See Also

KeyExistException Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
A class representing an estimate of the proportion of keys that are less than, equal to, and greater than a given key.

Values are in the range of 0 to 1; for example, if the field less is 0.05, 5% of the keys in the database are less than the key parameter. The value for equal will be zero if there is no matching key, and will be non-zero otherwise.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public class KeyRange

Visual Basic (Declaration)

Public Class KeyRange

Visual C++

public ref class KeyRange
Inheritance Hierarchy

System..:::Object
BerkeleyDB..:::KeyRange
See Also

KeyRange Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The **KeyRange** type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a <a href="#">String</a> that represents the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
</tbody>
</table>
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equal</td>
<td>A value between 0 and 1, the proportion of keys equal to the specified key.</td>
</tr>
<tr>
<td>Greater</td>
<td>A value between 0 and 1, the proportion of keys greater than the specified key.</td>
</tr>
<tr>
<td>Less</td>
<td>A value between 0 and 1, the proportion of keys less than the specified key.</td>
</tr>
</tbody>
</table>
See Also

KeyRange Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The **KeyRange** type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a>.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <a href="#">Type</a> of the current instance.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a <a href="#">String</a> that represents the current <a href="#">Object</a>.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Object</a>.)</td>
</tr>
</tbody>
</table>
See Also

KeyRange Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
KeyRange Properties

KeyRange Class  See Also

The KeyRange type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equal</td>
<td>A value between 0 and 1, the proportion of keys equal to the specified key.</td>
</tr>
<tr>
<td>Greater</td>
<td>A value between 0 and 1, the proportion of keys greater than the specified key.</td>
</tr>
<tr>
<td>Less</td>
<td>A value between 0 and 1, the proportion of keys less than the specified key.</td>
</tr>
</tbody>
</table>
See Also

KeyRange Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
A value between 0 and 1, the proportion of keys equal to the specified key.

Namespace: BerkeleyDB
Syntax

**C#**

public double Equal { get; }

**Visual Basic (Declaration)**

Public Readonly Property Equal As Double

**Visual C++**

public:
    property double Equal {
        double get ();
    }

See Also

KeyRange Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
A value between 0 and 1, the proportion of keys greater than the specified key.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

**C#**

```csharp
public double Greater { get; }
```

**Visual Basic (Declaration)**

```vbnet
Public ReadOnly Property Greater As Double
```

**Visual C++**

```cpp
public:
    property double Greater {
        double get();
    }
```
See Also

KeyRange Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
A value between 0 and 1, the proportion of keys less than the specified key.

**Namespace:**  [BerkeleyDB](https://www.berkeleydb.com/)

**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public double Less { get; }

Visual Basic (Declaration)

Public ReadOnly Property Less As Double

Visual C++

public:
property double Less {
    double get ();
}

See Also

KeyRange Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The site's replication master lease has expired.

**Namespace:** BerkeleyDB
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public class LeaseExpiredException : DatabaseException

Visual Basic (Declaration)

Public Class LeaseExpiredException
    Inherits DatabaseException

Visual C++

public ref class LeaseExpiredException : public DatabaseException
Inheritance Hierarchy

System::Object
System::Exception
BerkeleyDB::DatabaseException
BerkeleyDB::LeaseExpiredException
See Also

LeaseExpiredException Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `LeaseExpiredException` type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LeaseExpiredException</td>
<td>Initialize a new instance of the LeaseExpiredException</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a> is equal to the current <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.) When overridden in a derived class, returns the <a href="https://docs.microsoft.com/en-us/dotnet/api/system.exception">Exception</a> that is the root cause of one or more subsequent exceptions. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.exception">Exception</a>.)</td>
</tr>
<tr>
<td><strong>GetBaseException</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.) When overridden in a derived class, sets the <a href="https://docs.microsoft.com/en-us/dotnet/api/system.serialization.serializationinfo">SerializationInfo</a> with information about the exception. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.exception">Exception</a>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Gets the runtime type of the current instance. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.exception">Exception</a>.) Creates and returns a string representation of the current exception. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.exception">Exception</a>.)</td>
</tr>
<tr>
<td><strong>GetObjectData</strong></td>
<td></td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td></td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ErrorCode</td>
<td>The underlying error code from the Berkeley DB C library.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">DatabaseException</a>.)</td>
</tr>
</tbody>
</table>
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>HelpLink</td>
<td>Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>InnerException</td>
<td>Gets the Exception instance that caused the current exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>Message</td>
<td>Gets a message that describes the current exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>Source</td>
<td>Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)</td>
</tr>
<tr>
<td>StackTrace</td>
<td>Gets a string representation of the frames on the call stack at the time the current exception was thrown. (Inherited from Exception.)</td>
</tr>
<tr>
<td>TargetSite</td>
<td>Gets the method that throws the current exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>
See Also

LeaseExpiredException Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Initialize a new instance of the LeaseExpiredException

**Namespace:**  BerkeleyDB  
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
### Syntax

**C#**

```csharp
public LeaseExpiredException()
```

**Visual Basic (Declaration)**

```vbnet
Public Sub New
```

**Visual C++**

```cpp
public:
LeaseExpiredException()
```
See Also

LeaseExpiredException Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The LeaseExpiredException type exposes the following members.
## Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ErrorCode</td>
<td>The underlying error code from the Berkeley DB C library. (Inherited from <code>DatabaseException</code>.)</td>
</tr>
</tbody>
</table>
See Also

LeaseExpiredException Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `LeaseExpiredException` type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <strong>Object</strong> is equal to the current <strong>Object</strong>. (Inherited from <strong>Object</strong>.) When overridden in a derived class, returns the <strong>Exception</strong> that is the root cause of one or more subsequent exceptions. (Inherited from <strong>Exception</strong>.)</td>
</tr>
<tr>
<td><strong>GetBaseException</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <strong>Object</strong>.) When overridden in a derived class, sets the <strong>SerializationInfo</strong> with information about the exception. (Inherited from <strong>Exception</strong>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Gets the runtime type of the current instance. (Inherited from <strong>Exception</strong>.) Creates and returns a string representation of the current exception. (Inherited from <strong>Exception</strong>.)</td>
</tr>
<tr>
<td><strong>GetObjectData</strong></td>
<td>Creates and returns a string representation of the current exception. (Inherited from <strong>Exception</strong>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>(Inherited from <strong>Exception</strong>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>(Inherited from <strong>Exception</strong>.)</td>
</tr>
</tbody>
</table>
See Also

LeaseExpiredException Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `LeaseExpiredException` type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data</strong></td>
<td>Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td><strong>HelpLink</strong></td>
<td>Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td><strong>InnerException</strong></td>
<td>Gets the Exception instance that caused the current exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td><strong>Message</strong></td>
<td>Gets a message that describes the current exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td><strong>Source</strong></td>
<td>Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)</td>
</tr>
<tr>
<td><strong>StackTrace</strong></td>
<td>Gets a string representation of the frames on the call stack at the time the current exception was thrown. (Inherited from Exception.)</td>
</tr>
<tr>
<td><strong>TargetSite</strong></td>
<td>Gets the method that throws the current exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>
See Also

LeaseExpiredException Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation

LockingConfig Class

Members  See Also

A class representing configuration parameters for a DatabaseEnvironment’s locking subsystem.

Namespace: BerkeleyDB
Syntax

C#

public class LockingConfig

Visual Basic (Declaration)

Public Class LockingConfig

Visual C++

public ref class LockingConfig
Inheritance Hierarchy

System..::..Object
BerkeleyDB..::..LockingConfig
See Also

LockingConfig Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation
LockingConfig Members

LockingConfig Class  Constructors  Methods  Fields  Properties  See Also

The LockingConfig type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LockingConfig</td>
<td>Initializes a new instance of the LockingConfig class</td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="#">Object</a>)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a>)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a String that represents the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>DeadlockResolution</td>
<td>If non-null, the deadlock detector is to be run whenever a lock conflict occurs, lock request(s) should be rejected according to the specified policy.</td>
</tr>
</tbody>
</table>
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conflicts</td>
<td>The locking conflicts matrix.</td>
</tr>
<tr>
<td>MaxLockers</td>
<td>The maximum number of simultaneous locking entities supported by the Berkeley DB environment.</td>
</tr>
<tr>
<td>MaxLocks</td>
<td>The maximum number of locks supported by the Berkeley DB environment.</td>
</tr>
<tr>
<td>MaxObjects</td>
<td>The maximum number of locked objects supported by the Berkeley DB environment.</td>
</tr>
<tr>
<td>Partitions</td>
<td>The number of lock table partitions in the Berkeley DB environment.</td>
</tr>
</tbody>
</table>
See Also

LockingConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
LockingConfig Constructor

Initializes a new instance of the LockingConfig class

Namespace: BerkeleyDB
Syntax

C#

public LockingConfig()

Visual Basic (Declaration)

Public Sub New

Visual C++

public:  
LockingConfig()
See Also

LockingConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The LockingConfig type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DeadlockResolution</strong></td>
<td>If non-null, the deadlock detector is to be run whenever a lock conflict occurs, lock request(s) should be rejected according to the specified policy.</td>
</tr>
</tbody>
</table>
See Also

LockingConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If non-null, the deadlock detector is to be run whenever a lock conflict occurs, lock request(s) should be rejected according to the specified policy.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public DeadlockPolicy DeadlockResolution

Visual Basic (Declaration)

Public DeadlockResolution As DeadlockPolicy

Visual C++

public: DeadlockPolicy^ DeadlockResolution
Remarks

As transactions acquire locks on behalf of a single locker ID, rejecting a lock request associated with a transaction normally requires the transaction be aborted.
See Also

LockingConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `LockingConfig` type exposes the following members.
Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified Object is equal to the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as a hash function for a particular type. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a String that represents the current Object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>
See Also

LockingConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `LockingConfig` type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conflicts</td>
<td>The locking conflicts matrix.</td>
</tr>
<tr>
<td>MaxLockers</td>
<td>The maximum number of simultaneous locking entities supported by the Berkeley DB environment.</td>
</tr>
<tr>
<td>MaxLocks</td>
<td>The maximum number of locks supported by the Berkeley DB environment.</td>
</tr>
<tr>
<td>MaxObjects</td>
<td>The maximum number of locked objects supported by the Berkeley DB environment.</td>
</tr>
<tr>
<td>Partitions</td>
<td>The number of lock table partitions in the Berkeley DB environment.</td>
</tr>
</tbody>
</table>
See Also

LockingConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The locking conflicts matrix.

Namespace: BerkeleyDB
Syntax

C#

public byte[,] Conflicts { get; set; }

Visual Basic (Declaration)

Public Property Conflicts As Byte(,)

Visual C++

public:
property array<unsigned char, 2>^ Conflicts {
    array<unsigned char, 2>^ get ();
    void set (array<unsigned char, 2>^ value);
}
Remarks

If Conflicts is never set, a standard conflicts array is used; see Standard Lock Modes in the Programmer's Reference Guide for more information.

Conflicts parameter is an nmodes by nmodes array. A non-0 value for the array element indicates that requested_mode and held_mode conflict:

C#

```csharp
conflicts[requested_mode][held_mode]
```

The not-granted mode must be represented by 0.

If the database environment already exists when `Open(String, DatabaseEnvironmentConfig)` is called, the value of Conflicts will be ignored.
See Also

LockingConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The maximum number of simultaneous locking entities supported by the Berkeley DB environment

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public uint MaxLockers { get; set; }

Visual Basic (Declaration)

Public Property MaxLockers AsUInteger

Visual C++

public:
property unsigned int MaxLockers {
    unsigned int get ();
    void set (unsigned int value);
}
Remarks

This value is used by Open(String, DatabaseEnvironmentConfig) to estimate how much space to allocate for various lock-table data structures. The default value is 1000 lockers. For specific information on configuring the size of the lock subsystem, see Configuring locking: sizing the system in the Programmer's Reference Guide.

If the database environment already exists when Open(String, DatabaseEnvironmentConfig) is called, the value of MaxLockers will be ignored.
See Also

LockingConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The maximum number of locks supported by the Berkeley DB environment.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
**Syntax**

**C#**

```csharp
public uint MaxLocks { get; set; }
```

**Visual Basic (Declaration)**

```vbscript
Public Property MaxLocks AsUInteger
```

**Visual C++**

```csharp
public:
property unsigned int MaxLocks {
    unsigned int get ();
    void set (unsigned int value);
}
```
Remarks

This value is used by Open(String, DatabaseEnvironmentConfig) to estimate how much space to allocate for various lock-table data structures. The default value is 1000 lockers. For specific information on configuring the size of the lock subsystem, see Configuring locking: sizing the system in the Programmer's Reference Guide.

If the database environment already exists when Open(String, DatabaseEnvironmentConfig) is called, the value of MaxLocks will be ignored.
See Also

LockingConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The maximum number of locked objects supported by the Berkeley DB environment.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
**Syntax**

**C#**

```csharp
public uint MaxObjects { get; set; }
```

**Visual Basic (Declaration)**

```vbnet
Public Property MaxObjects As UInteger
```

**Visual C++**

```cpp
public:
property unsigned int MaxObjects {
    unsigned int get ();
    void set (unsigned int value);
}
```
Remarks

This value is used by `Open(String, DatabaseEnvironmentConfig)` to estimate how much space to allocate for various lock-table data structures. The default value is 1000 lockers. For specific information on configuring the size of the lock subsystem, see Configuring locking: sizing the system in the Programmer's Reference Guide.

If the database environment already exists when `Open(String, DatabaseEnvironmentConfig)` is called, the value of MaxObjects will be ignored.
See Also

*LockingConfig Class*
*BerkeleyDB Namespace*

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The number of lock table partitions in the Berkeley DB environment.

Namespace: BerkeleyDB
Syntax

C#

public uint Partitions { get; set; }

Visual Basic (Declaration)

Public Property Partitions AsUInteger

Visual C++

public:
property unsigned int Partitions {
    unsigned int get();
    void set (unsigned int value);
}
Remarks

The default value is 10 times the number of CPUs on the system if there is more than one CPU. Increasing the number of partitions can provide for greater throughput on a system with multiple CPUs and more than one thread contending for the lock manager. On single processor systems more than one partition may increase the overhead of the lock manager. Systems often report threading contexts as CPUs. If your system does this, set the number of partitions to 1 to get optimal performance.

If the database environment already exists when `Open(String, DatabaseEnvironmentConfig)` is called, the value of Partitions will be ignored.
See Also

LockingConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
A class representing the locking options for Berkeley DB operations.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public class LockingInfo

Visual Basic (Declaration)

Public Class LockingInfo

Visual C++

public ref class LockingInfo
Inheritance Hierarchy

System..:::.Object
BerkeleyDB..:::.LockingInfo
See Also

LockingInfo Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `LockingInfo` type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LockingInfo</td>
<td>Instantiate a new LockingInfo object</td>
</tr>
</tbody>
</table>
# Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a> is equal to the current <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.  (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type.  (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <a href="https://docs.microsoft.com/en-us/dotnet/api/system.type">Type</a> of the current instance.  (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a <a href="https://docs.microsoft.com/en-us/dotnet/api/system.string">String</a> that represents the current <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.  (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
</tbody>
</table>
# Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IsolationDegree</td>
<td>The isolation degree of the operation.</td>
</tr>
<tr>
<td>ReadModifyWrite</td>
<td>If true, acquire write locks instead of read locks when doing a read, if locking is configured.</td>
</tr>
</tbody>
</table>
See Also

LockingInfo Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Instantiate a new LockingInfo object

**Namespace:** BerkeleyDB
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public LockingInfo()

Visual Basic (Declaration)

Public Sub New

Visual C++

public:
LockingInfo()
See Also

LockingInfo Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `LockingInfo` type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IsolationDegree</td>
<td>The isolation degree of the operation.</td>
</tr>
<tr>
<td>ReadModifyWrite</td>
<td>If true, acquire write locks instead of read locks when doing a read, if locking is configured.</td>
</tr>
</tbody>
</table>
See Also

[Link: LockingInfo Class]
[Link: BerkeleyDB Namespace]

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The isolation degree of the operation.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public Isolation IsolationDegree

Visual Basic (Declaration)

Public IsolationDegree As Isolation

Visual C++

public:
Isolation IsolationDegree
See Also

LockingInfo Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, acquire write locks instead of read locks when doing a read, if locking is configured.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool ReadModifyWrite

Visual Basic (Declaration)

Public ReadModifyWrite As Boolean

Visual C++

public:
bool ReadModifyWrite
Remarks

Setting ReadModifyWrite can eliminate deadlock during a read-modify-write cycle by acquiring the write lock during the read part of the cycle so that another thread of control acquiring a read lock for the same item, in its own read-modify-write cycle, will not result in deadlock.
See Also

LockingInfo Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `LockingInfo` type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <code>Object</code> is equal to the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <code>Type</code> of the current instance. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a <code>String</code> that represents the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
</tbody>
</table>
See Also

LockingInfo Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If `TimeNotGranted` is true, database calls timing out based on lock or transaction timeout values will throw a `LockNotGrantedException`, instead of a `DeadlockException`.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll)  
**Version:** 4.8.24.0
Syntax

**C#**

```csharp
public class LockNotGrantedException : DatabaseException
```

**Visual Basic (Declaration)**

```vbnet
Public Class LockNotGrantedException
    Inherits DatabaseException
```

**Visual C++**

```cpp
public ref class LockNotGrantedException : public DatabaseException
```
Inheritance Hierarchy

System:::Object
System:::Exception
BerkeleyDB:::DatabaseException
BerkeleyDB:::LockNotGrantedException
See Also

LockNotGrantedException Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `LockNotGrantedException` type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LockNotGrantedException</td>
<td>Initialize a new instance of the LockNotGrantedException</td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a> is equal to the current <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.) When overridden in a derived class, returns the <a href="https://docs.microsoft.com/en-us/dotnet/api/system.exception">Exception</a> that is the root cause of one or more subsequent exceptions. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.exception">Exception</a>.)</td>
</tr>
<tr>
<td>GetBaseException</td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.) When overridden in a derived class, sets the <a href="https://docs.microsoft.com/en-us/dotnet/api/system.serialization.serializationinfo">SerializationInfo</a> with information about the exception. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.exception">Exception</a>.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Gets the runtime type of the current instance. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.exception">Exception</a>.) Creates and returns a string representation of the current exception. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.exception">Exception</a>.)</td>
</tr>
<tr>
<td>GetObjectData</td>
<td></td>
</tr>
<tr>
<td>GetType</td>
<td></td>
</tr>
<tr>
<td>ToString</td>
<td></td>
</tr>
</tbody>
</table>
### Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ErrorCode</td>
<td>The underlying error code from the Berkeley DB C library. (Inherited from <a href="https://example.com">DatabaseException</a>.)</td>
</tr>
</tbody>
</table>
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td>HelpLink</td>
<td>Gets or sets a link to the help file associated with this exception. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td>InnerException</td>
<td>Gets the <code>Exception</code> instance that caused the current exception. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td>Message</td>
<td>Gets a message that describes the current exception. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td>Source</td>
<td>Gets or sets the name of the application or the object that causes the error. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td>StackTrace</td>
<td>Gets a string representation of the frames on the call stack at the time the current exception was thrown. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td>TargetSite</td>
<td>Gets the method that throws the current exception. (Inherited from <code>Exception</code>.)</td>
</tr>
</tbody>
</table>
See Also

LockNotGrantedException Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Initialize a new instance of the LockNotGrantedException

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
### Syntax

#### C#

public LockNotGrantedException()

#### Visual Basic (Declaration)

Public Sub New

#### Visual C++

public:
LockNotGrantedException()
See Also

LockNotGrantedException Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `LockNotGrantedException` type exposes the following members.
# Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ErrorCode</td>
<td>The underlying error code from the Berkeley DB C library. (Inherited from DatabaseException. )</td>
</tr>
</tbody>
</table>
See Also

LockNotGrantedException Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The LockNotGrantedException type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <strong>Object</strong> is equal to the current <strong>Object</strong>. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td></td>
<td>When overridden in a derived class, returns the <strong>Exception</strong> that is the root cause of one or more subsequent exceptions. (Inherited from <strong>Exception</strong>.)</td>
</tr>
<tr>
<td><strong>GetBaseException</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td></td>
<td>When overridden in a derived class, sets the <strong>SerializationInfo</strong> with information about the exception. (Inherited from <strong>Exception</strong>.)</td>
</tr>
<tr>
<td><strong>GetObjectData</strong></td>
<td>Gets the runtime type of the current instance. (Inherited from <strong>Exception</strong>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Creates and returns a string representation of the current exception. (Inherited from <strong>Exception</strong>.)</td>
</tr>
</tbody>
</table>
See Also

LockNotGrantedException Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `LockNotGrantedException` type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data</strong></td>
<td>Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td><strong>HelpLink</strong></td>
<td>Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td><strong>InnerException</strong></td>
<td>Gets the Exception instance that caused the current exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td><strong>Message</strong></td>
<td>Gets a message that describes the current exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td><strong>Source</strong></td>
<td>Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)</td>
</tr>
<tr>
<td><strong>StackTrace</strong></td>
<td>Gets a string representation of the frames on the call stack at the time the current exception was thrown. (Inherited from Exception.)</td>
</tr>
<tr>
<td><strong>TargetSite</strong></td>
<td>Gets the method that throws the current exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>
See Also

LockNotGrantedException Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Statistical information about the locking subsystem

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public class LockStats

Visual Basic (Declaration)

Public Class LockStats

Visual C++

public ref class LockStats
Inheritance Hierarchy

System..:::Object
	BerkeleyDB..:::LockStats
See Also

LockStats Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `LockStats` type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a <a href="#">String</a> that represents the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
</tbody>
</table>
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LastAllocatedLockerID</td>
<td>Last allocated locker ID.</td>
</tr>
<tr>
<td>LockConflictsNoWait</td>
<td>Lock conflicts w/o subsequent wait</td>
</tr>
<tr>
<td>LockConflictsWait</td>
<td>Lock conflicts w/ subsequent wait</td>
</tr>
<tr>
<td>LockDeadlocks</td>
<td>Number of lock deadlocks.</td>
</tr>
<tr>
<td>LockDowngrades</td>
<td>Number of lock downgrades.</td>
</tr>
<tr>
<td>LockerNoWait</td>
<td>Locker lock granted without wait.</td>
</tr>
<tr>
<td>Lockers</td>
<td>Current number of lockers.</td>
</tr>
<tr>
<td>LockerWait</td>
<td>Locker lock granted after wait.</td>
</tr>
<tr>
<td>LockModes</td>
<td>Number of lock modes.</td>
</tr>
<tr>
<td>LockPuts</td>
<td>Number of lock puts.</td>
</tr>
<tr>
<td>LockRequests</td>
<td>Number of lock gets.</td>
</tr>
<tr>
<td>Locks</td>
<td>Current number of locks.</td>
</tr>
<tr>
<td>LockSteals</td>
<td>Number of lock steals so far.</td>
</tr>
<tr>
<td>LockTimeoutLength</td>
<td>Lock timeout.</td>
</tr>
<tr>
<td>LockTimeouts</td>
<td>Number of lock timeouts.</td>
</tr>
<tr>
<td>LockUpgrades</td>
<td>Number of lock upgrades.</td>
</tr>
<tr>
<td>MaxBucketLength</td>
<td>Max length of bucket.</td>
</tr>
<tr>
<td>MaxLockers</td>
<td>Maximum number of lockers so far.</td>
</tr>
<tr>
<td>MaxLockersInTable</td>
<td>Maximum num of lockers in table.</td>
</tr>
<tr>
<td>MaxLocks</td>
<td>Maximum number of locks so far.</td>
</tr>
<tr>
<td>MaxLocksInBucket</td>
<td>Maximum number of locks in any bucket.</td>
</tr>
<tr>
<td>MaxLocksInTable</td>
<td>Maximum number of locks in table.</td>
</tr>
<tr>
<td>MaxLockSteals</td>
<td>Maximum number steals in any partition.</td>
</tr>
<tr>
<td>MaxObjects</td>
<td>Maximum number of objects so far.</td>
</tr>
<tr>
<td>MaxObjectsInBucket</td>
<td>Maximum number of objects in any bucket.</td>
</tr>
<tr>
<td>MaxObjectsInTable</td>
<td>Maximum number of objects in table.</td>
</tr>
<tr>
<td>MaxObjectSteals</td>
<td>Maximum number of steals in any partition.</td>
</tr>
<tr>
<td>MaxPartitionLockNoWait</td>
<td>Max partition lock granted without wait.</td>
</tr>
<tr>
<td>Metric</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>------------------------------------------</td>
</tr>
<tr>
<td>MaxPartitionLockWait</td>
<td>Max partition lock granted after wait.</td>
</tr>
<tr>
<td>MaxUnusedID</td>
<td>Current maximum unused ID.</td>
</tr>
<tr>
<td>nPartitions</td>
<td>number of partitions.</td>
</tr>
<tr>
<td>ObjectNoWait</td>
<td>Object lock granted without wait.</td>
</tr>
<tr>
<td>Objects</td>
<td>Current number of objects.</td>
</tr>
<tr>
<td>ObjectSteals</td>
<td>Number of objects steals so far.</td>
</tr>
<tr>
<td>ObjectWait</td>
<td>Object lock granted after wait.</td>
</tr>
<tr>
<td>PartitionLockNoWait</td>
<td>Partition lock granted without wait.</td>
</tr>
<tr>
<td>PartitionLockWait</td>
<td>Partition lock granted after wait.</td>
</tr>
<tr>
<td>RegionNoWait</td>
<td>Region lock granted without wait.</td>
</tr>
<tr>
<td>RegionSize</td>
<td>Region size.</td>
</tr>
<tr>
<td>RegionWait</td>
<td>Region lock granted after wait.</td>
</tr>
<tr>
<td>TxnTimeoutLength</td>
<td>Transaction timeout.</td>
</tr>
<tr>
<td>TxnTimeouts</td>
<td>Number of transaction timeouts.</td>
</tr>
</tbody>
</table>
See Also

LockStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `LockStats` type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a <a href="#">String</a> that represents the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
</tbody>
</table>
See Also

LockStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The **LockStats** type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LastAllocatedLockerID</td>
<td>Last allocated locker ID.</td>
</tr>
<tr>
<td>LockConflictsNoWait</td>
<td>Lock conflicts w/o subsequent wait</td>
</tr>
<tr>
<td>LockConflictsWait</td>
<td>Lock conflicts w/ subsequent wait</td>
</tr>
<tr>
<td>LockDeadlocks</td>
<td>Number of lock deadlocks.</td>
</tr>
<tr>
<td>LockDowngrades</td>
<td>Number of lock downgrades.</td>
</tr>
<tr>
<td>LockerNoWait</td>
<td>Locker lock granted without wait.</td>
</tr>
<tr>
<td>Lockers</td>
<td>Current number of lockers.</td>
</tr>
<tr>
<td>LockerWait</td>
<td>Locker lock granted after wait.</td>
</tr>
<tr>
<td>LockModes</td>
<td>Number of lock modes.</td>
</tr>
<tr>
<td>LockPuts</td>
<td>Number of lock puts.</td>
</tr>
<tr>
<td>LockRequests</td>
<td>Number of lock gets.</td>
</tr>
<tr>
<td>Locks</td>
<td>Current number of locks.</td>
</tr>
<tr>
<td>LockSteals</td>
<td>Number of lock steals so far.</td>
</tr>
<tr>
<td>LockTimeoutLength</td>
<td>Lock timeout.</td>
</tr>
<tr>
<td>LockTimeouts</td>
<td>Number of lock timeouts.</td>
</tr>
<tr>
<td>LockUpgrades</td>
<td>Number of lock upgrades.</td>
</tr>
<tr>
<td>MaxBucketLength</td>
<td>Max length of bucket.</td>
</tr>
<tr>
<td>MaxLockers</td>
<td>Maximum number of lockers so far.</td>
</tr>
<tr>
<td>MaxLockersInTable</td>
<td>Maximum num of lockers in table.</td>
</tr>
<tr>
<td>MaxLocks</td>
<td>Maximum number of locks so far.</td>
</tr>
<tr>
<td>MaxLocksInBucket</td>
<td>Maximum number of locks in any bucket.</td>
</tr>
<tr>
<td>MaxLocksInTable</td>
<td>Maximum number of locks in table.</td>
</tr>
<tr>
<td>MaxLockSteals</td>
<td>Maximum number steals in any partition.</td>
</tr>
<tr>
<td>MaxObjects</td>
<td>Maximum number of objects so far.</td>
</tr>
<tr>
<td>MaxObjectsInBucket</td>
<td>Maximum number of objects in any bucket.</td>
</tr>
<tr>
<td>MaxObjectsInTable</td>
<td>Maximum number of objects in table.</td>
</tr>
<tr>
<td>MaxObjectSteals</td>
<td>Maximum number of steals in any partition.</td>
</tr>
<tr>
<td>MaxPartitionLockNoWait</td>
<td>Max partition lock granted without wait.</td>
</tr>
<tr>
<td>Metric</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>-----------------------------------------------------------</td>
</tr>
<tr>
<td>MaxPartitionLockWait</td>
<td>Max partition lock granted after wait.</td>
</tr>
<tr>
<td>MaxUnusedID</td>
<td>Current maximum unused ID.</td>
</tr>
<tr>
<td>nPartitions</td>
<td>number of partitions.</td>
</tr>
<tr>
<td>ObjectNoWait</td>
<td>Object lock granted without wait.</td>
</tr>
<tr>
<td>Objects</td>
<td>Current number of objects.</td>
</tr>
<tr>
<td>ObjectSteals</td>
<td>Number of objects steals so far.</td>
</tr>
<tr>
<td>ObjectWait</td>
<td>Object lock granted after wait.</td>
</tr>
<tr>
<td>PartitionLockNoWait</td>
<td>Partition lock granted without wait.</td>
</tr>
<tr>
<td>PartitionLockWait</td>
<td>Partition lock granted after wait.</td>
</tr>
<tr>
<td>RegionNoWait</td>
<td>Region lock granted without wait.</td>
</tr>
<tr>
<td>RegionSize</td>
<td>Region size.</td>
</tr>
<tr>
<td>RegionWait</td>
<td>Region lock granted after wait.</td>
</tr>
<tr>
<td>TxnTimeoutLength</td>
<td>Transaction timeout.</td>
</tr>
<tr>
<td>TxnTimeouts</td>
<td>Number of transaction timeouts.</td>
</tr>
</tbody>
</table>
See Also

LockStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Last allocated locker ID.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public uint LastAllocatedLockerID { get; }

Visual Basic (Declaration)

Public ReadOnly Property LastAllocatedLockerID AsUInteger

Visual C++

public:
property unsigned int LastAllocatedLockerID {
    unsigned int get ();
}
See Also

LockStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
LockStats Class  
See Also  

Lock conflicts w/o subsequent wait 

Namespace:  BerkeleyDB  
**Syntax**

**C#**

```csharp
public ulong LockConflictsNoWait { get; }
```

**Visual Basic (Declaration)**

```vbnet
Public ReadOnly Property LockConflictsNoWait As ULong
```

**Visual C++**

```cpp
public:
property unsigned long long LockConflictsNoWait {
    unsigned long long get ();
}
```
See Also

LockStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
LockStats Class  See Also

Lock conflicts w/ subsequent wait

Namespace:  BerkeleyDB
Syntax

C#

public ulong LockConflictsWait { get; }

Visual Basic (Declaration)

Public ReadOnly Property LockConflictsWait As ULong

Visual C++

public:
property unsigned long long LockConflictsWait {
        unsigned long long get ();
}

See Also

LockStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
LockStats Class See Also

Number of lock deadlocks.

Namespace: BerkeleyDB
Syntax

C#

public ulong LockDeadlocks { get; }

Visual Basic (Declaration)

Public ReadOnly Property LockDeadlocks As ULong

Visual C++

public:
property unsigned long long LockDeadlocks {
    unsigned long long get ();
}

See Also

LockStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
LockStats Class  See Also

Number of lock downgrades.

Namespace:  BerkeleyDB
Syntax

C#

public ulong LockDowngrades { get; }

Visual Basic (Declaration)

Public ReadOnly Property LockDowngrades As ULong

Visual C++

public:
property unsigned long long LockDowngrades {
    unsigned long long get ();
}

See Also

LockStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
LockStats Class  See Also

Locker lock granted without wait.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public ulong LockerNoWait { get; }

Visual Basic (Declaration)

Public ReadOnly Property LockerNoWait As ULong

Visual C++

public:
    property unsigned long long LockerNoWait {
        unsigned long long get ();
    }
}
See Also

LockStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
LockStats Class  See Also

Current number of lockers.

Namespace:  BerkeleyDB
# Syntax

**C#**

```csharp
public uint Lockers { get; }
```

**Visual Basic (Declaration)**

```vbnet
Public ReadOnly Property Lockers AsUInteger
```

**Visual C++**

```cpp
public:
property unsigned int Lockers {
  unsigned int get ();
}
```
See Also

LockStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
LockStats Class  See Also

Locker lock granted after wait.

Namespace:  BerkeleyDB
## Syntax

### C#

```
public ulong LockerWait { get; }
```

### Visual Basic (Declaration)

```
Public ReadOnly Property LockerWait As ULong
```

### Visual C++

```
public:
    property unsigned long long LockerWait {
        unsigned long long get ();
    }
```
See Also

LockStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
LockStats Class  See Also

Number of lock modes.

Namespace:  BerkeleyDB
**Syntax**

**C#**

```csharp
public int LockModes { get; }
```

**Visual Basic (Declaration)**

```vbnet
Public ReadOnly Property LockModes As Integer
```

**Visual C++**

```cpp
public:
property int LockModes {
    int get ()
}
```
See Also

LockStats Class  
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Number of lock puts.

Namespace: BerkeleyDB
Syntax

C#

```
public ulong LockPuts { get; }
```

Visual Basic (Declaration)

```
Public ReadOnly Property LockPuts As ULong
```

Visual C++

```
public:
property unsigned long long LockPuts {
    unsigned long long get ();
}
```
See Also

LockStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
LockRequests Property

Number of lock gets.

Namespace: BerkeleyDB
**Syntax**

**C#**

public ulong LockRequests { get; }

**Visual Basic (Declaration)**

Public ReadOnly Property LockRequests As ULong

**Visual C++**

public:
property unsigned long long LockRequests {
    unsigned long long get ();
}

See Also

LockStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
LockStats Class  See Also

Current number of locks.

Namespace:  BerkeleyDB
Syntax

C#

public uint Locks { get; }

Visual Basic (Declaration)

Public ReadOnly Property Locks As UInteger

Visual C++

public:
property unsigned int Locks {
    unsigned int get ();
}

See Also

LockStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
LockStats Class  See Also

Number of lock steals so far.

**Namespace:**  BerkeleyDB  
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public ulong LockSteals { get; }

Visual Basic (Declaration)

Public ReadOnly Property LockSteals As ULong

Visual C++

public:
property unsigned long long LockSteals {
    unsigned long long get ();
}
See Also

LockStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
LockStats.

LockTimeoutLength Property

See Also

Lock timeout.

Namespace: BerkeleyDB
Syntax

C#

public uint LockTimeoutLength { get; }

Visual Basic (Declaration)

Public ReadOnly Property LockTimeoutLength AsUInteger

Visual C++

public:
property unsigned int LockTimeoutLength {
    unsigned int get();
}
See Also

LockStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
LockStats Class  See Also

Number of lock timeouts.

Namespace:  BerkeleyDB
Syntax

C#

public ulong LockTimeouts { get; }

Visual Basic (Declaration)

Public ReadOnly Property LockTimeouts As ULong

Visual C++

public:
property unsigned long long LockTimeouts {
    unsigned long long get ();
}
See Also

LockStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Number of lock upgrades.

Namespace: BerkeleyDB
Syntax

C#

public ulong LockUpgrades { get; }

Visual Basic (Declaration)

Public ReadOnly Property LockUpgrades As ULong

Visual C++

public:
property unsigned long long LockUpgrades {
    unsigned long long get();
}
See Also

LockStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Max length of bucket.

Namespace:  [BerkeleyDB](https://github.com/BerkeleyDB/BerkeleyDB.NET)
Syntax

C#

public uint MaxBucketLength { get; }

Visual Basic (Declaration)

Public ReadOnly Property MaxBucketLength As UInteger

Visual C++

public:
property unsigned int MaxBucketLength {
    unsigned int get ();
}

See Also

LockStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
LockStats Class  See Also

Maximum number of lockers so far.

**Namespace:**  BerkeleyDB

**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public uint MaxLockers { get; }

Visual Basic (Declaration)

Public ReadOnly Property MaxLockers AsUInteger

Visual C++

public:
property unsigned int MaxLockers {
    unsigned int get ();
}

See Also

LockStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Maximum num of lockers in table.

**Namespace:**  [BerkeleyDB](#)
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

global uint MaxLockersInTable { get; }

Visual Basic (Declaration)

Public ReadOnly Property MaxLockersInTable AsUInteger

Visual C++

public:
property unsigned int MaxLockersInTable {
    unsigned int get ();
}
See Also

LockStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
LockStats Class  See Also

Maximum number of locks so far.

Namespace:  BerkeleyDB
Syntax

C#

public uint MaxLocks { get; }

Visual Basic (Declaration)

Public ReadOnly Property MaxLocks As UInteger

Visual C++

public:
property unsigned int MaxLocks {
    unsigned int get ();
}
See Also

LockStats Class  
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Maximum number of locks in any bucket.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public uint MaxLocksInBucket { get; }

Visual Basic (Declaration)

Public ReadOnly Property MaxLocksInBucket AsUInteger

Visual C++

public:
property unsigned int MaxLocksInBucket {
    unsigned int get ();
}
See Also

LockStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
LockStats.
.. maxlocksinetable property

Maximum number of locks in table.

**Namespace:** BerkeleyDB
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

```csharp
public uint MaxLocksInTable { get; }
```

Visual Basic (Declaration)

```vbnet
Public ReadOnly Property MaxLocksInTable AsUInteger
```

Visual C++

```cpp
public:
property unsigned int MaxLocksInTable {
    unsigned int get ();
}
```
See Also

LockStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Maximum number steals in any partition.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public ulong MaxLockSteals { get; }

Visual Basic (Declaration)

Public ReadOnly Property MaxLockSteals As ULong

Visual C++

public:
    property unsigned long long MaxLockSteals {
        unsigned long long get ();
    }
See Also

LockStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
LockStats Class  See Also

Maximum number of objects so far.

**Namespace:**  BerkeleyDB
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public uint MaxObjects { get; }

Visual Basic (Declaration)

Public ReadOnly Property MaxObjects AsUInteger

Visual C++

public:
property unsigned int MaxObjects {
    unsigned int get ();
}

See Also

LockStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Maximum number of objects in any bucket.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public uint MaxObjectsInBucket { get; }

Visual Basic (Declaration)

Public ReadOnly Property MaxObjectsInBucket As UInteger

Visual C++

public:
property unsigned int MaxObjectsInBucket {
    unsigned int get ();
}

See Also

LockStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
LockStats.MaxObjectsInTable Property

Maximum num of objects in table.

Namespace: BerkeleyDB
Syntax

C#

public uint MaxObjectsInTable { get; }

Visual Basic (Declaration)

Public ReadOnly Property MaxObjectsInTable AsUInteger

Visual C++

public:
property unsigned int MaxObjectsInTable {
        unsigned int get ();
}
See Also

LockStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Maximum number of steals in any partition.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public ulong MaxObjectSteals { get; }

Visual Basic (Declaration)

Public ReadOnly Property MaxObjectSteals As ULong

Visual C++

public:
property unsigned long long MaxObjectSteals {
    unsigned long long get ();
}
See Also

LockStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Max partition lock granted without wait.

Namespace: BerkeleyDB
Syntax

C#

public ulong MaxPartitionLockNoWait { get; }

Visual Basic (Declaration)

Public ReadOnly Property MaxPartitionLockNoWait As ULong

Visual C++

public:
property unsigned long long MaxPartitionLockNoWait {
    unsigned long long get ();
}

See Also

LockStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
LockStats Class  See Also

Max partition lock granted after wait.

Namespace:  BerkeleyDB
Syntax

C#

public ulong MaxPartitionLockWait { get; }

Visual Basic (Declaration)

Public ReadOnly Property MaxPartitionLockWait As ULong

Visual C++

public:
property unsigned long long MaxPartitionLockWait {
    unsigned long long get ();
}
See Also

LockStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Current maximum unused ID.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public uint MaxUnusedID { get; }

Visual Basic (Declaration)

Public ReadOnly Property MaxUnusedID As UInteger

Visual C++

public:
property unsigned int MaxUnusedID {
    unsigned int get ();
}
See Also

LockStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
LockStats Class  See Also

number of partitions.

Namespace:  BerkeleyDB
Syntax

C#

public uint nPartitions { get; }

Visual Basic (Declaration)

Public ReadOnly Property nPartitions As UInteger

Visual C++

public:
property unsigned int nPartitions {
    unsigned int get ();
}

See Also

LockStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Object lock granted without wait.

**Namespace:** BerkeleyDB
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public ulong ObjectNoWait { get; }

Visual Basic (Declaration)

Public ReadOnly Property ObjectNoWait As ULong

Visual C++

public:
property unsigned long long ObjectNoWait {
    unsigned long long get ();
}
See Also

LockStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
LockStats.

Objects Property

Current number of objects.

Namespace: BerkeleyDB
Syntax

C#

public uint Objects { get; }

Visual Basic (Declaration)

Public ReadOnly Property Objects As UInteger

Visual C++

public:
property unsigned int Objects {
    unsigned int get ();
}

See Also

LockStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Number of objects steals so far.

**Namespace:** BerkeleyDB
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public ulong ObjectSteals { get; }

Visual Basic (Declaration)

Public Readonly Property ObjectSteals As ULong

Visual C++

public:
property unsigned_long_long ObjectSteals {
    unsigned_long_long get ();
}
See Also

LockStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Object lock granted after wait.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public ulong ObjectWait { get; }

Visual Basic (Declaration)

Public ReadOnly Property ObjectWait As ULong

Visual C++

public:
property unsigned long long ObjectWait {
    unsigned long long get ();
}
See Also

LockStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Partition lock granted without wait.

**Namespace:**  BerkeleyDB  
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public ulong PartitionLockNoWait { get; }

Visual Basic (Declaration)

Public ReadOnly Property PartitionLockNoWait As ULong

Visual C++

public:
property unsigned long long PartitionLockNoWait {
    unsigned long long get ();
}
See Also

LockStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Partition lock granted after wait.

**Namespace:**  BerkeleyDB  
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public ulong PartitionLockWait { get; }

Visual Basic (Declaration)

Public ReadOnly Property PartitionLockWait As ULong

Visual C++

public:
property unsigned long long PartitionLockWait {
    unsigned long long get ();
}
See Also

LockStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Region lock granted without wait.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public ulong RegionNoWait { get; }

Visual Basic (Declaration)

Public ReadOnly Property RegionNoWait As ULong

Visual C++

public:
property unsigned long long RegionNoWait {
unsigned long long get ();
}
See Also

LockStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
LockStats Class  See Also

Region size.

Namespace:  BerkeleyDB
Syntax

**C#**

```csharp
public ulong RegionSize { get; }
```

**Visual Basic (Declaration)**

```vbnet
Public ReadOnly Property RegionSize As ULong
```

**Visual C++**

```cpp
public:
property unsigned long long RegionSize {
  unsigned long long get ();
}
```
See Also

LockStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Region lock granted after wait.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public ulong RegionWait { get; }

Visual Basic (Declaration)

Public ReadOnly Property RegionWait As ULong

Visual C++

public:
property unsigned long long RegionWait {
    unsigned long long get ();
}

See Also

LockStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Transaction timeout.

Namespace: BerkeleyDB
## Syntax

### C#

```csharp
public uint TxnTimeoutLength { get; }
```

### Visual Basic (Declaration)

```vbnet
Public Readonly Property TxnTimeoutLength As UInteger
```

### Visual C++

```cpp
public:
property unsigned int TxnTimeoutLength {
    unsigned int get ();
}
```
See Also

LockStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
LockStats Class  See Also

Number of transaction timeouts.

**Namespace:**  BerkeleyDB  
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public ulong TxnTimeouts { get; }

Visual Basic (Declaration)

Public ReadOnly Property TxnTimeouts As ULong

Visual C++

public:
property unsigned long long TxnTimeouts {
    unsigned long long get ();
}

See Also

LockStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
A class representing configuration parameters for a DatabaseEnvironment’s logging subsystem.

Namespace: BerkeleyDB
### Syntax

**C#**

```csharp
public class LogConfig
```

**Visual Basic (Declaration)**

```vbnet
Public Class LogConfig
```

**Visual C++**

```cpp
public ref class LogConfig
```
Inheritance Hierarchy

System::Object
BerkeleyDB::LogConfig
See Also

LogConfig Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The **LogConfig** type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LogConfig</td>
<td>Initializes a new instance of the LogConfig class</td>
</tr>
<tr>
<td><strong>Name</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>-----------</td>
<td>---------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a> is equal to the current <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <a href="https://docs.microsoft.com/en-us/dotnet/api/system.type">Type</a> of the current instance. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a <a href="https://docs.microsoft.com/en-us/dotnet/api/system.string">String</a> that represents the current <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>AutoRemove</strong></td>
<td>If true, Berkeley DB will automatically remove log files that are no longer needed.</td>
</tr>
<tr>
<td><strong>Dir</strong></td>
<td>The path of a directory to be used as the location of logging files. Log files created by the Log Manager subsystem will be created in this directory.</td>
</tr>
<tr>
<td><strong>ForceSync</strong></td>
<td>If true, Berkeley DB will flush log writes to the backing disk before returning from the write system call, rather than flushing log writes explicitly in a separate system call, as necessary.</td>
</tr>
<tr>
<td><strong>InMemory</strong></td>
<td>If true, maintain transaction logs in memory rather than on disk.</td>
</tr>
<tr>
<td><strong>NoBuffer</strong></td>
<td>If true, turn off system buffering of Berkeley DB log files to avoid double caching.</td>
</tr>
<tr>
<td><strong>ZeroOnCreate</strong></td>
<td>If true, zero all pages of a log file when that log file is created.</td>
</tr>
</tbody>
</table>
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BufferSize</strong></td>
<td>The size of the in-memory log buffer, in bytes.</td>
</tr>
<tr>
<td><strong>FileMode</strong></td>
<td>The absolute file mode for created log files.</td>
</tr>
<tr>
<td><strong>MaxFileSize</strong></td>
<td>The maximum size of a single file in the log, in bytes. Because <strong>Offset</strong> is an unsigned four-byte value, MaxFileSize may not be larger than the maximum unsigned four-byte value.</td>
</tr>
<tr>
<td><strong>RegionSize</strong></td>
<td>The size of the underlying logging area of the Berkeley DB environment, in bytes.</td>
</tr>
</tbody>
</table>
See Also

LogConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
LogConfig Constructor

Initializes a new instance of the `LogConfig` class

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
## Syntax

**C#**

```csharp
public LogConfig()
```

**Visual Basic (Declaration)**

```vbnet
Public Sub New
```

**Visual C++**

```cpp
public:
    LogConfig()
```
See Also

LogConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `LogConfig` type exposes the following members.
**Fields**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AutoRemove</td>
<td>If true, Berkeley DB will automatically remove log files that are no longer needed.</td>
</tr>
<tr>
<td>Dir</td>
<td>The path of a directory to be used as the location of logging files. Log files created by the Log Manager subsystem will be created in this directory.</td>
</tr>
<tr>
<td>ForceSync</td>
<td>If true, Berkeley DB will flush log writes to the backing disk before returning from the write system call, rather than flushing log writes explicitly in a separate system call, as necessary.</td>
</tr>
<tr>
<td>InMemory</td>
<td>If true, maintain transaction logs in memory rather than on disk.</td>
</tr>
<tr>
<td>NoBuffer</td>
<td>If true, turn off system buffering of Berkeley DB log files to avoid double caching.</td>
</tr>
<tr>
<td>ZeroOnCreate</td>
<td>If true, zero all pages of a log file when that log file is created.</td>
</tr>
</tbody>
</table>
See Also

LogConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, Berkeley DB will automatically remove log files that are no longer needed.

**Namespace:**  [BerkeleyDB](https://www.berkeleydb.com)

**Assembly:**  `libdb_dotnet48` (in `libdb_dotnet48.dll`) Version: 4.8.24.0
Syntax

C#

public bool AutoRemove

Visual Basic (Declaration)

Public AutoRemove As Boolean

Visual C++

public:
    bool AutoRemove
Remarks

Automatic log file removal is likely to make catastrophic recovery impossible.

Replication applications will rarely want to configure automatic log file removal as it increases the likelihood a master will be unable to satisfy a client's request for a recent log record.
See Also

LogConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
LogConfig...Dir Field

LogConfig Class  See Also

The path of a directory to be used as the location of logging files. Log files created by the Log Manager subsystem will be created in this directory.

Namespace:  BerkeleyDB
Syntax

C#

public string Dir

Visual Basic (Declaration)

Public Dir As String

Visual C++

public:
String^ Dir
Remarks

If no logging directory is specified, log files are created in the environment home directory. See Berkeley DB File Naming in the Programmer's Reference Guide for more information.

For the greatest degree of recoverability from system or application failure, database files and log files should be located on separate physical devices.

If the database environment already exists when Open(String, DatabaseEnvironmentConfig) is called, the value of Dir must be consistent with the existing environment or corruption can occur.
See Also

LogConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, Berkeley DB will flush log writes to the backing disk before returning from the write system call, rather than flushing log writes explicitly in a separate system call, as necessary.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
**Syntax**

**C#**

```
public bool ForceSync
```

**Visual Basic (Declaration)**

```
Public ForceSync As Boolean
```

**Visual C++**

```
public:
bool ForceSync
```
Remarks

This is only available on some systems (for example, systems supporting the IEEE/ANSI Std 1003.1 (POSIX) standard O_DSYNC flag, or systems supporting the Windows FILE_FLAG_WRITE_THROUGH flag). This flag may result in inaccurate file modification times and other file-level information for Berkeley DB log files. This flag may offer a performance increase on some systems and a performance decrease on others.
See Also

LogConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, maintain transaction logs in memory rather than on disk.

Namespace: BerkeleyDB
**Syntax**

**C#**

public bool InMemory

**Visual Basic (Declaration)**

Public InMemory As Boolean

**Visual C++**

public:

bool InMemory
Remarks

This means that transactions exhibit the ACI (atomicity, consistency, and isolation) properties, but not D (durability); that is, database integrity will be maintained, but if the application or system fails, integrity will not persist. All database files must be verified and/or restored from a replication group master or archival backup after application or system failure.

When in-memory logs are configured and no more log buffer space is available, Berkeley DB methods may throw FullLogBufferException. When choosing log buffer and file sizes for in-memory logs, applications should ensure the in-memory log buffer size is large enough that no transaction will ever span the entire buffer, and avoid a state where the in-memory buffer is full and no space can be freed because a transaction that started in the first log "file" is still active.
See Also

LogConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, turn off system buffering of Berkeley DB log files to avoid double caching.

Namespace: BerkeleyDB
Syntax

C#

public bool NoBuffer

Visual Basic (Declaration)

Public NoBuffer As Boolean

Visual C++

public:
bool NoBuffer
See Also

LogConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, zero all pages of a log file when that log file is created.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool ZeroOnCreate

Visual Basic (Declaration)

Public ZeroOnCreate As Boolean

Visual C++

public:
bool ZeroOnCreate
Remarks

This has shown to provide greater transaction throughput in some environments. The log file will be zeroed by the thread which needs to re-create the new log file. Other threads may not write to the log file while this is happening.
See Also

LogConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `LogConfig` type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a <a href="#">String</a> that represents the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
</tbody>
</table>
See Also

LogConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `LogConfig` type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BufferSize</strong></td>
<td>The size of the in-memory log buffer, in bytes.</td>
</tr>
<tr>
<td><strong>FileMode</strong></td>
<td>The absolute file mode for created log files.</td>
</tr>
<tr>
<td><strong>MaxFileSize</strong></td>
<td>The maximum size of a single file in the log, in bytes.</td>
</tr>
<tr>
<td><strong>Offset</strong></td>
<td>Because <strong>Offset</strong> is an unsigned four-byte value, MaxFileSize may not be larger than the maximum unsigned four-byte value.</td>
</tr>
<tr>
<td><strong>RegionSize</strong></td>
<td>The size of the underlying logging area of the Berkeley DB environment, in bytes.</td>
</tr>
</tbody>
</table>
See Also

LogConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
LogConfig.BufferSize Property

The size of the in-memory log buffer, in bytes.

Namespace: BerkeleyDB
Syntax

C#

public uint BufferSize { get; set; }

Visual Basic (Declaration)

Public Property BufferSize As UInteger

Visual C++

public:
property unsigned int BufferSize {
    unsigned int get ();
    void set (unsigned int value);
}
Remarks

When the logging subsystem is configured for on-disk logging, the default size of the in-memory log buffer is approximately 32KB. Log information is stored in-memory until the storage space fills up or transaction commit forces the information to be flushed to stable storage. In the presence of long-running transactions or transactions producing large amounts of data, larger buffer sizes can increase throughput.

When the logging subsystem is configured for in-memory logging, the default size of the in-memory log buffer is 1MB. Log information is stored in-memory until the storage space fills up or transaction abort or commit frees up the memory for new transactions. In the presence of long-running transactions or transactions producing large amounts of data, the buffer size must be sufficient to hold all log information that can accumulate during the longest running transaction. When choosing log buffer and file sizes for in-memory logs, applications should ensure the in-memory log buffer size is large enough that no transaction will ever span the entire buffer, and avoid a state where the in-memory buffer is full and no space can be freed because a transaction that started in the first log "file" is still active.

If the database environment already exists when `Open(String, DatabaseEnvironmentConfig)` is called, the value of BufferSize will be ignored.
See Also

LogConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The absolute file mode for created log files.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public int FileMode { get; set; }

Visual Basic (Declaration)

Public Property FileMode As Integer

Visual C++

public:
property int FileMode {
   int get ();
   void set (int value);
}

Remarks

This method is only useful for the rare Berkeley DB application that does not control its umask value.

Normally, if Berkeley DB applications set their umask appropriately, all processes in the application suite will have read permission on the log files created by any process in the application suite. However, if the Berkeley DB application is a library, a process using the library might set its umask to a value preventing other processes in the application suite from reading the log files it creates. In this rare case, the DB_ENV->set_lg_filemode() method can be used to set the mode of created log files to an absolute value.
See Also

LogConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
LogConfig.MaxFileSize Property

The maximum size of a single file in the log, in bytes. Because Offset is an unsigned four-byte value, MaxFileSize may not be larger than the maximum unsigned four-byte value.

Namespace: BerkeleyDB
Syntax

C#

```csharp
public uint MaxFileSize { get; set; }
```

Visual Basic (Declaration)

```vbnet
Public Property MaxFileSize As UInteger
```

Visual C++

```cpp
public:
property unsigned int MaxFileSize {
    unsigned int get ();
    void set (unsigned int value);
}
```
Remarks

When the logging subsystem is configured for on-disk logging, the default size of a log file is 10MB.

When the logging subsystem is configured for in-memory logging, the default size of a log file is 256KB. In addition, the configured log buffer size must be larger than the log file size. (The logging subsystem divides memory configured for in-memory log records into "files", as database environments configured for in-memory log records may exchange log records with other members of a replication group, and those members may be configured to store log records on-disk.) When choosing log buffer and file sizes for in-memory logs, applications should ensure the in-memory log buffer size is large enough that no transaction will ever span the entire buffer, and avoid a state where the in-memory buffer is full and no space can be freed because a transaction that started in the first log "file" is still active.


If no size is specified by the application, the size last specified for the database region will be used, or if no database region previously existed, the default will be used.
See Also

LogConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Te size of the underlying logging area of the Berkeley DB environment, in bytes.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public uint RegionSize { get; set; }

Visual Basic (Declaration)

Public Property RegionSize As UInteger

Visual C++

public:
property unsigned int RegionSize {
    unsigned int get ();
    void set (unsigned int value);
}
Remarks

By default, or if the value is set to 0, the default size is approximately 60KB. The log region is used to store filenames, and so may need to be increased in size if a large number of files will be opened and registered with the specified Berkeley DB environment's log manager.

If the database environment already exists when `Open(String, DatabaseEnvironmentConfig)` is called, the value of RegionSize will be ignored.
See Also

LogConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Statistical information about the logging subsystem

**Namespace:**  BerkeleyDB  
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public class LogStats

Visual Basic (Declaration)

Public Class LogStats

Visual C++

public ref class LogStats
Inheritance Hierarchy

System..::..Object
BerkeleyDB..::..LogStats
See Also

LogStats Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The LogStats type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a <a href="#">String</a> that represents the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
</tbody>
</table>
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BufferSize</td>
<td>Log buffer size.</td>
</tr>
<tr>
<td>Bytes</td>
<td>Bytes to log.</td>
</tr>
<tr>
<td>BytesSinceCheckpoint</td>
<td>Bytes to log since checkpoint.</td>
</tr>
<tr>
<td>CurrentFile</td>
<td>Current log file number.</td>
</tr>
<tr>
<td>CurrentOffset</td>
<td>Current log file offset.</td>
</tr>
<tr>
<td>DiskFileNumber</td>
<td>Known on disk log file number.</td>
</tr>
<tr>
<td>DiskOffset</td>
<td>Known on disk log file offset.</td>
</tr>
<tr>
<td>FileSize</td>
<td>Log file size.</td>
</tr>
<tr>
<td>MagicNumber</td>
<td>Log file magic number.</td>
</tr>
<tr>
<td>MaxCommitsPerFlush</td>
<td>Max number of commits in a flush.</td>
</tr>
<tr>
<td>MBytes</td>
<td>Megabytes to log.</td>
</tr>
<tr>
<td>MBytesSinceCheckpoint</td>
<td>Megabytes to log since checkpoint.</td>
</tr>
<tr>
<td>MinCommitsPerFlush</td>
<td>Min number of commits in a flush.</td>
</tr>
<tr>
<td>OverflowWrites</td>
<td>Overflow writes to the log.</td>
</tr>
<tr>
<td>PermissionsMode</td>
<td>Log file permissions mode.</td>
</tr>
<tr>
<td>Reads</td>
<td>Total I/O reads from the log.</td>
</tr>
<tr>
<td>Records</td>
<td>Records entered into the log.</td>
</tr>
<tr>
<td>RegionLockNoWait</td>
<td>Region lock granted without wait.</td>
</tr>
<tr>
<td>RegionLockWait</td>
<td>Region lock granted after wait.</td>
</tr>
<tr>
<td>RegionSize</td>
<td>Region size.</td>
</tr>
<tr>
<td>Syncs</td>
<td>Total syncs to the log.</td>
</tr>
<tr>
<td>Version</td>
<td>Log file version number.</td>
</tr>
<tr>
<td>Writes</td>
<td>Total I/O writes to the log.</td>
</tr>
</tbody>
</table>
See Also

LogStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The LogStats type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a <a href="#">String</a> that represents the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
</tbody>
</table>
See Also

LogStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The LogStats type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BufferSize</td>
<td>Log buffer size.</td>
</tr>
<tr>
<td>Bytes</td>
<td>Bytes to log.</td>
</tr>
<tr>
<td>BytesSinceCheckpoint</td>
<td>Bytes to log since checkpoint.</td>
</tr>
<tr>
<td>CurrentFile</td>
<td>Current log file number.</td>
</tr>
<tr>
<td>CurrentOffset</td>
<td>Current log file offset.</td>
</tr>
<tr>
<td>DiskFileNumber</td>
<td>Known on disk log file number.</td>
</tr>
<tr>
<td>DiskOffset</td>
<td>Known on disk log file offset.</td>
</tr>
<tr>
<td>FileSize</td>
<td>Log file size.</td>
</tr>
<tr>
<td>MagicNumber</td>
<td>Log file magic number.</td>
</tr>
<tr>
<td>MaxCommitsPerFlush</td>
<td>Max number of commits in a flush.</td>
</tr>
<tr>
<td>MBytes</td>
<td>Megabytes to log.</td>
</tr>
<tr>
<td>MBytesSinceCheckpoint</td>
<td>Megabytes to log since checkpoint.</td>
</tr>
<tr>
<td>MinCommitsPerFlush</td>
<td>Min number of commits in a flush.</td>
</tr>
<tr>
<td>OverflowWrites</td>
<td>Overflow writes to the log.</td>
</tr>
<tr>
<td>PermissionsMode</td>
<td>Log file permissions mode.</td>
</tr>
<tr>
<td>Reads</td>
<td>Total I/O reads from the log.</td>
</tr>
<tr>
<td>Records</td>
<td>Records entered into the log.</td>
</tr>
<tr>
<td>RegionLockNoWait</td>
<td>Region lock granted without wait.</td>
</tr>
<tr>
<td>RegionLockWait</td>
<td>Region lock granted after wait.</td>
</tr>
<tr>
<td>RegionSize</td>
<td>Region size.</td>
</tr>
<tr>
<td>Syncs</td>
<td>Total syncs to the log.</td>
</tr>
<tr>
<td>Version</td>
<td>Log file version number.</td>
</tr>
<tr>
<td>Writes</td>
<td>Total I/O writes to the log.</td>
</tr>
</tbody>
</table>
See Also

LogStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
LogStats...:::BufferSize Property

LogStats Class  See Also

Log buffer size.

Namespace:  BerkeleyDB
Syntax

C#

public uint BufferSize { get; }

Visual Basic (Declaration)

Public ReadOnly Property BufferSize As UInteger

Visual C++

public:
property unsigned int BufferSize {
    unsigned int get ();
}
See Also

LogStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Bytes to log.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public uint Bytes { get; }

Visual Basic (Declaration)

Public ReadOnly Property Bytes AsUInteger

Visual C++

public:
property unsigned int Bytes {
    unsigned int get ();
}
See Also

LogStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Bytes to log since checkpoint.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
### Syntax

**C#**

public `uint` BytesSinceCheckpoint { get; }

**Visual Basic (Declaration)**

Public ReadOnly Property BytesSinceCheckpoint As `UInteger`

**Visual C++**

```c++
public:
    property `unsigned int` BytesSinceCheckpoint {
        `unsigned int` get();
    }
```
See Also

LogStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Current log file number.

**Namespace:**  [BerkeleyDB](https://www.oracle.com/technetwork/products/berkeleydb/)

**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

**C#**

```csharp
public uint CurrentFile { get; }
```

**Visual Basic (Declaration)**

```vbnet
Public ReadOnly Property CurrentFile As UInteger
```

**Visual C++**

```cpp
public:
property unsigned int CurrentFile {
    unsigned int get ();
}
```
See Also

LogStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Current log file offset.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

**C#**

```csharp
public uint CurrentOffset { get; }
```

**Visual Basic (Declaration)**

```vbnet
Public ReadOnly Property CurrentOffset AsUInteger
```

**Visual C++**

```c++
public:
property unsigned int CurrentOffset {
    unsigned int get ();
}
```
See Also

LogStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Known on disk log file number.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
**Syntax**

**C#**

public uint DiskFileNumber { get; }

**Visual Basic (Declaration)**

Public ReadOnly Property DiskFileNumber AsUInteger

**Visual C++**

public:
property unsigned int DiskFileNumber {
    unsigned int get ();
}
See Also

LogStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
LogStats Class  See Also

Known on disk log file offset.

Namespace:  BerkeleyDB
Syntax

C#

public uint DiskOffset { get; }

Visual Basic (Declaration)

Public ReadOnly Property DiskOffset As UIInteger

Visual C++

public:
   property unsigned int DiskOffset {
      unsigned int get ();
   }
See Also

LogStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Log file size.

**Namespace:**  BerkeleyDB  
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public uint FileSize { get; }

Visual Basic (Declaration)

Public ReadOnly Property FileSize AsUInteger

Visual C++

public:
    property unsigned int FileSize {
        unsigned int get ();
    }

See Also

LogStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
LogStats Class  See Also

Log file magic number.

Namespace:  BerkeleyDB
Syntax

C#

public uint MagicNumber { get; }

Visual Basic (Declaration)

Public ReadOnly Property MagicNumber As UInteger

Visual C++

public:
property unsigned int MagicNumber {
    unsigned int get ();
}

See Also

LogStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Max number of commits in a flush.

Namespace: BerkeleyDB
Syntax

C#

public uint MaxCommitsPerFlush { get; }

Visual Basic (Declaration)

Public ReadOnly Property MaxCommitsPerFlush AsUInteger

Visual C++

public:
property unsigned int MaxCommitsPerFlush {
unsigned int get ();
}


See Also

LogStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Megabytes to log.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public uint MBytes { get; }

Visual Basic (Declaration)

Public ReadOnly Property MBytes As UInteger

Visual C++

public:
property unsigned int MBytes {
    unsigned int get ();
}

See Also

LogStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
MBytesSinceCheckpoint Property

LogStats Class  See Also

Megabytes to log since checkpoint.

Namespace:  BerkeleyDB
Syntax

C#

public uint MBytesSinceCheckpoint { get; }

Visual Basic (Declaration)

Public ReadOnly Property MBytesSinceCheckpoint AsUInteger

Visual C++

public:
property unsigned int MBytesSinceCheckpoint {
        unsigned int get ();
}


See Also

LogStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Min number of commits in a flush.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
**Syntax**

**C#**

```csharp
public uint MinCommitsPerFlush { get; }
```

**Visual Basic (Declaration)**

```vbnet
Public Readonly Property MinCommitsPerFlush AsUInteger
```

**Visual C++**

```cpp
public:
  property unsigned int MinCommitsPerFlush {
    unsigned int get ();
  }
```
See Also

LogStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Overflow writes to the log.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
### Syntax

**C#**

```csharp
public ulong OverflowWrites { get; }
```

**Visual Basic (Declaration)**

```vbnet
Public ReadOnly Property OverflowWrites As ULong
```

**Visual C++**

```cpp
public:
property unsigned long long OverflowWrites {
    unsigned long long get ();
}
```
See Also

LogStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Log file permissions mode.

Namespace: BerkeleyDB
Syntax

C#

public int PermissionsMode { get; }

Visual Basic (Declaration)

Public ReadOnly Property PermissionsMode As Integer

Visual C++

public:
property int PermissionsMode {
    int get ();
}


See Also

LogStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation

LogStats...:::Reads Property

**LogStats Class**  **See Also**

Total I/O reads from the log.

**Namespace:**  **BerkeleyDB**

**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public ulong Reads { get; }

Visual Basic (Declaration)

Public ReadOnly Property Reads As ULong

Visual C++

public:
property unsigned long long Reads {
    unsigned long long get ();
}

See Also

LogStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Records entered into the log.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public ulong Records { get; }

Visual Basic (Declaration)

Public Readonly Property Records As ULong

Visual C++

public:
property unsigned long long Records {
    unsigned long long get ();
}

See Also

LogStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Region lock granted without wait.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
### Syntax

#### C#

```csharp
public ulong RegionLockNoWait { get; }
```

#### Visual Basic (Declaration)

```vbnet
Public ReadOnly Property RegionLockNoWait As ULong
```

#### Visual C++

```c++
public:
property unsigned long long RegionLockNoWait {
    unsigned long long get ();
}
```
See Also

LogStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Region lock granted after wait.

Namespace: BerkeleyDB
Syntax

C#

public ulong RegionLockWait { get; }

Visual Basic (Declaration)

Public ReadOnly Property RegionLockWait As ULong

Visual C++

public:
property unsigned long long RegionLockWait {
    unsigned long long get ();
}
See Also

LogStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
LogStats.RegionSize Property

Region size.

Namespace: BerkeleyDB
Syntax

C#

public ulong RegionSize { get; }

Visual Basic (Declaration)

Public ReadOnly Property RegionSize As ULong

Visual C++

public:
property unsigned_long_long RegionSize {
    unsigned_long_long get ();
}
See Also

LogStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
LogStats...:::Syncs Property

Total syncs to the log.

**Namespace:** BerkeleyDB
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
**Syntax**

**C#**

```csharp
public ulong Syncs { get; }
```

**Visual Basic (Declaration)**

```vbnet
Public ReadOnly Property Syncs As ULong
```

**Visual C++**

```cpp
public:
property unsigned long long Syncs {
    unsigned long long get ();
}
```
See Also

LogStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Log file version number.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public uint Version { get; }

Visual Basic (Declaration)

Public ReadOnly Property Version AsUInteger

Visual C++

public:
property unsigned int Version {
    unsigned int get ();
}


See Also

LogStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
LogStats...:::Writes Property

LogStats Class  See Also

Total I/O writes to the log.

Namespace:  BerkeleyDB
Syntax

C#

class
{
    public ulong Writes { get; }
}

Visual Basic (Declaration)

Public ReadOnly Property Writes As ULong

Visual C++

public:
    property unsigned long long Writes {
        unsigned long long get ();
    }
}
See Also

LogStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
A log sequence number, which specifies a unique location in a log file.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

**C#**

public class LSN

**Visual Basic (Declaration)**

Public Class LSN

**Visual C++**

public ref class LSN
Inheritance Hierarchy

System::Object
BerkeleyDB::LSN
See Also

LSN Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The **LSN** type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSN</td>
<td>Instantiate a new LSN object</td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compare</td>
<td>Compare two LSNs.</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as a hash function for a particular type.</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the <a href="#">Type</a> of the current instance.</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a <a href="#">String</a> that represents the current <a href="#">Object</a>.</td>
</tr>
</tbody>
</table>
## Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LogFileNumber</td>
<td>The log file number.</td>
</tr>
<tr>
<td>Offset</td>
<td>The offset in the log file.</td>
</tr>
</tbody>
</table>
See Also

LSN Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Instantiate a new LSN object

**Namespace:**  BerkeleyDB  
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
# Syntax

**C#**

```csharp
public LSN(
    uint file,
    uint off
)
```

**Visual Basic (Declaration)**

```vbnet
Public Sub New (
    file As UInteger,
    off As UInteger
)
```

**Visual C++**

```cpp
public:
LSN(
    unsigned int file,
    unsigned int off
)
```

## Parameters

**file**
- Type: `System::::UInt32`
- The log file number.

**off**
- Type: `System::::UInt32`
- The offset in the log file.
See Also

LSN Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The LSN type exposes the following members.
# Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LogFileNumber</td>
<td>The log file number.</td>
</tr>
<tr>
<td>Offset</td>
<td>The offset in the log file.</td>
</tr>
</tbody>
</table>
See Also

LSN Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The log file number.

**Namespace:**  [BerkeleyDB](https://www.berkeleydb.com)

**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public  

Visual Basic (Declaration)

Public LogFileNumber As UInteger

Visual C++

public:

unsigned int LogFileNumber
See Also

LSN Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The offset in the log file.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#
public uint Offset

Visual Basic (Declaration)
Public Offset As UInteger

Visual C++
public:
unsigned int Offset
See Also

LSN Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `LSN` type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compare</td>
<td>Compare two LSNs.</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified Object is equal to the current Object.</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as a hash function for a particular type.</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a String that represents the current Object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>
See Also

LSN Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Compare two LSNs.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
## Syntax

### C#

```csharp
public static int Compare(
    LSN lsn1,
    LSN lsn2
)
```

### Visual Basic (Declaration)

```vbnet
Public Shared Function Compare ( _
    lsn1 As LSN, _
    lsn2 As LSN _
) As Integer
```

### Visual C++

```cpp
public:
    static int Compare(
        LSN^ lsn1,  
        LSN^ lsn2  
    )
```

## Parameters

### lsn1

Type: [BerkeleyDB::::LSN](#)

The first LSN to compare

### lsn2

Type: [BerkeleyDB::::LSN](#)

The second LSN to compare

## Return Value

0 if they are equal, 1 if lsn1 is greater than lsn2, and -1 if lsn1 is less than lsn2.
See Also

LSN Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
A class representing configuration parameters for a `DatabaseEnvironment`'s memory pool subsystem.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public class MPoolConfig

Visual Basic (Declaration)

Public Class MPoolConfig

Visual C++

public ref class MPoolConfig
Inheritance Hierarchy

System..:::Object
BerkeleyDB..:::MPoolConfig
See Also

MPoolConfig Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The **MPoolConfig** type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPoolConfig</td>
<td>Initializes a new instance of the MPoolConfig class</td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>SetMaxSequentialWrites</strong></td>
<td>Limit the number of sequential write operations scheduled by the library when flushing dirty pages from the cache.</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a <a href="#">String</a> that represents the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
</tbody>
</table>
## Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CacheSize</td>
<td>The size of the shared memory buffer pool — that is, the cache.</td>
</tr>
<tr>
<td>MaxCacheSize</td>
<td>The maximum cache size.</td>
</tr>
</tbody>
</table>
### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MaxOpenFiles</strong></td>
<td>The number of file descriptors the library will open concurrently when flushing dirty pages from the cache.</td>
</tr>
<tr>
<td><strong>MaxSequentialWrites</strong></td>
<td>The number of sequential write operations scheduled by the library when flushing dirty pages from the cache.</td>
</tr>
<tr>
<td><strong>MMapSize</strong></td>
<td>The maximum file size, in bytes, for a file to be mapped into the process address space. If no value is specified, it defaults to 10MB.</td>
</tr>
<tr>
<td><strong>SequentialWritePause</strong></td>
<td>The number of microseconds the thread of control should pause before scheduling further write operations.</td>
</tr>
</tbody>
</table>
See Also

MPoolConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Initializes a new instance of the `MPoolConfig` class

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public MPoolConfig()

Visual Basic (Declaration)

Public Sub New

Visual C++

public:
MPoolConfig()
See Also

MPoolConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `MPoolConfig` type exposes the following members.
### Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CacheSize</td>
<td>The size of the shared memory buffer pool — that is, the cache.</td>
</tr>
<tr>
<td>MaxCacheSize</td>
<td>The maximum cache size.</td>
</tr>
</tbody>
</table>
See Also

MPoolConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The size of the shared memory buffer pool — that is, the cache.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public CacheInfo CacheSize

Visual Basic (Declaration)

Public CacheSize As CacheInfo

Visual C++

public:
 CacheInfo^ CacheSize
Remarks

The cache should be the size of the normal working data set of the application, with some small amount of additional memory for unusual situations. (Note: the working set is not the same as the number of pages accessed simultaneously, and is usually much larger.)

The default cache size is 256KB, and may not be specified as less than 20KB. Any cache size less than 500MB is automatically increased by 25% to account for buffer pool overhead; cache sizes larger than 500MB are used as specified. The maximum size of a single cache is 4GB on 32-bit systems and 10TB on 64-bit systems. (All sizes are in powers-of-two, that is, 256KB is $2^{18}$ not 256,000.) For information on tuning the Berkeley DB cache size, see Selecting a cache size in the Programmer's Reference Guide.
See Also

MPoolConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The maximum cache size.

Namespace: BerkeleyDB
Syntax

C#

public CacheInfo MaxCacheSize

Visual Basic (Declaration)

Public MaxCacheSize As CacheInfo

Visual C++

public:
CacheInfo^ MaxCacheSize
Remarks

The specified size is rounded to the nearest multiple of the cache region size, which is the initial cache size divided by CacheSize.NCaches. If no value is specified, it defaults to the initial cache size.
See Also

MPoolConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `MPoolConfig` type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a> is equal to the current <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <a href="https://docs.microsoft.com/en-us/dotnet/api/system.type">Type</a> of the current instance. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td><strong>SetMaxSequentialWrites</strong></td>
<td>Limit the number of sequential write operations scheduled by the library when flushing dirty pages from the cache.</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a <a href="https://docs.microsoft.com/en-us/dotnet/api/system.string">String</a> that represents the current <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
</tbody>
</table>
See Also

MPoolConfig Class  
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Limit the number of sequential write operations scheduled by the library when flushing dirty pages from the cache.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
**Syntax**

**C#**

```csharp
public void SetMaxSequentialWrites(
    int maxWrites,
    uint pause
)
```

**Visual Basic (Declaration)**

```vbnet
Public Sub SetMaxSequentialWrites ( _
    maxWrites As Integer, _
    pause As UInteger _
)
```

**Visual C++**

```c++
public:
void SetMaxSequentialWrites( 
    int maxWrites,
    unsigned int pause
)
```

**Parameters**

**maxWrites**
Type: `System::::Int32`
The maximum number of sequential write operations scheduled by the library when flushing dirty pages from the cache, or 0 if there is no limitation on the number of sequential write operations.

**pause**
Type: `System::::UInt32`
The number of microseconds the thread of control should pause before scheduling further write operations. It must be specified as an unsigned 32-bit number of microseconds, limiting the maximum pause to roughly 71 minutes.
See Also

MPoolConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `MPoolConfig` type exposes the following members.
### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MaxOpenFiles</strong></td>
<td>The number of file descriptors the library will open concurrently when flushing dirty pages from the cache.</td>
</tr>
<tr>
<td><strong>MaxSequentialWrites</strong></td>
<td>The number of sequential write operations scheduled by the library when flushing dirty pages from the cache.</td>
</tr>
<tr>
<td><strong>MMapSize</strong></td>
<td>The maximum file size, in bytes, for a file to be mapped into the process address space. If no value is specified, it defaults to 10MB.</td>
</tr>
<tr>
<td><strong>SequentialWritePause</strong></td>
<td>The number of microseconds the thread of control should pause before scheduling further write operations.</td>
</tr>
</tbody>
</table>
See Also

MPoolConfig Class
BerkeleyDB Namespace
Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The number of file descriptors the library will open concurrently when flushing dirty pages from the cache.

**Namespace:**  BerkeleyDB  
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public int MaxOpenFiles { get; set; }

Visual Basic (Declaration)

Public Property MaxOpenFiles As Integer

Visual C++

public:
property int MaxOpenFiles {
    int get ();
    void set (int value);
}
See Also

MPoolConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
MPoolConfig Class  See Also

The number of sequential write operations scheduled by the library when flushing dirty pages from the cache.

Namespace:  BerkeleyDB
Syntax

C#

```csharp
public int MaxSequentialWrites { get; }
```

Visual Basic (Declaration)

```vbnet
Public ReadOnly Property MaxSequentialWrites As Integer
```

Visual C++

```cpp
public:
property int MaxSequentialWrites {
    int get ();
}
```
See Also

MPoolConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The maximum file size, in bytes, for a file to be mapped into the process address space. If no value is specified, it defaults to 10MB.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public uint MMapSize { get; set; }

Visual Basic (Declaration)

Public Property MMapSize AsUInteger

Visual C++

public:
property unsigned int MMapSize {
    unsigned int get () ;
    void set (unsigned int value);
}

Remarks

Files that are opened read-only in the cache (and that satisfy a few other criteria) are, by default, mapped into the process address space instead of being copied into the local cache. This can result in better-than-usual performance because available virtual memory is normally much larger than the local cache, and page faults are faster than page copying on many systems. However, it can cause resource starvation in the presence of limited virtual memory, and it can result in immense process sizes in the presence of large databases.
See Also

MPoolConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The number of microseconds the thread of control should pause before scheduling further write operations.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public uint SequentialWritePause { get; }

Visual Basic (Declaration)

Public ReadOnly Property SequentialWritePause AsUInteger

Visual C++

public:
property unsigned int SequentialWritePause {
    unsigned int get ();
}
See Also

MPoolConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Statistical information about a file in the memory pool

**Namespace:**  [BerkeleyDB](https://www.berkeleydb.com/)

**Assembly:**  `libdb_dotnet48` (in `libdb_dotnet48.dll`) Version: 4.8.24.0
Syntax

C#

public class MPoolFileStats

Visual Basic (Declaration)

Public Class MPoolFileStats

Visual C++

public ref class MPoolFileStats
Inheritance Hierarchy

System:::Object
BerkeleyDB:::MPoolFileStats
See Also

MPoolFileStats Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `MPoolFileSyncStats` type exposes the following members.
### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a <a href="#">String</a> that represents the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
</tbody>
</table>
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FileName</td>
<td>File name.</td>
</tr>
<tr>
<td>MappedPages</td>
<td>Pages from mapped files.</td>
</tr>
<tr>
<td>PagesCreatedInCache</td>
<td>Pages created in the cache.</td>
</tr>
<tr>
<td>PagesInCache</td>
<td>Pages found in the cache.</td>
</tr>
<tr>
<td>PageSize</td>
<td>Page size.</td>
</tr>
<tr>
<td>PagesNotInCache</td>
<td>Pages not found in the cache.</td>
</tr>
<tr>
<td>PagesRead</td>
<td>Pages read in.</td>
</tr>
<tr>
<td>PagesWritten</td>
<td>Pages written out.</td>
</tr>
</tbody>
</table>
See Also

MPoolFileStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The **MPoolFileStats** type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified <code>Object</code> is equal to the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as a hash function for a particular type. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the <code>Type</code> of the current instance. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a <code>String</code> that represents the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
</tbody>
</table>
See Also

MPoolFileStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `MPoolFileStats` type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FileName</td>
<td>File name.</td>
</tr>
<tr>
<td>MappedPages</td>
<td>Pages from mapped files.</td>
</tr>
<tr>
<td>PagesCreatedInCache</td>
<td>Pages created in the cache.</td>
</tr>
<tr>
<td>PagesInCache</td>
<td>Pages found in the cache.</td>
</tr>
<tr>
<td>PageSize</td>
<td>Page size.</td>
</tr>
<tr>
<td>PagesNotInCache</td>
<td>Pages not found in the cache.</td>
</tr>
<tr>
<td>PagesRead</td>
<td>Pages read in.</td>
</tr>
<tr>
<td>PagesWritten</td>
<td>Pages written out.</td>
</tr>
</tbody>
</table>
See Also

MPoolFileStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
MPoolFileStats...:::FileName Property

See Also

File name.

Namespace: BerkeleyDB
Syntax

C#

public string FileName { get; }

Visual Basic (Declaration)

Public ReadOnly Property FileName As String

Visual C++

public:
property String^ FileName {
    String^ get ();
}
}
See Also

MPoolFileStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
MPoolFileStats.MappedPages Property

Pages from mapped files.

Namespace: BerkeleyDB
Syntax

C#

public uint MappedPages { get; }

Visual Basic (Declaration)

Public ReadOnly Property MappedPages AsUInteger

Visual C++

public:
property unsigned int MappedPages {
    unsigned int get ();
}

See Also

MPoolFileStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Pages created in the cache.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public ulong PagesCreatedInCache { get; }

Visual Basic (Declaration)

Public ReadOnly Property PagesCreatedInCache As ULong

Visual C++

public:
property unsigned long long PagesCreatedInCache {
    unsigned long long get ();
}
See Also

MPoolFileStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
MPoolFileStats

::

PagesInCache Property

Namespace: BerkeleyDB
Syntax

C#

public ulong PagesInCache { get; }

Visual Basic (Declaration)

Public ReadOnly Property PagesInCache As ULong

Visual C++

public:
property unsigned long long PagesInCache {
unsigned long long get ();
}
See Also

MPoolFileStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
MPoolFileStats Class  See Also

Page size.

Namespace:  BerkeleyDB
Syntax

C#

public uint PageSize { get; }

Visual Basic (Declaration)

Public ReadOnly Property PageSize AsUInteger

Visual C++

public:
property unsigned int PageSize {
    unsigned int get ();
}

See Also

MPoolFileStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Pages not found in the cache.

**Namespace:** [BerkeleyDB](BerkeleyDB)

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

```csharp
public ulong PagesNotInCache { get; }
```

Visual Basic (Declaration)

```vbnet
Public ReadOnly Property PagesNotInCache As ULong
```

Visual C++

```cpp
public:
    property unsigned long long PagesNotInCache {
        unsigned long long get();
    }
```
See Also

MPoolFileStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Pages read in.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

```csharp
public ulong PagesRead { get; }
```

Visual Basic (Declaration)

```
Public ReadOnly Property PagesRead As ULong
```

Visual C++

```cpp
public:
property unsigned long long PagesRead {
    unsigned long long get ()
}
```
See Also

MPoolFileStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Pages written out.

**Namespace:**  BerkeleyDB

**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public ulong PagesWritten { get; }

Visual Basic (Declaration)

Public Readonly Property PagesWritten As ULong

Visual C++

public:
property unsigned long long PagesWritten {
    unsigned long long get ();
}

See Also

MPoolFileStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Statistical information about the memory pool subsystem

**Namespace:** [BerkeleyDB](https://www.oracle.com/berkeley-db-net/index.html)

**Assembly:** `libdb_dotnet48` (in `libdb_dotnet48.dll`) Version: 4.8.24.0
Syntax

C#
public class MPoolStats

Visual Basic (Declaration)
Public Class MPoolStats

Visual C++
public ref class MPoolStats
Inheritance Hierarchy

System..:::.Object
BerkeleyDB..:::.MPoolStats
See Also

MPoolStats Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `MPoolStats` type exposes the following members.
# Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <em>Object</em> is equal to the current <em>Object</em>.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <em>Object</em>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <em>Object</em>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <em>Type</em> of the current instance.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <em>Object</em>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a <em>String</em> that represents the current <em>Object</em>.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <em>Object</em>.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>BlockedOperations</td>
<td>Thread waited on buffer I/O.</td>
</tr>
<tr>
<td>BucketsCheckedDuringAlloc</td>
<td>Buckets checked during allocation.</td>
</tr>
<tr>
<td>CacheRegions</td>
<td>Maximum number of regions.</td>
</tr>
<tr>
<td>CacheSettings</td>
<td>Total cache size and number of regions</td>
</tr>
<tr>
<td>CleanPages</td>
<td>Clean pages.</td>
</tr>
<tr>
<td>CleanPagesEvicted</td>
<td>Clean pages forced from the cache.</td>
</tr>
<tr>
<td>DirtyPages</td>
<td>Dirty pages.</td>
</tr>
<tr>
<td>DirtyPagesEvicted</td>
<td>Dirty pages forced from the cache.</td>
</tr>
<tr>
<td>Files</td>
<td>Stats for files open in the memory pool</td>
</tr>
<tr>
<td>FrozenBuffers</td>
<td>Buffers frozen.</td>
</tr>
<tr>
<td>FrozenBuffersFreed</td>
<td>Frozen buffers freed.</td>
</tr>
<tr>
<td>HashBuckets</td>
<td>Number of hash buckets.</td>
</tr>
<tr>
<td>HashChainSearches</td>
<td>Total hash chain searches.</td>
</tr>
<tr>
<td>HashEntriesSearched</td>
<td>Total hash entries searched.</td>
</tr>
<tr>
<td>HashLockNoWait</td>
<td>Hash lock granted with nowait.</td>
</tr>
<tr>
<td>HashLockWait</td>
<td>Hash lock granted after wait.</td>
</tr>
<tr>
<td>LongestHashChainSearch</td>
<td>Longest hash chain searched.</td>
</tr>
<tr>
<td>MappedPages</td>
<td>Pages from mapped files.</td>
</tr>
<tr>
<td>MaxBucketsCheckedDuringAlloc</td>
<td>Max checked during allocation.</td>
</tr>
<tr>
<td>MaxBufferWrites</td>
<td>Maximum buffers to write.</td>
</tr>
<tr>
<td>MaxBufferWritesSleep</td>
<td>Sleep after writing max buffers.</td>
</tr>
<tr>
<td>MaxHashLockNoWait</td>
<td>Max hash lock granted with nowait.</td>
</tr>
<tr>
<td>MaxHashLockWait</td>
<td>Max hash lock granted after wait.</td>
</tr>
<tr>
<td>MaxMMapSize</td>
<td>Maximum file size for mmap.</td>
</tr>
<tr>
<td>MaxOpenFileDescriptors</td>
<td>Maximum number of open fd's.</td>
</tr>
<tr>
<td>MaxPagesCheckedDuringAlloc</td>
<td>Max checked during allocation.</td>
</tr>
<tr>
<td>Metric</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>--------------------------------------------</td>
</tr>
<tr>
<td>PageAllocations</td>
<td>Number of page allocations.</td>
</tr>
<tr>
<td>Pages</td>
<td>Total number of pages.</td>
</tr>
<tr>
<td>PagesCheckedDuringAlloc</td>
<td>Pages checked during allocation.</td>
</tr>
<tr>
<td>PagesCreatedInCache</td>
<td>Pages created in the cache.</td>
</tr>
<tr>
<td>PagesInCache</td>
<td>Pages found in the cache.</td>
</tr>
<tr>
<td>PageSize</td>
<td>Assumed page size.</td>
</tr>
<tr>
<td>PagesNotInCache</td>
<td>Pages not found in the cache.</td>
</tr>
<tr>
<td>PagesRead</td>
<td>Pages read in.</td>
</tr>
<tr>
<td>PagesTrickled</td>
<td>Pages written by memp_trickle.</td>
</tr>
<tr>
<td>PagesWritten</td>
<td>Pages written out.</td>
</tr>
<tr>
<td>RegionLockNoWait</td>
<td>Region lock granted with nowait.</td>
</tr>
<tr>
<td>RegionLockWait</td>
<td>Region lock granted after wait.</td>
</tr>
<tr>
<td>RegionSize</td>
<td>Region size.</td>
</tr>
<tr>
<td>SyncInterrupted</td>
<td>Number of times sync interrupted.</td>
</tr>
<tr>
<td>ThawedBuffers</td>
<td>Buffers thawed.</td>
</tr>
</tbody>
</table>
See Also

MPoolStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `MPoolStats` type exposes the following members.
# Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a>.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <a href="#">Type</a> of the current instance.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a <a href="#">String</a> that represents the current <a href="#">Object</a>.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Object</a>.)</td>
</tr>
</tbody>
</table>
See Also

MPoolStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `MPoolStats` type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BlockedOperations</td>
<td>Thread waited on buffer I/O.</td>
</tr>
<tr>
<td>BucketsCheckedDuringAlloc</td>
<td>Buckets checked during allocation.</td>
</tr>
<tr>
<td>CacheRegions</td>
<td>Maximum number of regions.</td>
</tr>
<tr>
<td>CacheSettings</td>
<td>Total cache size and number of regions</td>
</tr>
<tr>
<td>CleanPages</td>
<td>Clean pages.</td>
</tr>
<tr>
<td>CleanPagesEvicted</td>
<td>Clean pages forced from the cache.</td>
</tr>
<tr>
<td>DirtyPages</td>
<td>Dirty pages.</td>
</tr>
<tr>
<td>DirtyPagesEvicted</td>
<td>Dirty pages forced from the cache.</td>
</tr>
<tr>
<td>Files</td>
<td>Stats for files open in the memory pool</td>
</tr>
<tr>
<td>FrozenBuffers</td>
<td>Buffers frozen.</td>
</tr>
<tr>
<td>FrozenBuffersFreed</td>
<td>Frozen buffers freed.</td>
</tr>
<tr>
<td>HashBuckets</td>
<td>Number of hash buckets.</td>
</tr>
<tr>
<td>HashChainSearches</td>
<td>Total hash chain searches.</td>
</tr>
<tr>
<td>HashEntriesSearched</td>
<td>Total hash entries searched.</td>
</tr>
<tr>
<td>HashLockNoWait</td>
<td>Hash lock granted with nowait.</td>
</tr>
<tr>
<td>HashLockWait</td>
<td>Hash lock granted after wait.</td>
</tr>
<tr>
<td>LongestHashChainSearch</td>
<td>Longest hash chain searched.</td>
</tr>
<tr>
<td>MappedPages</td>
<td>Pages from mapped files.</td>
</tr>
<tr>
<td>MaxBucketsCheckedDuringAlloc</td>
<td>Max checked during allocation.</td>
</tr>
<tr>
<td>MaxBufferWrites</td>
<td>Maximum buffers to write.</td>
</tr>
<tr>
<td>MaxBufferWritesSleep</td>
<td>Sleep after writing max buffers.</td>
</tr>
<tr>
<td>MaxHashLockNoWait</td>
<td>Max hash lock granted with nowait.</td>
</tr>
<tr>
<td>MaxHashLockWait</td>
<td>Max hash lock granted after wait.</td>
</tr>
<tr>
<td>MaxMMMapSize</td>
<td>Maximum file size for mmap.</td>
</tr>
<tr>
<td>MaxOpenFileDescriptors</td>
<td>Maximum number of open fd's.</td>
</tr>
<tr>
<td>MaxPagesCheckedDuringAlloc</td>
<td>Max checked during allocation.</td>
</tr>
<tr>
<td>Metric</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-------------------------------------------------------</td>
</tr>
<tr>
<td>PageAllocations</td>
<td>Number of page allocations.</td>
</tr>
<tr>
<td>Pages</td>
<td>Total number of pages.</td>
</tr>
<tr>
<td>PagesCheckedDuring Alloc</td>
<td>Pages checked during allocation.</td>
</tr>
<tr>
<td>PagesCreatedInCache</td>
<td>Pages created in the cache.</td>
</tr>
<tr>
<td>PagesInCache</td>
<td>Pages found in the cache.</td>
</tr>
<tr>
<td>PageSize</td>
<td>Assumed page size.</td>
</tr>
<tr>
<td>PagesNotInCache</td>
<td>Pages not found in the cache.</td>
</tr>
<tr>
<td>PagesRead</td>
<td>Pages read in.</td>
</tr>
<tr>
<td>PagesTrickled</td>
<td>Pages written by memp_trickle.</td>
</tr>
<tr>
<td>PagesWritten</td>
<td>Pages written out.</td>
</tr>
<tr>
<td>RegionLockNoWait</td>
<td>Region lock granted with nowait.</td>
</tr>
<tr>
<td>RegionLockWait</td>
<td>Region lock granted after wait.</td>
</tr>
<tr>
<td>RegionSize</td>
<td>Region size.</td>
</tr>
<tr>
<td>SyncInterrupted</td>
<td>Number of times sync interrupted.</td>
</tr>
<tr>
<td>ThawedBuffers</td>
<td>Buffers thawed.</td>
</tr>
</tbody>
</table>
See Also

MPoolStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
MPoolStats Block Operations Property

Thread waited on buffer I/O.

Namespace: BerkeleyDB
Syntax

C#

public ulong BlockedOperations { get; }

Visual Basic (Declaration)

Public ReadOnly Property BlockedOperations As ULong

Visual C++

public:
property unsigned long long BlockedOperations {
    unsigned long long get ();
}


See Also

MPoolStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
MPoolStats Class

See Also

Buckets checked during allocation.

Namespace: BerkeleyDB
Syntax

C#

public ulong BucketsCheckedDuringAlloc { get; }

Visual Basic (Declaration)

Public ReadOnly Property BucketsCheckedDuringAlloc As ULong

Visual C++

public:
    property unsigned long long BucketsCheckedDuringAlloc {
        unsigned long long get ();
    }

See Also

MPoolStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
MPoolStats CacheRegions Property

Maximum number of regions.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public uint CacheRegions { get; }

Visual Basic (Declaration)

Public ReadOnly Property CacheRegions As UInteger

Visual C++

public:
property unsigned int CacheRegions {
    unsigned int get ();
}

See Also

MPoolStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
MPoolStats Class

See Also

Total cache size and number of regions

Namespace: BerkeleyDB
Syntax

C#

public CacheInfo CacheSettings { get; }

Visual Basic (Declaration)

Public ReadOnly Property CacheSettings As CacheInfo

Visual C++

public:
property CacheInfo^ CacheSettings {
    CacheInfo^ get ();
}

See Also

MPoolStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Clean pages.

Namespace: BerkeleyDB
## Syntax

**C#**

```csharp
public uint CleanPages { get; }
```

**Visual Basic (Declaration)**

```vbnet
Public ReadOnly Property CleanPages As UInteger
```

**Visual C++**

```c++
public:
property unsigned int CleanPages {
    unsigned int get ();
}
```
See Also

MPoolStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
MPoolStats..:..:CleanPagesEvicted Property

Clean pages forced from the cache.

Namespace: BerkeleyDB
Syntax

**C#**

public ulong CleanPagesEvicted { get; }

**Visual Basic (Declaration)**

Public ReadOnly Property CleanPagesEvicted As ULong

**Visual C++**

public:
property unsigned long long CleanPagesEvicted {
    unsigned long long get ();
}
See Also

MPoolStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Dirty pages.

Namespace: BerkeleyDB
Syntax

C#

public uint DirtyPages { get; }

Visual Basic (Declaration)

Public ReadOnly Property DirtyPages As UInteger

Visual C++

public:
property unsigned int DirtyPages {
    unsigned int get ();
}

See Also

MPoolStats Class
BerkeleyDBNamespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Dirty pages forced from the cache.

**Namespace:** [BerkeleyDB](https://github.com/berkeley-db-net)

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
**Syntax**

**C#**

```csharp
public ulong DirtyPagesEvicted { get; }
```

**Visual Basic (Declaration)**

```vbnet
Public ReadOnly Property DirtyPagesEvicted As ULong
```

**Visual C++**

```csharp
public:
property unsigned long long DirtyPagesEvicted {
    unsigned long long get ();
}
```
See Also

MPoolStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Stats for files open in the memory pool

**Namespace:**  [BerkeleyDB](http)

**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public List<MPoolFileStats> Files { get; }

Visual Basic (Declaration)

Public ReadOnly Property Files As List(Of MPoolFileStats)

Visual C++

public:
property List<MPoolFileStats>^ Files {
List<MPoolFileStats>^ get ();
}

See Also

MPoolStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Buffers frozen.


**Assembly:** `libdb_dotnet48` (in `libdb_dotnet48.dll`) Version: 4.8.24.0
Syntax

C#

public ulong FrozenBuffers { get; }

Visual Basic (Declaration)

Public ReadOnly Property FrozenBuffers As ULong

Visual C++

public:
property unsigned long long FrozenBuffers {
    unsigned long long get ();
}

See Also

MPoolStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Frozen buffers freed.

Namespace:  BerkeleyDB
Syntax

C#

public ulong FrozenBuffersFreed { get; }

Visual Basic (Declaration)

Public ReadOnly Property FrozenBuffersFreed As ULong

Visual C++

public:
property unsigned long long FrozenBuffersFreed {
    unsigned long long get ();
}
See Also

MPoolStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Number of hash buckets.

**Namespace:**  [BerkeleyDB](https://docs.oracle.com/cd/E19884-01/820-1114/)  
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public uint HashBuckets { get; }

Visual Basic (Declaration)

Public ReadOnly Property HashBuckets As UInteger

Visual C++

public:
property unsigned int HashBuckets {
    unsigned int get ();
}

See Also

MPoolStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
MPoolStats..:.HashChainSearches Property

Total hash chain searches.

Namespace:  BerkeleyDB
Syntax

C#

public uint HashChainSearches { get; }

Visual Basic (Declaration)

Public ReadOnly Property HashChainSearches AsUInteger

Visual C++

public:
property unsigned int HashChainSearches {
    unsigned int get ();
}

See Also

MPoolStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation
MPoolStats..::.HashEntriesSearched Property

**MPoolStats Class**  See Also

Total hash entries searched.

**Namespace:**  BerkeleyDB

**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
**Syntax**

**C#**

```csharp
public ulong HashEntriesSearched { get; }
```

**Visual Basic (Declaration)**

```vbnet
Public ReadOnly Property HashEntriesSearched As ULong
```

**Visual C++**

```cpp
public:
    property unsigned long long HashEntriesSearched {
        unsigned long long get ();
    }
```
See Also

MPoolStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Hash lock granted with nowait.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public ulong HashLockNoWait { get; }

Visual Basic (Declaration)

Public ReadOnly Property HashLockNoWait As ULong

Visual C++

public:
    property unsigned long long HashLockNoWait {
        unsigned long long get ();
    }
See Also

MPoolStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Hash lock granted after wait.

Namespace: BerkeleyDB
Syntax

C#

public ulong HashLockWait { get; }

Visual Basic (Declaration)

Public ReadOnly Property HashLockWait As ULong

Visual C++

public:
property unsigned long long HashLockWait {
    unsigned long long get ();
}


See Also

MPoolStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation

MPoolStats::{::LongestHashChainSearch Property

MPoolStats Class  See Also

Longest hash chain searched.

Namespace:  BerkeleyDB
Syntax

C#

public uint LongestHashChainSearch { get; }

Visual Basic (Declaration)

Public ReadOnly Property LongestHashChainSearch AsUInteger

Visual C++

public:
property unsigned int LongestHashChainSearch {
    unsigned int get();
}
See Also

MPoolStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Pages from mapped files.

**Namespace:**  BerkeleyDB

**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public uint MappedPages { get; }

Visual Basic (Declaration)

Public ReadOnly Property MappedPages As UInteger

Visual C++

public:
property unsigned int MappedPages {
    unsigned int get ();
}
See Also

MPoolStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
MPoolStats Class  See Also

Max checked during allocation.

Namespace:  BerkeleyDB
Syntax

C#

public ulong MaxBucketsCheckedDuringAlloc { get; }

Visual Basic (Declaration)

Public ReadOnly Property MaxBucketsCheckedDuringAlloc As ULong

Visual C++

public:
    property unsigned long long MaxBucketsCheckedDuringAlloc {
        unsigned long long get ();
    }
See Also

MPoolStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
MPoolStats Class  See Also

Maximum buffers to write.

**Namespace:**  BerkeleyDB
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public int MaxBufferWrites { get; }

Visual Basic (Declaration)

Public ReadOnly Property MaxBufferWrites As Integer

Visual C++

public:
property int MaxBufferWrites {
    int get ();
}

See Also

MPoolStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
MPoolStats Class  See Also

Sleep after writing max buffers.

Namespace:  BerkeleyDB
Syntax

C#

class MyClass
{
    public uint MaxBufferWritesSleep { get; }
}

Visual Basic (Declaration)

Public ReadOnly Property MaxBufferWritesSleep As UInteger

Visual C++

public:
    property unsigned int MaxBufferWritesSleep {
        unsigned int get ();
    }
}
See Also

MPoolStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Max hash lock granted with nowait.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public ulong MaxHashLockNoWait { get; }

Visual Basic (Declaration)

Public ReadOnly Property MaxHashLockNoWait As ULong

Visual C++

public:
property unsigned long long MaxHashLockNoWait {
  unsigned long long get ();
}

See Also

MPoolStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Max hash lock granted after wait.

Namespace: BerkeleyDB
Syntax

C#

public ulong MaxHashLockWait { get; }

Visual Basic (Declaration)

Public ReadOnly Property MaxHashLockWait As ULong

Visual C++

public:
property unsigned long long MaxHashLockWait {
    unsigned long long get ();
}
See Also

MPoolStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Maximum file size for mmap.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public ulong MaxMMapSize { get; }

Visual Basic (Declaration)

Public ReadOnly Property MaxMMapSize As ULong

Visual C++

public:
property unsigned long long MaxMMapSize {
    unsigned long long get ();
}

See Also

MPoolStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Maximum number of open fd's.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

```csharp
public int MaxOpenFileDescriptors { get; }
```

Visual Basic (Declaration)

```vbnet
Public ReadOnly Property MaxOpenFileDescriptors As Integer
```

Visual C++

```cpp
public:
property int MaxOpenFileDescriptors {
    int get ();
}
```
See Also

MPoolStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Max checked during allocation.

**Namespace:** [BerkeleyDB](#)  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public ulong MaxPagesCheckedDuringAlloc { get; }

Visual Basic (Declaration)

Public ReadOnly Property MaxPagesCheckedDuringAlloc As ULong

Visual C++

public:
property unsigned long long MaxPagesCheckedDuringAlloc {
unsigned long long get ();
}
See Also

MPoolStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Number of page allocations.

**Namespace:** BerkeleyDB
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public ulong PageAllocations { get; }

Visual Basic (Declaration)

Public ReadOnly Property PageAllocations As ULong

Visual C++

public:
property unsigned long long PageAllocations {
    unsigned long long get ();
}
See Also

MPoolStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
MPoolStats Class  See Also

Total number of pages.

Namespace: BerkeleyDB
Syntax

C#

public uint Pages { get; }

Visual Basic (Declaration)

Public ReadOnly Property Pages As UInteger

Visual C++

public:
property unsigned int Pages {
    unsigned int get ();
}

See Also

MPoolStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Pages checked during allocation.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public ulong PagesCheckedDuringAlloc { get; }

Visual Basic (Declaration)

Public ReadOnly Property PagesCheckedDuringAlloc As ULong

Visual C++

public:
property unsigned long long PagesCheckedDuringAlloc {
unsigned long long get();
}


See Also

MPoolStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Pages created in the cache.

*Namespace:* [BerkeleyDB](#)  
*Assembly:* libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public ulong PagesCreatedInCache { get; }

Visual Basic (Declaration)

Public ReadOnly Property PagesCreatedInCache As ULong

Visual C++

public:
property unsigned long long PagesCreatedInCache {
    unsigned long long get();
}
See Also

MPoolStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
MPoolStats Class | See Also

Pages found in the cache.

**Namespace:** [BerkeleyDB](https://example.com)

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public ulong PagesInCache { get; }

Visual Basic (Declaration)

Public ReadOnly Property PagesInCache As ULong

Visual C++

public:
property unsigned long long PagesInCache {
    unsigned long long get ();
}

See Also

MPoolStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Assumed page size.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
**Syntax**

**C#**

```csharp
public uint PageSize { get; }
```

**Visual Basic (Declaration)**

```vbnet
Public ReadOnly Property PageSize AsUInteger
```

**Visual C++**

```c++
public:
property unsigned int PageSize {
    unsigned int get ();
}
```
See Also

MPoolStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Pages not found in the cache.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public ulong PagesNotInCache { get; }

Visual Basic (Declaration)

Public ReadOnly Property PagesNotInCache As ULong

Visual C++

public:
property unsigned long long PagesNotInCache {
    unsigned long long get ();
}

See Also

MPoolStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
MPoolStats Class  See Also

Pages read in.

**Namespace:**  BerkeleyDB
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
## Syntax

### C#

```csharp
public ulong PagesRead { get; }
```

### Visual Basic (Declaration)

```vbnet
Public ReadOnly Property PagesRead As ULong
```

### Visual C++

```cpp
public:
    property unsigned long long PagesRead {
        unsigned long long get ();
    }
```
See Also

MPoolStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
MPoolStats Class

See Also

Pages written by memp_trickle.

Namespace: BerkeleyDB
Syntax

C#

public ulong PagesTrickled { get; }

Visual Basic (Declaration)

Public ReadOnly Property PagesTrickled As ULong

Visual C++

public:
property unsigned long long PagesTrickled {
    unsigned long long get ();
}

See Also

MPoolStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Pages written out.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public ulong PagesWritten { get; }

Visual Basic (Declaration)

Public ReadOnly Property PagesWritten As ULong

Visual C++

public: 
property unsigned long long PagesWritten {
    unsigned long long get ();
    }
See Also

MPoolStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Region lock granted with nowait.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public ulong RegionLockNoWait { get; }

Visual Basic (Declaration)

Public ReadOnly Property RegionLockNoWait As ULong

Visual C++

public:
property unsigned long long RegionLockNoWait {
    unsigned long long get ();
}
See Also

MPoolStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Region lock granted after wait.

Namespace: BerkeleyDB
C#

public ulong RegionLockWait { get; }

Visual Basic (Declaration)

Public ReadOnly Property RegionLockWait As ULong

Visual C++

public:
property unsigned long long RegionLockWait {
    unsigned long long get ();
}


See Also

MPoolStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
MPoolStats class

See Also

Region size.

Namespace: BerkeleyDB
Syntax

C#

public  _ulong  RegionSize { get; }  

Visual Basic (Declaration)

Public  ReadOnly  Property  RegionSize  As  ULong  

Visual C++

public:
property  unsigned long long  RegionSize {
    unsigned long long  get ();
}
See Also

MPoolStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Number of times sync interrupted.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
**Syntax**

**C#**

```csharp
public ulong SyncInterrupted { get; }
```

**Visual Basic (Declaration)**

```vbnet
Public ReadOnly Property SyncInterrupted As ULong
```

**Visual C++**

```cpp
public:
    property unsigned long long SyncInterrupted {
        unsigned long long get ();
    }
```
See Also

MPoolStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Buffers thawed.

**Namespace:**  BerkeleyDB  
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
**Syntax**

**C#**

```csharp
public ulong ThawedBuffers { get; }
```

**Visual Basic (Declaration)**

```vbnet
Public ReadOnly Property ThawedBuffers As ULong
```

**Visual C++**

```cpp
public:
    property unsigned long long ThawedBuffers {
        unsigned long long get ();
    }
```
See Also

MPoolStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
A class providing access to multiple DatabaseEntry objects.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public class MultipleDatabaseEntry : IEnumerable<DatabaseEntry>, IEnumerable

Visual Basic (Declaration)

Public Class MultipleDatabaseEntry
    Implements IEnumerable(Of DatabaseEntry), IEnumerable

Visual C++

public ref class MultipleDatabaseEntry : IEnumerable<DatabaseEntry^>, IEnumerable

IEnumerable
Inheritance Hierarchy

System..:::Object
BerkeleyDB..:::MultipleDatabaseEntry
See Also

MultipleDatabaseEntry Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `MultipleDatabaseEntry` type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <em>Object</em> is equal to the current <em>Object</em>. (Inherited from <em>Object</em>.)*</td>
</tr>
<tr>
<td><strong>GetEnumerator</strong></td>
<td>DatabaseEntry objects represented by the MultipleDatabaseEntry.</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <em>Object</em>.)*</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the Type of the current instance. (Inherited from <em>Object</em>.)*</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a String that represents the current <em>Object</em>. (Inherited from <em>Object</em>.)*</td>
</tr>
</tbody>
</table>
See Also

MultipleDatabaseEntry Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `MultipleDatabaseEntry` type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetEnumerator</strong> <a href="#">DatabaseEntry</a> objects represented by the <a href="#">MultipleDatabaseEntry</a>.</td>
<td></td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a <a href="#">String</a> that represents the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
</tbody>
</table>
See Also

MultipleDatabaseEntry Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
MultipleDatabaseEntry..::.GetEnumerator Method

Return an enumerator which iterates over all DatabaseEntry objects represented by the MultipleDatabaseEntry.

Namespace: BerkeleyDB
Syntax

C#

```csharp
public IEnumerator<DatabaseEntry> GetEnumerator()
```

Visual Basic (Declaration)

```vbnet
Public Function GetEnumerator As IEnumerator(Of DatabaseEntry)
```

Visual C++

```cpp
public:
virtual IEnumerator<DatabaseEntry> GetEnumerator()^>^ GetEnumerator() sealed
```

Return Value

An enumerator for the `MultipleDatabaseEntry`

Implements

```csharp
IEnumerator<Of (T)>::GetEnumerator()
```
See Also

MultipleDatabaseEntry Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
A class providing access to multiple key/data pairs.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public class MultipleKeyDatabaseEntry : IEnumerable<KeyValuePair<
    DatabaseEntry, IEnumerable<
    DatabaseEntry> IEnumerable> IEnumerable

Visual Basic (Declaration)

Public Class MultipleKeyDatabaseEntry _
    Implements IEnumerable(Of KeyValuePair(Of DatabaseEntry, DatabaseEntry) IEnumera-
    IEnumerable> IEnumerable

Visual C++

public ref class MultipleKeyDatabaseEntry : IEnumerable<
    KeyValuePair<
    IEnumerable<
    DatabaseEntry IEnumerab
Inheritance Hierarchy

System...:::Object
BerkeleyDB...::.MultipleKeyDatabaseEntry
See Also

MultipleKeyDatabaseEntry Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `MultipleKeyDatabaseEntry` type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a> is equal to the current <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetEnumerator</strong></td>
<td>Return an enumerator which iterates over all <a href="https://docs.microsoft.com/en-us/dotnet/api/system.data.jdbc.databaseentry">DatabaseEntry</a> pairs represented by the <a href="https://docs.microsoft.com/en-us/dotnet/api/system.data.jdbc.multiplekeydatabaseentry">MultipleKeyDatabaseEntry</a>.</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <a href="https://docs.microsoft.com/en-us/dotnet/api/system.type">Type</a> of the current instance. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a <a href="https://docs.microsoft.com/en-us/dotnet/api/system.string">String</a> that represents the current <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
</tbody>
</table>
See Also

MultipleKeyDatabaseEntry Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `MultipleKeyDatabaseEntry` type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a>.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetEnumerator</strong> <strong>DatabaseEntry</strong></td>
<td>Return an enumerator which iterates over all <a href="#">MultipleKeyDatabaseEntry</a> pairs represented by the</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a <a href="#">String</a> that represents the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
</tbody>
</table>
See Also

MultipleKeyDatabaseEntry Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
MultipleKeyDatabaseEntry Class  See Also

Return an enumerator which iterates over all DatabaseEntry pairs represented by the MultipleKeyDatabaseEntry.

Namespace:  BerkeleyDB
Syntax

C#
public IEnumerator<KeyValuePair<DatabaseEntry, DatabaseEntry>> GetEnumerator

Visual Basic (Declaration)
Public Function GetEnumerator As IEnumerator(Of KeyValuePair(Of DatabaseEntry, DatabaseEntry))

Visual C++

public:
virtual IEnumerator<KeyValuePair<DatabaseEntry^, DatabaseEntry^>>^ GetEnumerator

Return Value

An enumerator for the MultipleDatabaseEntry

Implements

IEnumerable<(Of <(T)>),::GetEnumerator()()
See Also

MultipleKeyDatabaseEntry Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
A class representing configuration parameters for a **DatabaseEnvironment**’s mutex subsystem.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#
public class MutexConfig

Visual Basic (Declaration)
Public Class MutexConfig

Visual C++
public ref class MutexConfig
Inheritance Hierarchy

System::Object
BerkeleyDB::MutexConfig
See Also

MutexConfig Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `MutexConfig` type exposes the following members.
### Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MutexConfig</td>
<td>Initializes a new instance of the MutexConfig class</td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a <a href="#">String</a> that represents the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
</tbody>
</table>
# Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Alignment</strong></td>
<td>The mutex alignment, in bytes.</td>
</tr>
<tr>
<td><strong>Increment</strong></td>
<td>Configure the number of additional mutexes to allocate.</td>
</tr>
<tr>
<td><strong>MaxMutexes</strong></td>
<td>The total number of mutexes to allocate.</td>
</tr>
<tr>
<td><strong>NumTestAndSetSpins</strong></td>
<td>The number of spins test-and-set mutexes should execute before blocking.</td>
</tr>
</tbody>
</table>
See Also

MutexConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
MutexConfig Constructor

Initializes a new instance of the MutexConfig class

Namespace: BerkeleyDB
Syntax

C#

public MutexConfig()

Visual Basic (Declaration)

Public Sub New

Visual C++

public:
MutexConfig()
See Also

MutexConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The **MutexConfig** type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a> is equal to the current <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the Type of the current instance. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a String that represents the current <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
</tbody>
</table>
See Also

MutexConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `MutexConfig` type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Alignment</strong></td>
<td>The mutex alignment, in bytes.</td>
</tr>
<tr>
<td><strong>Increment</strong></td>
<td>Configure the number of additional mutexes to allocate.</td>
</tr>
<tr>
<td><strong>MaxMutexes</strong></td>
<td>The total number of mutexes to allocate.</td>
</tr>
<tr>
<td><strong>NumTestAndSetSpins</strong></td>
<td>The number of spins test-and-set mutexes should execute before blocking.</td>
</tr>
</tbody>
</table>
See Also

 MutexConfig Class
 BerkeleyDB Namespace

 Report Feedback on this item in the Oracle Technology Network Forum

 Copyright (c) 1996-2009 Oracle. All rights reserved.
The mutex alignment, in bytes.

**Namespace:** [BerkeleyDB](https://www.berkeleydb.com/)

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
**Syntax**

**C#**

```csharp
public uint Alignment { get; set; }
```

**Visual Basic (Declaration)**

Public Property Alignment AsUInteger

**Visual C++**

```cpp
public:
property unsigned int Alignment {
    unsigned int get ();
    void set (unsigned int value);
}
```
Remarks

It is sometimes advantageous to align mutexes on specific byte boundaries in order to minimize cache line collisions. Alignment specifies an alignment for mutexes allocated by Berkeley DB.

If the database environment already exists when `Open(String, DatabaseEnvironmentConfig)` is called, the value of Alignment will be ignored.
See Also

MutexConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Configure the number of additional mutexes to allocate.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public uint Increment { get; set; }

Visual Basic (Declaration)

Public Property Increment AsUInteger

Visual C++

public:
property unsigned int Increment {
    unsigned int get ();
    void set (unsigned int value);
}
Remarks

If both Increment and MaxMutexes are set, the value of Increment will be silently ignored.

If the database environment already exists when Open(String, DatabaseEnvironmentConfig) is called, the value of Increment will be ignored.
See Also

MutexConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
MutexConfig Class  See Also

The total number of mutexes to allocate.

Namespace:  BerkeleyDB
Syntax

C#

```csharp
public uint MaxMutexes { get; set; }
```

Visual Basic (Declaration)

```vbnet
Public Property MaxMutexes As UInteger
```

Visual C++

```cpp
public:
property unsigned int MaxMutexes {
    unsigned int get ();
    void set (unsigned int value);
}
```
Remarks

Berkeley DB allocates a default number of mutexes based on the initial configuration of the database environment. That default calculation may be too small if the application has an unusual need for mutexes (for example, if the application opens an unexpectedly large number of databases) or too large (if the application is trying to minimize its memory footprint). MaxMutexes is used to specify an absolute number of mutexes to allocate.

If both Increment and MaxMutexes are set, the value of Increment will be silently ignored.

If the database environment already exists when Open(String, DatabaseEnvironmentConfig) is called, the value of MaxMutexes will be ignored.
See Also

MutexConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
 MutexConfig.

 NumTestAndSetSpins Property

 MutexConfig Class See Also

 The number of spins test-and-set mutexes should execute before blocking.

 **Namespace:** BerkeleyDB

 **Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public uint NumTestAndSetSpins { get; set; }

Visual Basic (Declaration)

Public Property NumTestAndSetSpins AsUInteger

Visual C++

public:
property unsigned int NumTestAndSetSpins {
    unsigned int get ();
    void set (unsigned int value);
}
See Also

MutexConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Statistical information about the mutex subsystem

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public class MutexStats

Visual Basic (Declaration)

Public Class MutexStats

Visual C++

public ref class MutexStats
Inheritance Hierarchy

System..:::.Object
BerkeleyDB..:::.MutexStats
See Also

MutexStats Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The **MutexStats** type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a <a href="#">String</a> that represents the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
</tbody>
</table>
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Alignment</strong></td>
<td>Mutex alignment</td>
</tr>
<tr>
<td><strong>Available</strong></td>
<td>Available mutexes</td>
</tr>
<tr>
<td><strong>Count</strong></td>
<td>Mutex count</td>
</tr>
<tr>
<td><strong>InUse</strong></td>
<td>Mutexes in use</td>
</tr>
<tr>
<td><strong>MaxInUse</strong></td>
<td>Maximum mutexes ever in use</td>
</tr>
<tr>
<td><strong>RegionNoWait</strong></td>
<td>Region lock granted without wait.</td>
</tr>
<tr>
<td><strong>RegionSize</strong></td>
<td>Region size.</td>
</tr>
<tr>
<td><strong>RegionWait</strong></td>
<td>Region lock granted after wait.</td>
</tr>
<tr>
<td><strong>TASSpins</strong></td>
<td>Mutex test-and-set spins</td>
</tr>
</tbody>
</table>
See Also

MutexStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The MutexStats type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a <a href="#">String</a> that represents the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
</tbody>
</table>
See Also

MutexStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `MutexStats` type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alignment</td>
<td>Mutex alignment</td>
</tr>
<tr>
<td>Available</td>
<td>Available mutexes</td>
</tr>
<tr>
<td>Count</td>
<td>Mutex count</td>
</tr>
<tr>
<td>InUse</td>
<td>Mutexes in use</td>
</tr>
<tr>
<td>MaxInUse</td>
<td>Maximum mutexes ever in use</td>
</tr>
<tr>
<td>RegionNoWait</td>
<td>Region lock granted without wait.</td>
</tr>
<tr>
<td>RegionSize</td>
<td>Region size.</td>
</tr>
<tr>
<td>RegionWait</td>
<td>Region lock granted after wait.</td>
</tr>
<tr>
<td>TASSpins</td>
<td>Mutex test-and-set spins</td>
</tr>
</tbody>
</table>
See Also

MutexStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
MutexStats Alignment Property

Mutex alignment

Namespace: BerkeleyDB
Syntax

C#

public uint Alignment { get; }

Visual Basic (Declaration)

Public ReadOnly Property Alignment As UInteger

Visual C++

public:
property unsigned int Alignment {
    unsigned int get ();
}
See Also

MutexStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Available mutexes

**Namespace:** BerkeleyDB
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

class
{
public uint Available { get; }
}

Visual Basic (Declaration)

Public ReadOnly Property Available As UInteger

Visual C++

public:
property unsigned int Available {
    unsigned int get ();
}


See Also

MutexStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
MutexStats.Count Property

Namespace: BerkeleyDB
Syntax

**C#**

public uint Count { get; }

**Visual Basic (Declaration)**

Public ReadOnly Property Count AsUInteger

**Visual C++**

public:
property unsigned int Count {
    unsigned int get ();
}

See Also

MutexStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
MutexStats Class

See Also

Mutexes in use

*Namespace:* BerkeleyDB

*Assembly:* libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public uint InUse { get; }

Visual Basic (Declaration)

Public ReadOnly Property InUse As UInteger

Visual C++

public:
property unsigned int InUse {
    unsigned int get ();
}

See Also

MutexStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Maximum mutexes ever in use

**Namespace:** [BerkeleyDB](#)

**Assembly:** `libdb_dotnet48` (in `libdb_dotnet48.dll`) Version: 4.8.24.0
Syntax

C#

public uint MaxInUse { get; }

Visual Basic (Declaration)

Public ReadOnly Property MaxInUse AsUInteger

Visual C++

public:
property unsigned int MaxInUse {
    unsigned int get ();
}

See Also

MutexStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Region lock granted without wait.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public ulong RegionNoWait { get; }

Visual Basic (Declaration)

Public ReadOnly Property RegionNoWait As ULong

Visual C++

public:
property unsigned long long RegionNoWait {
  unsigned long long get ();
}
See Also

 MutexStats Class
 BerkeleyDB Namespace

 Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
MutexStats Class  See Also

Region size.

Namespace:  BerkeleyDB
Syntax

C#

public ulong RegionSize { get; }

Visual Basic (Declaration)

Public ReadOnly Property RegionSize As ULong

Visual C++

public:
property unsigned long long RegionSize {
    unsigned long long get ();
}

See Also

MutexStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Region lock granted after wait.

**Namespace:**  BerkeleyDB  
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public ulong RegionWait { get; }

Visual Basic (Declaration)

Public ReadOnly Property RegionWait As ULong

Visual C++

c public:
property unsigned long long RegionWait {
    unsigned long long get ();
}

See Also

MutexStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
MutexStats Class  See Also

Mutex test-and-set spins

Namespace: BerkeleyDB
Syntax

C#

public uint TASSpins { get; }

Visual Basic (Declaration)

Public ReadOnly Property TASSpins As UInteger

Visual C++

public:
property unsigned int TASSpins {
    unsigned int get ();
}

See Also

MutexStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The requested key/data pair did not exist in the database or that start-of- or end-of-file has been reached by a cursor.

**Namespace:** BerkeleyDB
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public class NotFoundException : DatabaseException

Visual Basic (Declaration)

Public Class NotFoundException _
    Inherits DatabaseException

Visual C++

public ref class NotFoundException : public DatabaseException
Inheritance Hierarchy

System:::Object
System:::Exception
BerkeleyDB:::DatabaseException
BerkeleyDB:::NotFoundException
See Also

NotFoundException Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `NotFoundException` type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NotFoundException</td>
<td>Initialize a new instance of the NotFoundException</td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetBaseException</strong></td>
<td>When overridden in a derived class, returns the <a href="#">Exception</a> that is the root cause of one or more subsequent exceptions. (Inherited from <a href="#">Exception</a>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetObjectData</strong></td>
<td>When overridden in a derived class, sets the <a href="#">SerializationInfo</a> with information about the exception. (Inherited from <a href="#">Exception</a>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the runtime type of the current instance. (Inherited from <a href="#">Exception</a>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Creates and returns a string representation of the current exception. (Inherited from <a href="#">Exception</a>.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ErrorCode</td>
<td>The underlying error code from the Berkeley DB C library. (Inherited from <code>DatabaseException</code>.)</td>
</tr>
</tbody>
</table>
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>HelpLink</td>
<td>Gets or sets a link to the help file associated with this exception.</td>
</tr>
<tr>
<td>InnerException</td>
<td>Gets the Exception instance that caused the current exception.</td>
</tr>
<tr>
<td>Source</td>
<td>Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)</td>
</tr>
<tr>
<td>StackTrace</td>
<td>Gets a string representation of the frames on the call stack at the time the current exception was thrown. (Inherited from Exception.)</td>
</tr>
<tr>
<td>TargetSite</td>
<td>Gets the method that throws the current exception.</td>
</tr>
</tbody>
</table>

(Inherited from Exception.)
See Also

NotFoundException Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Initialize a new instance of the `NotFoundException` exception.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public NotFoundException()

Visual Basic (Declaration)

Public Sub New

Visual C++

public:
NotFoundException()
See Also

NotFoundException Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `NotFoundException` type exposes the following members.
<table>
<thead>
<tr>
<th><strong>Name</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ErrorCode</strong></td>
<td>The underlying error code from the Berkeley DB C library. (Inherited from DatabaseException.)</td>
</tr>
</tbody>
</table>
See Also

NotFoundException Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `NotFoundException` type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a> is equal to the current <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.) When overridden in a derived class, returns the <a href="https://docs.microsoft.com/en-us/dotnet/api/system.exception">Exception</a> that is the root cause of one or more subsequent exceptions. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.exception">Exception</a>.)</td>
</tr>
<tr>
<td><strong>GetBaseException</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.) When overridden in a derived class, sets the <a href="https://docs.microsoft.com/en-us/dotnet/api/system.serialization.serializationinfo">SerializationInfo</a> with information about the exception. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.exception">Exception</a>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Gets the runtime type of the current instance. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.exception">Exception</a>.) Creates and returns a string representation of the current exception. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.exception">Exception</a>.)</td>
</tr>
<tr>
<td><strong>GetObjectData</strong></td>
<td></td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td></td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td></td>
</tr>
</tbody>
</table>
See Also

NotFoundException Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The **NotFoundException** type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data</strong></td>
<td>Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from <a href="#">Exception</a>.)</td>
</tr>
<tr>
<td><strong>HelpLink</strong></td>
<td>Gets or sets a link to the help file associated with this exception. (Inherited from <a href="#">Exception</a>.)</td>
</tr>
<tr>
<td><strong>InnerException</strong></td>
<td>Gets the <a href="#">Exception</a> instance that caused the current exception. (Inherited from <a href="#">Exception</a>.)</td>
</tr>
<tr>
<td><strong>Message</strong></td>
<td>Gets a message that describes the current exception. (Inherited from <a href="#">Exception</a>.)</td>
</tr>
<tr>
<td><strong>Source</strong></td>
<td>Gets or sets the name of the application or the object that causes the error. (Inherited from <a href="#">Exception</a>.)</td>
</tr>
<tr>
<td><strong>StackTrace</strong></td>
<td>Gets a string representation of the frames on the call stack at the time the current exception was thrown. (Inherited from <a href="#">Exception</a>.)</td>
</tr>
<tr>
<td><strong>TargetSite</strong></td>
<td>Gets the method that throws the current exception. (Inherited from <a href="#">Exception</a>.)</td>
</tr>
</tbody>
</table>
See Also

NotFoundException Class  
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Specify a Berkeley DB event

**Namespace:** [BerkeleyDB](#)  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public enum NotificationEvent

Visual Basic (Declaration)

Public Enumeration NotificationEvent

Visual C++

public enum class NotificationEvent
### Members

<table>
<thead>
<tr>
<th>Member name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PANIC</td>
<td>The database environment has failed.</td>
</tr>
<tr>
<td>REP_CLIENT</td>
<td>The local site is now a replication client.</td>
</tr>
<tr>
<td>REP_ELECTED</td>
<td>The local replication site has just won an election.</td>
</tr>
<tr>
<td></td>
<td>The local site is now the master site of its replication group. It is the application's responsibility to begin acting as the master environment.</td>
</tr>
<tr>
<td>REP_MASTER</td>
<td>The replication group of which this site is a member has just established a new master; the local site is not the new master. The event_info parameter to the EventNotifyDelegate stores an integer containing the environment ID of the new master.</td>
</tr>
<tr>
<td>REP_NEWMASTER</td>
<td>The replication manager did not receive enough acknowledgements (based on the acknowledgement policy configured with RepMgrAckPolicy) to ensure a transaction's durability within the replication group. The transaction will be flushed to the master's local disk storage for durability.</td>
</tr>
<tr>
<td>REP_PERM_FAILED</td>
<td>The client has completed startup synchronization and is now processing live log records received from the master.</td>
</tr>
<tr>
<td>WRITE_FAILED</td>
<td>A Berkeley DB write to stable storage failed.</td>
</tr>
</tbody>
</table>
See Also

BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
This version of Berkeley DB is unable to upgrade a given database.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public class OldVersionException : DatabaseException

Visual Basic (Declaration)

Public Class OldVersionException
    Inherits DatabaseException

Visual C++

public ref class OldVersionException : public DatabaseException
Inheritance Hierarchy

System..::.Object
  System..::.Exception
    BerkeleyDB..::.DatabaseException
      BerkeleyDB..::.OldVersionException
See Also

OldVersionException Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The **OldVersionException** type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>OldVersionException</code></td>
<td>Initialize a new instance of the OldVersionException</td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="https://docs.oracle.com/javase/8/docs/api/java/lang/Object.html">Object</a> is equal to the current <a href="https://docs.oracle.com/javase/8/docs/api/java/lang/Object.html">Object</a>. (Inherited from <a href="https://docs.oracle.com/javase/8/docs/api/java/lang/Object.html">Object</a>.) When overridden in a derived class, returns the <a href="https://docs.oracle.com/javase/8/docs/api/java/lang/Exception.html">Exception</a> that is the root cause of one or more subsequent exceptions. (Inherited from <a href="https://docs.oracle.com/javase/8/docs/api/java/lang/Exception.html">Exception</a>.)</td>
</tr>
<tr>
<td><strong>GetBaseException</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="https://docs.oracle.com/javase/8/docs/api/java/lang/Object.html">Object</a>.) When overridden in a derived class, sets the <a href="https://docs.oracle.com/javase/8/docs/api/java/io/SerializationInfo.html">SerializationInfo</a> with information about the exception. (Inherited from <a href="https://docs.oracle.com/javase/8/docs/api/java/lang/Exception.html">Exception</a>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Gets the runtime type of the current instance. (Inherited from <a href="https://docs.oracle.com/javase/8/docs/api/java/lang/Exception.html">Exception</a>.) Creates and returns a string representation of the current exception. (Inherited from <a href="https://docs.oracle.com/javase/8/docs/api/java/lang/Exception.html">Exception</a>.)</td>
</tr>
<tr>
<td><strong>GetObjectData</strong></td>
<td></td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td></td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ErrorCode</td>
<td>The underlying error code from the Berkeley DB C library. (Inherited from <code>DatabaseException</code>.)</td>
</tr>
</tbody>
</table>
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data</strong></td>
<td>Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td><strong>HelpLink</strong></td>
<td>Gets or sets a link to the help file associated with this exception. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td><strong>InnerException</strong></td>
<td>Gets the <code>Exception</code> instance that caused the current exception. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td><strong>Message</strong></td>
<td>Gets a message that describes the current exception. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td><strong>Source</strong></td>
<td>Gets or sets the name of the application or the object that causes the error. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td><strong>StackTrace</strong></td>
<td>Gets a string representation of the frames on the call stack at the time the current exception was thrown. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td><strong>TargetSite</strong></td>
<td>Gets the method that throws the current exception. (Inherited from <code>Exception</code>.)</td>
</tr>
</tbody>
</table>
See Also

OldVersionException Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Initialize a new instance of the OldVersionException

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public OldVersionException()

Visual Basic (Declaration)

Public Sub New

Visual C++

public:
OldVersionException()
See Also

OldVersionException Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The OldVersionException type exposes the following members.
**Fields**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ErrorCode</td>
<td>The underlying error code from the Berkeley DB C library. (Inherited from <a href="#">DatabaseException</a>)</td>
</tr>
</tbody>
</table>
See Also

OldVersionException Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The **OldVersionException** type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="System.Object">Object</a> is equal to the current <a href="System.Object">Object</a>. When overridden in a derived class, returns the <a href="System.Exception">Exception</a> that is the root cause of one or more subsequent exceptions. (Inherited from <a href="System.Exception">Exception</a>.)</td>
</tr>
<tr>
<td><strong>GetBaseException</strong></td>
<td>Serves as a hash function for a particular type. When overridden in a derived class, sets the <a href="System.Runtime.Serialization.SerializationInfo">SerializationInfo</a> with information about the exception. (Inherited from <a href="System.Exception">Exception</a>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td></td>
</tr>
<tr>
<td><strong>GetObjectData</strong></td>
<td>Gets the runtime type of the current instance. (Inherited from <a href="System.Exception">Exception</a>.) Creates and returns a string representation of the current exception. (Inherited from <a href="System.Exception">Exception</a>.)</td>
</tr>
</tbody>
</table>
See Also

OldVersionException Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `OldVersionException` type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data</strong></td>
<td>Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td><strong>HelpLink</strong></td>
<td>Gets or sets a link to the help file associated with this exception. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td><strong>InnerException</strong></td>
<td>Gets the <code>Exception</code> instance that caused the current exception. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td><strong>Message</strong></td>
<td>Gets a message that describes the current exception. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td><strong>Source</strong></td>
<td>Gets or sets the name of the application or the object that causes the error. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td><strong>StackTrace</strong></td>
<td>Gets a string representation of the frames on the call stack at the time the current exception was thrown. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td><strong>TargetSite</strong></td>
<td>Gets the method that throws the current exception. (Inherited from <code>Exception</code>.)</td>
</tr>
</tbody>
</table>
See Also

OldVersionException Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Namespace: BerkeleyDB
Syntax

**C#**

```csharp
public class PageNotFoundException : DatabaseException
```

**Visual Basic (Declaration)**

```vbnet
Public Class PageNotFoundException
    Inherits DatabaseException
```

**Visual C++**

```cpp
public ref class PageNotFoundException : public DatabaseException
```
Inheritance Hierarchy

System::Object
  System::Exception
    BerkeleyDB::DatabaseException
      BerkeleyDB::PageNotFoundException
See Also

PageNotFoundException Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `PageNotFoundException` type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PageNotFoundException</td>
<td>Initializes a new instance of the PageNotFoundException class</td>
</tr>
</tbody>
</table>
### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a> is equal to the current <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.) When overridden in a derived class, returns the <a href="https://docs.microsoft.com/en-us/dotnet/api/system.exception">Exception</a> that is the root cause of one or more subsequent exceptions. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.exception">Exception</a>.)</td>
</tr>
<tr>
<td><strong>GetBaseException</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.) When overridden in a derived class, sets the <a href="https://docs.microsoft.com/en-us/dotnet/api/system.serialization.serializationinfo">SerializationInfo</a> with information about the exception. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.exception">Exception</a>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Gets the runtime type of the current instance. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.exception">Exception</a>.) Creates and returns a string representation of the current exception. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.exception">Exception</a>.)</td>
</tr>
<tr>
<td><strong>GetObjectData</strong></td>
<td></td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td></td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>ErrorCode</td>
<td>The underlying error code from the Berkeley DB C library. (Inherited from DatabaseException.)</td>
</tr>
</tbody>
</table>
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data</strong></td>
<td>Gets a collection of key/value pairs that provide additional user-defined information about the exception.</td>
</tr>
<tr>
<td><strong>HelpLink</strong></td>
<td>Gets or sets a link to the help file associated with this exception.</td>
</tr>
<tr>
<td><strong>InnerException</strong></td>
<td>Gets the Exception instance that caused the current exception.</td>
</tr>
<tr>
<td><strong>Message</strong></td>
<td>Gets a message that describes the current exception.</td>
</tr>
<tr>
<td><strong>Source</strong></td>
<td>Gets or sets the name of the application or the object that causes the error.</td>
</tr>
<tr>
<td><strong>StackTrace</strong></td>
<td>Gets a string representation of the frames on the call stack at the time the current exception was thrown.</td>
</tr>
<tr>
<td><strong>TargetSite</strong></td>
<td>Gets the method that throws the current exception.</td>
</tr>
</tbody>
</table>
See Also

PageNotFoundException Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
PageNotFoundException Constructor

Initializes a new instance of the PageNotFoundException class

Namespace: BerkeleyDB
Syntax

**C#**

```csharp
public PageNotFoundException()
```

**Visual Basic (Declaration)**

```vbnet
Public Sub New
```

**Visual C++**

```cpp
public:
PageNotFoundException()
```
See Also

PageNotFoundException Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The **PageNotFoundException** type exposes the following members.
<table>
<thead>
<tr>
<th><strong>Name</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ErrorCode</strong></td>
<td>The underlying error code from the Berkeley DB C library. (Inherited from <a href="#">DatabaseException</a>.)</td>
</tr>
</tbody>
</table>
See Also

PageNotFoundException Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `PageNotFoundException` type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a> is equal to the current <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetBaseException</strong></td>
<td>When overridden in a derived class, returns the <a href="https://docs.microsoft.com/en-us/dotnet/api/system.exception">Exception</a> that is the root cause of one or more subsequent exceptions. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.exception">Exception</a>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetObjectData</strong></td>
<td>When overridden in a derived class, sets the <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">SerializationInfo</a> with information about the exception. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.exception">Exception</a>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the runtime type of the current instance. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.exception">Exception</a>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Creates and returns a string representation of the current exception. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.exception">Exception</a>.)</td>
</tr>
</tbody>
</table>
See Also

PageNotFoundException Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `PageNotFoundException` type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data</strong></td>
<td>Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td><strong>HelpLink</strong></td>
<td>Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td><strong>InnerException</strong></td>
<td>Gets the Exception instance that caused the current exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td><strong>Message</strong></td>
<td>Gets a message that describes the current exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td><strong>Source</strong></td>
<td>Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)</td>
</tr>
<tr>
<td><strong>StackTrace</strong></td>
<td>Gets a string representation of the frames on the call stack at the time the current exception was thrown. (Inherited from Exception.)</td>
</tr>
<tr>
<td><strong>TargetSite</strong></td>
<td>Gets the method that throws the current exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>
See Also

PageNotFoundException Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
A class representing a transaction that must be resolved by the application following `Recover(UInt32, Boolean)`.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
**Syntax**

**C#**

```csharp
public class PreparedTransaction
```

**Visual Basic (Declaration)**

```vbnet
Public Class PreparedTransaction
```

**Visual C++**

```cpp
public ref class PreparedTransaction
```
Inheritance Hierarchy

System::Object
BerkeleyDB::PreparedTransaction
See Also

PreparedTransaction Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The PreparedTransaction type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Equals</code></td>
<td>Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><code>GetHashCode</code></td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><code>GetType</code></td>
<td>Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><code>ToString</code></td>
<td>Returns a <a href="#">String</a> that represents the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
</tbody>
</table>
### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GlobalID</td>
<td>The global transaction ID for the transaction. The global transaction ID is the one specified when the transaction was prepared. The application is responsible for ensuring uniqueness among global transaction IDs.</td>
</tr>
<tr>
<td>Txn</td>
<td>The transaction which must be committed, aborted or discarded.</td>
</tr>
</tbody>
</table>
See Also

PreparedTransaction Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The **PreparedTransaction** type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a <a href="#">String</a> that represents the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
</tbody>
</table>
See Also

PreparedTransaction Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `PreparedTransaction` type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GlobalID</td>
<td>The global transaction ID for the transaction. The global transaction ID is the one specified when the transaction was prepared. The application is responsible for ensuring uniqueness among global transaction IDs.</td>
</tr>
<tr>
<td>Txn</td>
<td>The transaction which must be committed, aborted or discarded.</td>
</tr>
</tbody>
</table>
See Also

PreparedTransaction Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
PreparedTransaction...:.GlobalID Property

PreparedTransaction Class  See Also

The global transaction ID for the transaction. The global transaction ID is the one specified when the transaction was prepared. The application is responsible for ensuring uniqueness among global transaction IDs.

Namespace:  BerkeleyDB
**Syntax**

**C#**

```csharp
public byte[] GlobalID { get; }
```

**Visual Basic (Declaration)**

```vbnet
Public Readonly Property GlobalID As Byte()
```

**Visual C++**

```cpp
public:
property array<unsigned char>^ GlobalID {
    array<unsigned char>^ get ();
}
```
See Also

PreparedTransaction Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
PreparedTransaction Class  See Also

The transaction which must be committed, aborted or discarded.

Namespace:  BerkeleyDB
Syntax

C#

public Transaction Txn { get; }

Visual Basic (Declaration)

Public Readonly Property Txn As Transaction

Visual C++

public:
property Transaction^ Txn {
    Transaction^ get ();
}

See Also

PreparedTransaction Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
A class representing a QueueDatabase. The Queue format supports fast access to fixed-length records accessed sequentially or by logical record number.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public class QueueDatabase : Database

Visual Basic (Declaration)

Public Class QueueDatabase _
    Inherits Database

Visual C++

public ref class QueueDatabase : public Database
Inheritance Hierarchy

System:::Object
BerkeleyDB:::BaseDatabase
BerkeleyDB:::Database
BerkeleyDB:::QueueDatabase
See Also

QueueDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The **QueueDatabase** type exposes the following members.
### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Append</td>
<td>Overloaded. Overloaded.</td>
</tr>
</tbody>
</table>

Although closing a database will close any open cursors, it is recommended that applications explicitly close all their Cursor objects before closing the database. The reason why is that when the cursor is explicitly closed, the memory allocated for it is reclaimed; however, this will not happen if you close a database while cursors are still opened.

The same rule, for the same reasons, hold true for Transaction objects. Simply make sure you resolve all your transaction objects before closing your database handle.

Because key/data pairs are cached in memory, applications should make a point to always either close database handles or sync their data to disk (using Sync() before exiting, to ensure that any data cached in main memory are reflected in the underlying file system.

When called on a database that is the primary database for a secondary index, the primary database should be closed only after all secondary indices referencing it have been closed.

When multiple threads are using the object concurrently, only a single thread may call the Close method.

The object may not be accessed again after Close is called, regardless of its outcome.
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Consume</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>Cursor</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>Delete</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>Dispose</td>
<td>Release the resources held by this object, and close the database if it's still open. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified Object is equal to the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Exists</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>FastStats</td>
<td>Among other things, this method makes it possible for applications to request key and record counts without incurring the performance penalty of traversing the entire database. The statistical information is described by the BTreeStats, HashStats, QueueStats, and RecnoStats classes.</td>
</tr>
<tr>
<td>Get</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>GetBoth</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>GetBothMultiple</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as a hash function for a particular type. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetMultiple</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>Join</td>
<td>Create a specialized join cursor for use in performing equality or natural joins on secondary indices. (Inherited from Database.)</td>
</tr>
<tr>
<td>Open</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>PrintFastStats</td>
<td>The statistical information is described by the</td>
</tr>
</tbody>
</table>
The statistical information is described by the **BTreeStats**, **HashStats**, **QueueStats**, and **RecnoStats** classes.

Overloaded.

**PrintStats**

The statistical information is described by the **BTreeStats**, **HashStats**, **QueueStats**, and **RecnoStats** classes.

Overloaded.

**Put**

If the database supports duplicates, add the new data value at the end of the duplicate set. If the database supports sorted duplicates, the new data value is inserted at the correct sorted location.

**PutNoOverwrite**

Overloaded.

**Stats**

The statistical information is described by **BTreeStats**.

Overloaded.

**Sync**

Flush any cached information to disk.

(Inherited from **BaseDatabase**.)

**ToString**

Returns a **String** that represents the current **Object**.

(Inherited from **Object**.)

Overloaded.

When called on a database configured with secondary indices, Truncate will truncate the primary database and all secondary indices. A count of the records discarded from the primary database is returned.
## Properties

<table>
<thead>
<tr>
<th><strong>Name</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>AutoCommit</td>
<td>If true, all database modification operations based on this object will be</td>
</tr>
<tr>
<td></td>
<td>transactionally protected. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>CacheSize</td>
<td>The size of the shared memory buffer pool -- that is, the cache.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>Creation</td>
<td>The CreatePolicy with which this database was opened.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>DatabaseName</td>
<td>The name of this database, if it has one.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>DoChecksum</td>
<td>If true, do checksum verification of pages read into the cache from the</td>
</tr>
<tr>
<td></td>
<td>backing filestore.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>EncryptAlgorithm</td>
<td>The algorithm used by the Berkeley DB library to perform encryption and</td>
</tr>
<tr>
<td></td>
<td>decryption.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>Encrypted</td>
<td>If true, encrypt all data stored in the database.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>Endianness</td>
<td>The database byte order.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>ErrorFeedback</td>
<td>The mechanism for reporting detailed error messages to the application.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>ErrorPrefix</td>
<td>The prefix string that appears before error messages issued by Berkeley DB.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>ExtentSize</td>
<td>The size of the extents used to hold pages in a QueueDatabase, specified as</td>
</tr>
<tr>
<td></td>
<td>a number of pages.</td>
</tr>
<tr>
<td>Feedback</td>
<td>Monitor progress within long running operations.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>FileName</strong></td>
<td>The filename of this database, if it has one. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>FreeThreaded</strong></td>
<td>If true, the object is free-threaded; that is, concurrently usable by multiple threads in the address space. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>HasMultiple</strong></td>
<td>If true, the object references a physical file supporting multiple databases. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>InHostOrder</strong></td>
<td>If true, the underlying database files were created on an architecture of the same byte order as the current one. This information may be used to determine whether application data needs to be adjusted for this architecture or not. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>InOrder</strong></td>
<td>If true, modify the operation of Consume(Boolean) to return key/data pairs in order. That is, they will always return the key/data item from the head of the queue.</td>
</tr>
<tr>
<td><strong>Length</strong></td>
<td>The length of records in the database.</td>
</tr>
<tr>
<td><strong>NoMMap</strong></td>
<td>If true, this database is not mapped into process memory. See MMapSize for further information. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>NonDurableTxns</strong></td>
<td>If true, Berkeley DB will not write log records for this database. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>PadByte</strong></td>
<td>The padding character for short, fixed-length records.</td>
</tr>
<tr>
<td><strong>Pagesize</strong></td>
<td>The database's current page size. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>Priority</strong></td>
<td>The cache priority for pages referenced by this object. (Inherited from BaseDatabase.)</td>
</tr>
</tbody>
</table>
ReadOnly
If true, this database has been opened for reading only. Any attempt to modify items in the database will fail, regardless of the actual permissions of any underlying files.
(Inherited from BaseDatabase.)

ReadUncommitted
If true, this database supports transactional read operations with degree 1 isolation. Read operations on the database may request the return of modified but not yet committed data.
(Inherited from BaseDatabase.)

Transactional
If true, this database has been opened in a transactional mode.
(Inherited from BaseDatabase.)

Truncated
If true, the underlying file was physically truncated upon open, discarding all previous databases it might have held.
(Inherited from BaseDatabase.)

Type
The type of the underlying access method (and file format). This value may be used to determine the type of the database after an Open(String, DatabaseConfig).
(Inherited from BaseDatabase.)

UseMVCC
If true, the database was opened with support for multiversion concurrency control.
(Inherited from BaseDatabase.)
See Also

QueueDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `QueueDatabase` type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Append</td>
<td>Overloaded. Overloaded.</td>
</tr>
</tbody>
</table>

Although closing a database will close any open cursors, it is recommended that applications explicitly close all their Cursor objects before closing the database. The reason why is that when the cursor is explicitly closed, the memory allocated for it is reclaimed; however, this will not happen if you close a database while cursors are still opened.

The same rule, for the same reasons, hold true for Transaction objects. Simply make sure you resolve all your transaction objects before closing your database handle.

Because key/data pairs are cached in memory, applications should make a point to always either close database handles or sync their data to disk (using `Sync()` before exiting, to ensure that any data cached in main memory are reflected in the underlying file system.

When called on a database that is the primary database for a secondary index, the primary database should be closed only after all secondary indices referencing it have been closed.

When multiple threads are using the object concurrently, only a single thread may call the Close method.

The object may not be accessed again after Close is called, regardless of its outcome.
- **Consume** Overloaded.
- **Cursor** Overloaded.
- **Delete** Overloaded.
- **Dispose**
  Release the resources held by this object, and close the database if it's still open.
  (Inherited from `BaseDatabase`.)
- **Equals**
  Determines whether the specified `Object` is equal to the current `Object`.
  (Inherited from `Object`.)
- **Exists** Overloaded.
- **FastStats**
  Among other things, this method makes it possible for applications to request key and record counts without incurring the performance penalty of traversing the entire database.
  The statistical information is described by the `BTreeStats`, `HashStats`, `QueueStats`, and `RecnoStats` classes.
- **Get** Overloaded.
- **GetBoth** Overloaded.
- **GetBothMultiple** Overloaded.
- **GetHashCode**
  Serves as a hash function for a particular type.
  (Inherited from `Object`.)
- **GetMultiple** Overloaded.
- **GetType**
  Gets the `Type` of the current instance.
  (Inherited from `Object`.)
- **Join**
  Create a specialized join cursor for use in performing equality or natural joins on secondary indices.
  (Inherited from `Database`.)
- **Open** Overloaded.
- **PrintFastStats**
  The statistical information is described by the
**BTreeStats, HashStats, QueueStats, and RecnoStats** classes.
Overloaded.

The statistical information is described by the
**BTreeStats, HashStats, QueueStats, and RecnoStats** classes.
Overloaded.

If the database supports duplicates, add the new data
value at the end of the duplicate set. If the database
supports sorted duplicates, the new data value is
inserted at the correct sorted location.

**PrintStats**

Overloaded.

**Put**

Overloaded.

The statistical information is described by **BTreeStats**.

**PutNoOverwrite**

Overloaded.

**Stats**

Flush any cached information to disk.
(Inherited from **BaseDatabase**.)

**Sync**

Returns a **String** that represents the current **Object**.
(Inherited from **Object**.)

**ToString**

Overloaded.

When called on a database configured with secondary
indices, Truncate will truncate the primary database
and all secondary indices. A count of the records
discarded from the primary database is returned.

**Truncate**
See Also

QueueDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
QueueDatabase Class  See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Append(DatabaseEntry)</code></td>
<td>Append the data item to the end of the database.</td>
</tr>
<tr>
<td><code>Append(DatabaseEntry, Transaction)</code></td>
<td>Append the data item to the end of the database.</td>
</tr>
</tbody>
</table>
See Also

QueueDatabase Class
QueueDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Append the data item to the end of the database.

Namespace: BerkeleyDB
Syntax

C#

public uint Append(
    DatabaseEntry data
)

Visual Basic (Declaration)

Public Function Append ( _
    data As DatabaseEntry _
) AsUInteger

Visual C++

public:
    unsigned int Append(
        DatabaseEntry^ data
    )

Parameters

data
    Type: BerkeleyDB::DatabaseEntry
    The data item to store in the database

Return Value

The record number allocated to the record
See Also

QueueDatabase Class
Append Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Append the data item to the end of the database.

**Namespace:**  [BerkeleyDB](mailto:BerkeleyDB)

**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

```csharp
public uint Append(
    DatabaseEntry data,
    Transaction txn
)
```

Visual Basic (Declaration)

```vbnet
Public Function Append ( _
    data As DatabaseEntry, _
    txn As Transaction _
) AsUInteger
```

Visual C++

```cpp
public:
    unsigned int Append(
        DatabaseEntry^ data,
        Transaction^ txn
    )
```

Parameters

data
Type: BerkeleyDB::::DatabaseEntry
The data item to store in the database

taxn
Type: BerkeleyDB::::Transaction
If the operation is part of an application-specified transaction, txn is a Transaction object returned from BeginTransaction(); if the operation is part of a Berkeley DB Concurrent Data Store group, txn is a handle returned from BeginCDSGroup(); otherwise null.

Return Value
The record number allocated to the record
Remarks

There is a minor behavioral difference between Append(DatabaseEntry) and Append(DatabaseEntry). If a transaction enclosing an Append operation aborts, the record number may be reallocated in a subsequent Append(DatabaseEntry) operation, but it will not be reallocated in a subsequent Append(DatabaseEntry) operation.
See Also

QueueDatabase Class
Append Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Although closing a database will close any open cursors, it is recommended that applications explicitly close all their Cursor objects before closing the database. The reason why is that when the cursor is explicitly closed, the memory allocated for it is reclaimed; however, this will not happen if you close a database while cursors are still opened.

The same rule, for the same reasons, hold true for Transaction objects. Simply make sure you resolve all your transaction objects before closing your database handle.

Because key/data pairs are cached in memory, applications should make a point to always either close database handles or sync their data to disk (using Sync() before exiting, to ensure that any data cached in main memory are reflected in the underlying file system.

When called on a database that is the primary database for a secondary index, the primary database should be closed only after all secondary indices referencing it have been closed.

When multiple threads are using the object concurrently, only a single thread may call the Close method.

The object may not be accessed again after Close is called, regardless of its outcome.
### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Close()()</td>
<td>Flush any cached database information to disk, close any open Cursor()() objects, free any allocated resources, and close any underlying files. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>Close(Boolean)</td>
<td>Optionally flush any cached database information to disk, close any open Cursor()() objects, free any allocated resources, and close any underlying files. (Inherited from BaseDatabase.)</td>
</tr>
</tbody>
</table>
See Also

QueueDatabase Class
QueueDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
QueueDatabase..::.Consume Method

QueueDatabase Class  See Also
### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Consume(Boolean)</td>
<td>Return the record number and data from the available record closest to the head of the queue, and delete the record.</td>
</tr>
<tr>
<td>Consume(Boolean, Transaction)</td>
<td>Return the record number and data from the available record closest to the head of the queue, and delete the record.</td>
</tr>
<tr>
<td>Consume(Boolean, Transaction, LockingInfo)</td>
<td>Return the record number and data from the available record closest to the head of the queue, and delete the record.</td>
</tr>
</tbody>
</table>
See Also

QueueDatabase Class
QueueDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
QueueDatabase class

Consume Method (Boolean)

QueueDatabase..::.Consume Method (Boolean)

QueueDatabase Class  See Also

Return the record number and data from the available record closest to the head of the queue, and delete the record.

Namespace:  BerkeleyDB
### Syntax

#### C#

```csharp
public KeyValuePair<uint, DatabaseEntry> Consume(
    bool wait
)
```

#### Visual Basic (Declaration)

```vbnet
Public Function Consume (_
    wait As Boolean _
) As KeyValuePair(Of UInteger, DatabaseEntry)
```

#### Visual C++

```cpp
public:
    KeyValuePair<unsigned int, DatabaseEntry^> Consume(
    bool wait
)
```

### Parameters

**wait**

Type: `System::::Boolean`

If true and the Queue database is empty, the thread of control will wait until there is data in the queue before returning.

### Return Value

A `KeyValuePair(Of (Of TKey, TValue)>)` whose Key parameter is the record number and whose Value parameter is the retrieved data.
## Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>BerkeleyDB.:::LockNotGrantedException</td>
<td>If lock or transaction timeouts have been specified, a <strong>LockNotGrantedException</strong> may be thrown. This failure, by itself, does not require the enclosing transaction be aborted.</td>
</tr>
</tbody>
</table>
See Also

QueueDatabase Class
Consume Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
QueueDatabase::<Consume Method (Boolean, Transaction)

QueueDatabase Class  See Also

Return the record number and data from the available record closest to the head of the queue, and delete the record.

Namespace:  BerkeleyDB
### Syntax

#### C#

```csharp
public KeyValuePair<uint, DatabaseEntry> Consume(
    bool wait,
    Transaction txn
)
```

#### Visual Basic (Declaration)

```vbnet
Public Function Consume (_
    wait As Boolean, _
    txn As Transaction _
) As KeyValuePair(Of UInteger, DatabaseEntry)
```

#### Visual C++

```cpp
public:
    KeyValuePair<unsigned int, DatabaseEntry> Consume(
        bool wait,
        Transaction^ txn
    )
```

### Parameters

**wait**
- Type: `System::::Boolean`
- If true and the Queue database is empty, the thread of control will wait until there is data in the queue before returning.

**txn**
- Type: `BerkeleyDB::::Transaction`
  - txn is a Transaction object returned from `BeginTransaction()();` if the operation is part of a Berkeley DB Concurrent Data Store group, txn is a handle returned from `BeginCDSGroup()();` otherwise null.

### Return Value
A `KeyValuePair<Of (TKey, TValue)>` whose Key parameter is the record number and whose Value parameter is the retrieved data.
<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>BerkeleyDB...:::LockNotGrantedException</td>
<td>If lock or transaction timeouts have been specified, a LockNotGrantedException may be thrown. This failure, by itself, does not require the enclosing transaction be aborted.</td>
</tr>
</tbody>
</table>
See Also

QueueDatabase Class  
Consume Overload  
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
QueueDatabase Class  See Also

Return the record number and data from the available record closest to the head of the queue, and delete the record.

Namespace:  BerkeleyDB
Syntax

C#

public KeyValuePair<uint, DatabaseEntry> Consume(
    bool wait,
    Transaction txn,
    LockingInfo info
)

Visual Basic (Declaration)

Public Function Consume (_
    wait As Boolean, _
    txn As Transaction, _
    info As LockingInfo _
) As KeyValuePair(Of UInteger, DatabaseEntry)

Visual C++

public:
    KeyValuePair<unsigned int, DatabaseEntry> Consume(
        bool wait,
        Transaction^ txn,
        LockingInfo^ info
    )

Parameters

wait
Type: System::Boolean
If true and the Queue database is empty, the thread of control will wait until there is data in the queue before returning.

txn
Type: BerkeleyDB::Transaction
txn is a Transaction object returned from BeginTransaction(); if the operation is part of a Berkeley DB Concurrent Data Store group, txn is a handle returned from BeginCDSGroup(); otherwise null.
Info

Type: BerkeleyDB::::LockingInfo
The locking behavior to use.

Return Value

A KeyValuePair(Of (TKey, TValue)> whose Key parameter is the record number and whose Value parameter is the retrieved data.
## Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BerkeleyDB::::LockNotGrantedException</strong></td>
<td>If lock or transaction timeouts have been specified, a <strong>LockNotGrantedException</strong> may be thrown. This failure, by itself, does not require the enclosing transaction be aborted.</td>
</tr>
</tbody>
</table>
See Also

QueueDatabase Class
Consume Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
QueueDatabase Class  See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Cursor()</code></td>
<td>Create a database cursor.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
<tr>
<td><code>Cursor(CursorConfig)</code></td>
<td>Create a database cursor with the given configuration.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
<tr>
<td><code>Cursor(Transaction)</code></td>
<td>Create a transactionally protected database cursor.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
<tr>
<td><code>Cursor(CursorConfig, Transaction)</code></td>
<td>Create a transactionally protected database cursor with the given configuration.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
</tbody>
</table>
See Also

QueueDatabase Class
QueueDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
QueueDatabase Class  See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delete(DatabaseEntry)</td>
<td>Remove key/data pairs from the database. The key/data pair associated with key is discarded from the database. In the presence of duplicate key values, all records associated with the designated key will be discarded. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>Delete(DatabaseEntry, Transaction)</td>
<td>Remove key/data pairs from the database. The key/data pair associated with key is discarded from the database. In the presence of duplicate key values, all records associated with the designated key will be discarded. (Inherited from BaseDatabase.)</td>
</tr>
</tbody>
</table>
See Also

QueueDatabase Class
QueueDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
QueueDatabase::Exists Method
QueueDatabase Class  See Also
### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exists(DatabaseEntry)</td>
<td>Check whether key appears in the database. (Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
<tr>
<td>Exists(DatabaseEntry, Transaction)</td>
<td>Check whether key appears in the database. (Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
<tr>
<td>Exists(DatabaseEntry, Transaction, LockingInfo)</td>
<td>Check whether key appears in the database. (Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
</tbody>
</table>
See Also

QueueDatabase Class
QueueDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Among other things, this method makes it possible for applications to request key and record counts without incurring the performance penalty of traversing the entire database.

The statistical information is described by the BTreeStats, HashStats, QueueStats, and RecnoStats classes.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FastStats()()()</td>
<td>Return the database statistical information which does not require traversal of the database.</td>
</tr>
<tr>
<td>FastStats(Transaction)</td>
<td>Return the database statistical information which does not require traversal of the database.</td>
</tr>
<tr>
<td>FastStats(Transaction, Isolation)</td>
<td>Return the database statistical information which does not require traversal of the database.</td>
</tr>
</tbody>
</table>
See Also

QueueDatabase Class
QueueDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
QueueDatabase Class  See Also

Return the database statistical information which does not require traversal of the database.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public QueueStats FastStats()

Visual Basic (Declaration)

Public Function FastStats As QueueStats

Visual C++

public: QueueStats^ FastStats()

Return Value

The database statistical information which does not require traversal of the database.
See Also

QueueDatabase Class
FastStats Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Return the database statistical information which does not require traversal of the database.

**Namespace:**  BerkeleyDB  
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll)  Version: 4.8.24.0
Syntax

C#

```csharp
public QueueStats FastStats(
    Transaction txn
)
```

Visual Basic (Declaration)

```vbnet
Public Function FastStats ( _
    txn As Transaction _
) As QueueStats
```

Visual C++

```cpp
public:
    QueueStats^ FastStats(
        Transaction^ txn
    )
```

Parameters

txn

Type: BerkeleyDB::::Transaction
If the operation is part of an application-specified transaction, txn is a Transaction object returned from BeginTransaction(); if the operation is part of a Berkeley DB Concurrent Data Store group, txn is a handle returned from BeginCDSGroup(); otherwise null.

Return Value

The database statistical information which does not require traversal of the database.
See Also

QueueDatabase Class
FastStats Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
QueueDatabase..::.FastStats Method (Transaction, Isolation)

QueueDatabase Class  See Also

Return the database statistical information which does not require traversal of the database.

Namespace:  BerkeleyDB
### Syntax

#### C#

```csharp
public QueueStats FastStats(
    Transaction txn,
    Isolation isoDegree
)
```

#### Visual Basic (Declaration)

```vbnet
Public Function FastStats (_
    txn As Transaction, _
    isoDegree As Isolation _
) As QueueStats
```

#### Visual C++

```cpp
public:
    QueueStats^ FastStats(
        Transaction^ txn,
        Isolation isoDegree
    )
```

### Parameters

txn
Type: `BerkeleyDB::::Transaction`
If the operation is part of an application-specified transaction, txn is a Transaction object returned from `BeginTransaction();` if the operation is part of a Berkeley DB Concurrent Data Store group, txn is a handle returned from `BeginCDSGroup();` otherwise null.

isoDegree
Type: `BerkeleyDB::::Isolation`
The level of isolation for database reads. `DEGREE_ONE` will be silently ignored for databases which did not specify `ReadUncommitted`.
**Return Value**

The database statistical information which does not require traversal of the database.
See Also

QueueDatabase Class
FastStats Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
QueueDatabase Class  See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Get(DatabaseEntry)</code></td>
<td>Retrieve a key/data pair from the database. In the presence of duplicate key values, Get will return the first data item for key.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">BaseDatabase</a>).</td>
</tr>
<tr>
<td>`Get(DatabaseEntry,</td>
<td>Retrieve a key/data pair from the database. In the presence of duplicate key values, Get will return the first data item for key.</td>
</tr>
<tr>
<td>Transaction)`</td>
<td>(Inherited from <a href="#">BaseDatabase</a>).</td>
</tr>
<tr>
<td>`Get(DatabaseEntry,</td>
<td>Retrieve a key/data pair from the database. In the presence of duplicate key values, Get will return the first data item for key.</td>
</tr>
<tr>
<td>Transaction,</td>
<td>(Inherited from <a href="#">BaseDatabase</a>).</td>
</tr>
<tr>
<td>LockingInfo)`</td>
<td>(Inherited from <a href="#">BaseDatabase</a>).</td>
</tr>
</tbody>
</table>
See Also

QueueDatabase Class
QueueDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
QueueDatabase Class  See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetBoth(DatabaseEntry, DatabaseEntry)</td>
<td>Retrieve a key/data pair from the database which matches key and data. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>GetBoth(DatabaseEntry, DatabaseEntry, Transaction)</td>
<td>Retrieve a key/data pair from the database which matches key and data. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>GetBoth(DatabaseEntry, DatabaseEntry, Transaction, LockingInfo)</td>
<td>Retrieve a key/data pair from the database which matches key and data. (Inherited from BaseDatabase.)</td>
</tr>
</tbody>
</table>
See Also

QueueDatabase Class
QueueDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
QueueDatabase Class  See Also
### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>GetBothMultiple(DatabaseEntry, DatabaseEntry)</code></td>
<td>If a key/data pair in the database matches key and data, return the key and all duplicate data items. (Inherited from <code>Database</code>.)</td>
</tr>
<tr>
<td><code>GetBothMultiple(DatabaseEntry, DatabaseEntry, Int32)</code></td>
<td>If a key/data pair in the database matches key and data, return the key and all duplicate data items. (Inherited from <code>Database</code>.)</td>
</tr>
<tr>
<td><code>GetBothMultiple(DatabaseEntry, DatabaseEntry, Int32, Transaction)</code></td>
<td>If a key/data pair in the database matches key and data, return the key and all duplicate data items. (Inherited from <code>Database</code>.)</td>
</tr>
<tr>
<td><code>GetBothMultiple(DatabaseEntry, DatabaseEntry, Int32, Transaction, LockingInfo)</code></td>
<td>If a key/data pair in the database matches key and data, return the key and all duplicate data items. (Inherited from <code>Database</code>.)</td>
</tr>
</tbody>
</table>
See Also

QueueDatabase Class
QueueDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
QueueDatabase Class See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetMultiple(DatabaseEntry)</td>
<td>Retrieve a key and all duplicate data items from the database. (Inherited from <a href="#">Database</a>.)</td>
</tr>
<tr>
<td>GetMultiple(DatabaseEntry, Int32)</td>
<td>Retrieve a key and all duplicate data items from the database. (Inherited from <a href="#">Database</a>.)</td>
</tr>
<tr>
<td>GetMultiple(DatabaseEntry, Int32, Transaction)</td>
<td>Retrieve a key and all duplicate data items from the database. (Inherited from <a href="#">Database</a>.)</td>
</tr>
<tr>
<td>GetMultiple(DatabaseEntry, Int32, Transaction, LockingInfo)</td>
<td>Retrieve a key and all duplicate data items from the database. (Inherited from <a href="#">Database</a>.)</td>
</tr>
</tbody>
</table>
See Also

QueueDatabase Class
QueueDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
QueueDatabase Class  See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Open(String, QueueDatabaseConfig)" /></td>
<td>Instantiate a new QueueDatabase object and open the database represented by Filename.</td>
</tr>
<tr>
<td><img src="image" alt="Open(String, QueueDatabaseConfig, Transaction)" /></td>
<td>Instantiate a new QueueDatabase object and open the database represented by Filename.</td>
</tr>
</tbody>
</table>
See Also

QueueDatabase Class
QueueDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Instantiate a new QueueDatabase object and open the database represented by Filename.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

**C#**

```csharp
public static QueueDatabase Open(
    string Filename,
    QueueDatabaseConfig cfg
)
```

**Visual Basic (Declaration)**

```vbnet
Public Shared Function Open ( _
    Filename As String, _
    cfg As QueueDatabaseConfig _
) As QueueDatabase
```

**Visual C++**

```cpp
public:
static QueueDatabase^ Open(
    String^ Filename,
    QueueDatabaseConfig^ cfg
)
```

**Parameters**

**Filename**

Type: `System::String`

The name of an underlying file that will be used to back the database. In-memory databases never intended to be preserved on disk may be created by setting this parameter to null.

**cfg**

Type: `BerkeleyDB::QueueDatabaseConfig`

The database's configuration

**Return Value**
A new, open database object
Remarks

If Filename is null, the database is strictly temporary and cannot be opened by any other thread of control, thus the database can only be accessed by sharing the single database object that created it, in circumstances where doing so is safe.

If AutoCommit is set, the operation will be implicitly transaction protected. Note that transactionally protected operations on a database object requires the object itself be transactionally protected during its open.
See Also

QueueDatabase Class
Open Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
QueueDatabase Class  

Instantiate a new QueueDatabase object and open the database represented by Filename.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

```csharp
public static QueueDatabase Open(
    string Filename,
    QueueDatabaseConfig cfg,
    Transaction txn
)
```

Visual Basic (Declaration)

```vbnet
Public Shared Function Open (_
    Filename As String,
    cfg As QueueDatabaseConfig, _
    txn As Transaction _
) As QueueDatabase
```

Visual C++

```cpp
public:
static QueueDatabase^ Open(
    String^ Filename, 
    QueueDatabaseConfig^ cfg, 
    Transaction^ txn
)
```

Parameters

Filename

Type: `System::String`

The name of an underlying file that will be used to back the database. In-memory databases never intended to be preserved on disk may be created by setting this parameter to null.

cfg

Type: `BerkeleyDB::QueueDatabaseConfig`

The database’s configuration
txn

Type: BerkeleyDB::::Transaction
If the operation is part of an application-specified transaction, txn is a Transaction object returned from BeginTransaction(); if the operation is part of a Berkeley DB Concurrent Data Store group, txn is a handle returned from BeginCDSGroup(); otherwise null.

Return Value

A new, open database object
Remarks

If Filename is null, the database is strictly temporary and cannot be opened by any other thread of control, thus the database can only be accessed by sharing the single database object that created it, in circumstances where doing so is safe.

If txn is null, but AutoCommit is set, the operation will be implicitly transaction protected. Note that transactionally protected operations on a database object requires the object itself be transactionally protected during its open. Also note that the transaction must be committed before the object is closed.
See Also

QueueDatabase Class
Open Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
QueueDatabase Class  See Also

The statistical information is described by the BTreeStats, HashStats, QueueStats, and RecnoStats classes.
### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>PrintFastStats()()()</code></td>
<td>Display the database statistical information which does not require traversal of the database. (Inherited from <code>BaseDatabase</code>.)</td>
</tr>
<tr>
<td><code>PrintFastStats(Boolean)</code></td>
<td>Display the database statistical information which does not require traversal of the database. (Inherited from <code>BaseDatabase</code>.)</td>
</tr>
</tbody>
</table>
See Also

QueueDatabase Class
QueueDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
QueueDatabase Class  See Also

The statistical information is described by the BTreeStats, HashStats, QueueStats, and RecnoStats classes.
# Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PrintStats()</td>
<td>Display the database statistical information. (Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
<tr>
<td>PrintStats(Boolean)</td>
<td>Display the database statistical information. (Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
</tbody>
</table>
See Also

QueueDatabase Class
QueueDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If the database supports duplicates, add the new data value at the end of the duplicate set. If the database supports sorted duplicates, the new data value is inserted at the correct sorted location.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Put(DatabaseEntry, DatabaseEntry)</td>
<td>Store the key/data pair in the database, replacing any previously existing key if duplicates are disallowed, or adding a duplicate data item if duplicates are allowed. (Inherited from Database.)</td>
</tr>
<tr>
<td>Put(DatabaseEntry, DatabaseEntry, Transaction)</td>
<td>Store the key/data pair in the database, replacing any previously existing key if duplicates are disallowed, or adding a duplicate data item if duplicates are allowed. (Inherited from Database.)</td>
</tr>
</tbody>
</table>
See Also

QueueDatabase Class
QueueDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
QueueDatabase Class  See Also
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>PutNoOverwrite(DatabaseEntry, DatabaseEntry)</code></td>
<td>Store the key/data pair in the database, only if the key does not already appear in the database. (Inherited from <a href="#">Database</a>.)</td>
</tr>
<tr>
<td><code>PutNoOverwrite(DatabaseEntry, DatabaseEntry, Transaction)</code></td>
<td>Store the key/data pair in the database, only if the key does not already appear in the database. (Inherited from <a href="#">Database</a>.)</td>
</tr>
</tbody>
</table>
See Also

QueueDatabase Class
QueueDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The statistical information is described by BTreeStats.
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stats()()</td>
<td>Return the database statistical information for this database.</td>
</tr>
<tr>
<td>Stats(Transaction)</td>
<td>Return the database statistical information for this database.</td>
</tr>
<tr>
<td>Stats(Transaction, Isolation)</td>
<td>Return the database statistical information for this database.</td>
</tr>
</tbody>
</table>
See Also

QueueDatabase Class
QueueDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
QueueDatabase Class

See Also

Return the database statistical information for this database.

Namespace: BerkeleyDB
Syntax

C#

public QueueStats Stats()

Visual Basic (Declaration)

Public Function Stats As QueueStats

Visual C++

public:
QueueStats^ Stats()

Return Value

Database statistical information.
See Also

QueueDatabase Class
Stats Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Return the database statistical information for this database.

**Namespace:** [BerkeleyDB](https://en.wikipedia.org/wiki/Berkeley_DB)  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll)  
**Version:** 4.8.24.0
Syntax

C#

public QueueStats Stats(
   Transaction txn
)

Visual Basic (Declaration)

Public Function Stats (_
   txn As Transaction _
) As QueueStats

Visual C++

public:
   QueueStats^ Stats(
      Transaction^ txn
   )

Parameters

txn

Type: BerkeleyDB::Transaction
If the operation is part of an application-specified transaction, txn is a Transaction object returned from BeginTransaction(); if the operation is part of a Berkeley DB Concurrent Data Store group, txn is a handle returned from BeginCDSGroup(); otherwise null.

Return Value

Database statistical information.
See Also

QueueDatabase Class
Stats Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
QueueDatabase...::Stats Method (Transaction, Isolation)

QueueDatabase Class  See Also

Return the database statistical information for this database.

Namespace:  BerkeleyDB
Syntax

C#

public QueueStats Stats(
    Transaction txn,
    Isolation isoDegree
)

Visual Basic (Declaration)

Public Function Stats ( _
    txn As Transaction, _
    isoDegree As Isolation _
) As QueueStats

Visual C++

public:
QueueStats^ Stats(  
    Transaction^ txn,  
    Isolation isoDegree
)

Parameters

txn
Type: BerkeleyDB::::Transaction
If the operation is part of an application-specified transaction, txn is a Transaction object returned from BeginTransaction(); if the operation is part of a Berkeley DB Concurrent Data Store group, txn is a handle returned from BeginCDSGroup(); otherwise null.

isoDegree
Type: BerkeleyDB::::Isolation
The level of isolation for database reads. DEGREE_ONE will be silently ignored for databases which did not specify ReadUncommitted.
Return Value

Database statistical information.
See Also

QueueDatabase Class
Stats Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
QueueDatabase Class  See Also

When called on a database configured with secondary indices, Truncate will truncate the primary database and all secondary indices. A count of the records discarded from the primary database is returned.
### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Truncate()</strong></td>
<td>Empty the database, discarding all records it contains.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
<tr>
<td><strong>Truncate(Transaction)</strong></td>
<td>Empty the database, discarding all records it contains.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
</tbody>
</table>
See Also

QueueDatabase Class
QueueDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The QueueDatabase type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AutoCommit</td>
<td>If true, all database modification operations based on this object will be transactionally protected. (Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
<tr>
<td>CacheSize</td>
<td>The size of the shared memory buffer pool -- that is, the cache. (Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
<tr>
<td>Creation</td>
<td>The CreatePolicy with which this database was opened. (Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
<tr>
<td>DatabaseName</td>
<td>The name of this database, if it has one. (Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
<tr>
<td>DoChecksum</td>
<td>If true, do checksum verification of pages read into the cache from the backing filestore. (Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
<tr>
<td>EncryptAlgorithm</td>
<td>The algorithm used by the Berkeley DB library to perform encryption and decryption. (Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
<tr>
<td>Encrypted</td>
<td>If true, encrypt all data stored in the database. (Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
<tr>
<td>Endianness</td>
<td>The database byte order. (Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
<tr>
<td>ErrorFeedback</td>
<td>The mechanism for reporting detailed error messages to the application. (Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
<tr>
<td>ErrorPrefix</td>
<td>The prefix string that appears before error messages issued by Berkeley DB. (Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
<tr>
<td>ExtentSize</td>
<td>The size of the extents used to hold pages in a <a href="#">QueueDatabase</a>, specified as a number of pages.</td>
</tr>
<tr>
<td>Feedback</td>
<td>Monitor progress within long running operations. (Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>FileName</td>
<td>The filename of this database, if it has one.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>FreeThreaded</td>
<td>If true, the object is free-threaded; that is, concurrently usable by multiple threads in the address space.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>HasMultiple</td>
<td>If true, the object references a physical file supporting multiple databases.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>InHostOrder</td>
<td>If true, the underlying database files were created on an architecture of the same byte order as the current one. This information may be used to determine whether application data needs to be adjusted for this architecture or not.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>InOrder</td>
<td>If true, modify the operation of Consume(Boolean) to return key/data pairs in order. That is, they will always return the key/data item from the head of the queue.</td>
</tr>
<tr>
<td>Length</td>
<td>The length of records in the database.</td>
</tr>
<tr>
<td></td>
<td>If true, this database is not mapped into process memory.</td>
</tr>
<tr>
<td>NoMMap</td>
<td>See MMapSize for further information.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>NonDurableTxns</td>
<td>If true, Berkeley DB will not write log records for this database.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>PadByte</td>
<td>The padding character for short, fixed-length records.</td>
</tr>
<tr>
<td>Pagesize</td>
<td>The database's current page size.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>Priority</td>
<td>The cache priority for pages referenced by this object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>ReadOnly</strong></td>
<td>If true, this database has been opened for reading only. Any attempt to modify items in the database will fail, regardless of the actual permissions of any underlying files. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>ReadUncommitted</strong></td>
<td>If true, this database supports transactional read operations with degree 1 isolation. Read operations on the database may request the return of modified but not yet committed data. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>Transactional</strong></td>
<td>If true, this database has been opened in a transactional mode. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>Truncated</strong></td>
<td>If true, the underlying file was physically truncated upon open, discarding all previous databases it might have held. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>Type</strong></td>
<td>The type of the underlying access method (and file format). This value may be used to determine the type of the database after an Open(String, DatabaseConfig). (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>UseMVCC</strong></td>
<td>If true, the database was opened with support for multiversion concurrency control. (Inherited from BaseDatabase.)</td>
</tr>
</tbody>
</table>
See Also

QueueDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The size of the extents used to hold pages in a QueueDatabase, specified as a number of pages.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public uint ExtentSize { get; }

Visual Basic (Declaration)

Public ReadOnly Property ExtentSize AsUInteger

Visual C++

public:
property unsigned int ExtentSize {
    unsigned int get ();
}

See Also

QueueDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, modify the operation of `Consume(Boolean)` to return key/data pairs in order. That is, they will always return the key/data item from the head of the queue.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
**Syntax**

**C#**

```csharp
public bool InOrder { get; }
```

**Visual Basic (Declaration)**

```vbnet
Public ReadOnly Property InOrder As Boolean
```

**Visual C++**

```cpp
public:
property bool InOrder {
    bool get ();
}
```
See Also

QueueDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
QueueDatabase Class  See Also

The length of records in the database.

Namespace:  BerkeleyDB  
Syntax

C#

public uint Length { get; }

Visual Basic (Declaration)

Public ReadOnly Property Length As UInteger

Visual C++

public:
property unsigned int Length {
            unsigned int get ();
}

See Also

QueueDatabase Class  BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The padding character for short, fixed-length records.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public int PadByte { get; }

Visual Basic (Declaration)

Public ReadOnly Property PadByte As Integer

Visual C++

public:
property int PadByte {
    int get ()
}

See Also

QueueDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation
QueueDatabaseConfig Class

A class representing configuration parameters for QueueDatabase

**Namespace:** BerkeleyDB
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public class QueueDatabaseConfig : DatabaseConfig

Visual Basic (Declaration)

Public Class QueueDatabaseConfig _
    Inherits DatabaseConfig

Visual C++

public ref class QueueDatabaseConfig : public DatabaseConfig
Inheritance Hierarchy

System::Object
BerkeleyDB::DatabaseConfig
BerkeleyDB::QueueDatabaseConfig
See Also

QueueDatabaseConfig Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
QueueDatabaseConfig Members

QueueDatabaseConfig Class  Constructors  Methods  Fields  Properties
See Also

The QueueDatabaseConfig type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>QueueDatabaseConfig</td>
<td>Instantiate a new QueueDatabaseConfig object</td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>SetEncryption</strong></td>
<td>Set the password and algorithm used by the Berkeley DB library to perform encryption and decryption. (Inherited from <a href="#">DatabaseConfig</a>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a <a href="#">String</a> that represents the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
</tbody>
</table>
## Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Append</strong></td>
<td>A function to call after the record number has been selected but before the data has been stored into the database.</td>
</tr>
<tr>
<td><strong>AutoCommit</strong></td>
<td>Enclose the open call within a transaction. If the call succeeds, the open operation will be recoverable and all subsequent database modification operations based on this handle will be transactionally protected. If the call fails, no database will have been created. (Inherited from <code>DatabaseConfig</code>.)</td>
</tr>
<tr>
<td><strong>ByteOrder</strong></td>
<td>The byte order for integers in the stored database metadata. The host byte order of the machine where the Berkeley DB library was compiled is the default value. (Inherited from <code>DatabaseConfig</code>.)</td>
</tr>
<tr>
<td><strong>CacheSize</strong></td>
<td>The size of the shared memory buffer pool -- that is, the cache. (Inherited from <code>DatabaseConfig</code>.)</td>
</tr>
<tr>
<td><strong>ConsumeInOrder</strong></td>
<td>If true, modify the operation of <code>Consume(Boolean)</code> to return key/data pairs in order. That is, they will always return the key/data item from the head of the queue.</td>
</tr>
<tr>
<td><strong>Creation</strong></td>
<td>The policy for how to handle database creation. If true, do checksum verification of pages read into the cache from the backing filestore. (Inherited from <code>DatabaseConfig</code>.)</td>
</tr>
<tr>
<td><strong>DoChecksum</strong></td>
<td>The Berkeley DB environment within which to create a database. If null, the database will be created stand-alone; that is, it is not part of any Berkeley DB environment. (Inherited from <code>DatabaseConfig</code>.)</td>
</tr>
<tr>
<td><strong>Env</strong></td>
<td>The mechanism for reporting error messages to the</td>
</tr>
<tr>
<td>Configuration Item</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| `ErrorFeedback`    | application.  
                      (Inherited from `DatabaseConfig`.) |
| `ErrorPrefix`      | The prefix string that appears before error messages issued by Berkeley DB.  
                      (Inherited from `DatabaseConfig`.) |
| `Feedback`         | Cause the database object to be free-threaded; that is, concurrently usable by multiple threads in the address space.  
                      (Inherited from `DatabaseConfig`.) |
| `FreeThreaded`     | Do not map this database into process memory.  
                      (Inherited from `DatabaseConfig`.) |
| `NoMMap`           | If true, Berkeley DB will not write log records for this database.  
                      (Inherited from `DatabaseConfig`.) |
| `NonDurableTxns`   | The cache priority for pages referenced by the database.  
                      (Inherited from `DatabaseConfig`.) |
| `Priority`         | Open the database for reading only. Any attempt to modify items in the database will fail, regardless of the actual permissions of any underlying files.  
                      (Inherited from `DatabaseConfig`.) |
| `ReadOnly`         | Support transactional read operations with degree 1 isolation.  
                      (Inherited from `DatabaseConfig`.) |
| `ReadUncommitted`  | Physically truncate the underlying file, discarding all previous databases it might have held.  
                      (Inherited from `DatabaseConfig`.) |
| `Truncate`         | Open the database with support for multiversion concurrency control.  
                      (Inherited from `DatabaseConfig`.) |
| `UseMVCC`          | |
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EncryptAlgorithm</td>
<td>The algorithm used to perform encryption and decryption. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td>EncryptionPassword</td>
<td>The password used to perform encryption and decryption. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td>ExtentSize</td>
<td>The size of the extents used to hold pages in a QueueDatabase, specified as a number of pages.</td>
</tr>
<tr>
<td>Length</td>
<td>Specify the length of records in the database.</td>
</tr>
<tr>
<td>PadByte</td>
<td>The padding character for short, fixed-length records.</td>
</tr>
<tr>
<td>PageSize</td>
<td>The size of the pages used to hold items in the database, in bytes.</td>
</tr>
</tbody>
</table>
See Also

QueueDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Instantiate a new QueueDatabaseConfig object

Namespace: BerkeleyDB
Syntax

C#

public QueueDatabaseConfig()

Visual Basic (Declaration)

Public Sub New

Visual C++

public:
QueueDatabaseConfig()
See Also

QueueDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
QueueDatabaseConfig Fields

See Also

The QueueDatabaseConfig type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Append</td>
<td>A function to call after the record number has been selected but before the data has been stored into the database.</td>
</tr>
<tr>
<td>AutoCommit</td>
<td>Enclose the open call within a transaction. If the call succeeds, the open operation will be recoverable and all subsequent database modification operations based on this handle will be transactionally protected. If the call fails, no database will have been created. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td>ByteOrder</td>
<td>The byte order for integers in the stored database metadata. The host byte order of the machine where the Berkeley DB library was compiled is the default value. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td>CacheSize</td>
<td>The size of the shared memory buffer pool -- that is, the cache. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td>ConsumeInOrder</td>
<td>If true, modify the operation of Consume(Boolean) to return key/data pairs in order. That is, they will always return the key/data item from the head of the queue.</td>
</tr>
<tr>
<td>Creation</td>
<td>The policy for how to handle database creation. If true, do checksum verification of pages read into the cache from the backing filestore. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td>DoChecksum</td>
<td>The Berkeley DB environment within which to create a database. If null, the database will be created stand-alone; that is, it is not part of any Berkeley DB environment. (Inherited from DatabaseConfig.)</td>
</tr>
</tbody>
</table>
| Env          | The mechanism for reporting error messages to the
<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ErrorFeedback</strong></td>
<td>application. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td><strong>ErrorPrefix</strong></td>
<td>The prefix string that appears before error messages issued by Berkeley DB. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td><strong>Feedback</strong></td>
<td>Cause the database object to be free-threaded; that is, concurrently usable by multiple threads in the address space. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td><strong>FreeThreaded</strong></td>
<td>(Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td><strong>NoMMap</strong></td>
<td>Do not map this database into process memory. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td><strong>NonDurableTxns</strong></td>
<td>If true, Berkeley DB will not write log records for this database. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td><strong>Priority</strong></td>
<td>The cache priority for pages referenced by the database. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td><strong>ReadOnly</strong></td>
<td>Open the database for reading only. Any attempt to modify items in the database will fail, regardless of the actual permissions of any underlying files. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td><strong>ReadUncommitted</strong></td>
<td>Support transactional read operations with degree 1 isolation. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td><strong>Truncate</strong></td>
<td>Physically truncate the underlying file, discarding all previous databases it might have held. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td><strong>UseMVCC</strong></td>
<td>Open the database with support for multiversion concurrency control. (Inherited from DatabaseConfig.)</td>
</tr>
</tbody>
</table>
See Also

QueueDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
A function to call after the record number has been selected but before the data has been stored into the database.

**Namespace:**  [BerkeleyDB](https://www.oracle.com/technetwork/database/database-technologies/berkeleydb/)

**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#
public AppendRecordDelegate Append

Visual Basic (Declaration)
Public Append As AppendRecordDelegate

Visual C++
public: AppendRecordDelegate^ Append
Remarks

When using Append(DatabaseEntry), it may be useful to modify the stored data based on the generated key. If a delegate is specified, it will be called after the record number has been selected, but before the data has been stored.
See Also

QueueDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, modify the operation of `Consume(Boolean)` to return key/data pairs in order. That is, they will always return the key/data item from the head of the queue.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
## Syntax

### C#

`public bool ConsumeInOrder`

### Visual Basic (Declaration)

`Public ConsumeInOrder As Boolean`

### Visual C++

```cpp
public:
    bool ConsumeInOrder
```
Remarks

The default behavior of queue databases is optimized for multiple readers, and does not guarantee that record will be retrieved in the order they are added to the queue. Specifically, if a writing thread adds multiple records to an empty queue, reading threads may skip some of the initial records when the next Consume(Boolean) call returns.

This setting modifies Consume(Boolean) to verify that the record being returned is in fact the head of the queue. This will increase contention and reduce concurrency when there are many reading threads.
See Also

QueueDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The policy for how to handle database creation.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public CreatePolicy Creation

Visual Basic (Declaration)

Public Creation As CreatePolicy

Visual C++

public:
CreatePolicy Creation
Remarks

If the database does not already exist and NEVER is set, Open(String, QueueDatabaseConfig) will fail.
See Also

QueueDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The **QueueDatabaseConfig** type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <code>Object</code> is equal to the current <code>Object</code>.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <code>Type</code> of the current instance.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>SetEncryption</strong></td>
<td>Set the password and algorithm used by the Berkeley</td>
</tr>
<tr>
<td></td>
<td>DB library to perform encryption and decryption.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>DatabaseConfig</code>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a <code>String</code> that represents the current <code>Object</code>.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
</tbody>
</table>
See Also

QueueDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `QueueDatabaseConfig` type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EncryptAlgorithm</td>
<td>The algorithm used to perform encryption and decryption.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td>EncryptionPassword</td>
<td>The password used to perform encryption and decryption.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td>ExtentSize</td>
<td>The size of the extents used to hold pages in a QueueDatabase, specified as a number of pages.</td>
</tr>
<tr>
<td>Length</td>
<td>Specify the length of records in the database.</td>
</tr>
<tr>
<td>PadByte</td>
<td>The padding character for short, fixed-length records.</td>
</tr>
<tr>
<td>PageSize</td>
<td>The size of the pages used to hold items in the database, in bytes.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from DatabaseConfig.)</td>
</tr>
</tbody>
</table>
See Also

QueueDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
QueueDatabaseConfig..::.ExtentSize Property

QueueDatabaseConfig Class  See Also

The size of the extents used to hold pages in a QueueDatabase, specified as a number of pages.

Namespace: BerkeleyDB
Syntax

C#

public uint ExtentSize { get; set; }

Visual Basic (Declaration)

Public Property ExtentSize As UInteger

Visual C++

public:
property unsigned int ExtentSize {
unsigned int get ();
void set (unsigned int value);
}
Remarks

Each extent is created as a separate physical file. If no extent size is set, the default behavior is to create only a single underlying database file.

For information on tuning the extent size, see Selecting a extent size in the Programmer's Reference Guide.

If the database already exists, this setting will be ignored.
See Also

QueueDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Specify the length of records in the database.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll)  
**Version:** 4.8.24.0
Syntax

C#

```csharp
public uint Length { get; set; }
```

Visual Basic (Declaration)

```
Public Property Length AsUInteger
```

Visual C++

```cpp
public:
property unsigned int Length {
unsigned int get ();
void set (unsigned int value);
}
```
Remarks

The record length must be enough smaller than \texttt{PageSize} that at least one record plus the database page's metadata information can fit on each database page.

Any records added to the database that are less than \texttt{Length} bytes long are automatically padded (see \texttt{PadByte} for more information).

Any attempt to insert records into the database that are greater than \texttt{Length} bytes long will cause the call to fail immediately and return an error.

If the database already exists, this setting will be ignored.
See Also

QueueDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
QueueDatabaseConfig::PadByte Property

The padding character for short, fixed-length records.

**Namespace:** [BerkeleyDB](https://github.com/berkeley-db-net/berkeley-db-dotnet)

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public int PadByte { get; set; }

Visual Basic (Declaration)

Public Property PadByte As Integer

Visual C++

public:
property int PadByte {
    int get ();
    void set (int value);
}


Remarks

If no pad character is specified, space characters (that is, ASCII 0x20) are used for padding.

If the database already exists, this setting will be ignored.
See Also

QueueDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
QueueStats Class

Statistical information about a QueueDatabase

Namespace: BerkeleyDB
Syntax

C#

public class QueueStats

Visual Basic (Declaration)

Public Class QueueStats

Visual C++

public ref class QueueStats
Inheritance Hierarchy

System..:::Object
BerkeleyDB..:::QueueStats
See Also

QueueStats Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `QueueStats` type exposes the following members.
### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a> is equal to the current <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the <a href="https://docs.microsoft.com/en-us/dotnet/api/system.type">Type</a> of the current instance. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a <a href="https://docs.microsoft.com/en-us/dotnet/api/system.string">String</a> that represents the current <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.</td>
</tr>
</tbody>
</table>
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DataPages</td>
<td>Data pages.</td>
</tr>
<tr>
<td>DataPagesBytesFree</td>
<td>Bytes free in data pages.</td>
</tr>
<tr>
<td>FirstRecordNumber</td>
<td>First not deleted record.</td>
</tr>
<tr>
<td>MagicNumber</td>
<td>Magic number.</td>
</tr>
<tr>
<td>MetadataFlags</td>
<td>Metadata flags.</td>
</tr>
<tr>
<td>nData</td>
<td>Number of data items.</td>
</tr>
<tr>
<td>NextRecordNumber</td>
<td>Next available record number.</td>
</tr>
<tr>
<td>nKeys</td>
<td>Number of unique keys.</td>
</tr>
<tr>
<td>PageSize</td>
<td>Page size.</td>
</tr>
<tr>
<td>PagesPerExtent</td>
<td>Pages per extent.</td>
</tr>
<tr>
<td>RecordLength</td>
<td>Fixed-length record length.</td>
</tr>
<tr>
<td>RecordPadByte</td>
<td>Fixed-length record pad.</td>
</tr>
<tr>
<td>Version</td>
<td>Version number.</td>
</tr>
</tbody>
</table>
See Also

QueueStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The **QueueStats** type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a>.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <a href="#">Type</a> of the current instance.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a <a href="#">String</a> that represents the current <a href="#">Object</a>.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Object</a>.)</td>
</tr>
</tbody>
</table>
See Also

QueueStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `QueueStats` type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DataPages</strong></td>
<td>Data pages.</td>
</tr>
<tr>
<td><strong>DataPagesBytesFree</strong></td>
<td>Bytes free in data pages.</td>
</tr>
<tr>
<td><strong>FirstRecordNumber</strong></td>
<td>First not deleted record.</td>
</tr>
<tr>
<td><strong>MagicNumber</strong></td>
<td>Magic number.</td>
</tr>
<tr>
<td><strong>MetadataFlags</strong></td>
<td>Metadata flags.</td>
</tr>
<tr>
<td><strong>nData</strong></td>
<td>Number of data items.</td>
</tr>
<tr>
<td><strong>NextRecordNumber</strong></td>
<td>Next available record number.</td>
</tr>
<tr>
<td><strong>nKeys</strong></td>
<td>Number of unique keys.</td>
</tr>
<tr>
<td><strong>PageSize</strong></td>
<td>Page size.</td>
</tr>
<tr>
<td><strong>PagesPerExtent</strong></td>
<td>Pages per extent.</td>
</tr>
<tr>
<td><strong>RecordLength</strong></td>
<td>Fixed-length record length.</td>
</tr>
<tr>
<td><strong>RecordPadByte</strong></td>
<td>Fixed-length record pad.</td>
</tr>
<tr>
<td><strong>Version</strong></td>
<td>Version number.</td>
</tr>
</tbody>
</table>
See Also

QueueStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
QueueStats..::..DataPages Property

Data pages.

Namespace: BerkeleyDB
Syntax

C#

public uint DataPages { get; }

Visual Basic (Declaration)

Public ReadOnly Property DataPages As UInteger

Visual C++

public:
property unsigned int DataPages {
    unsigned int get ();
}

See Also

QueueStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Bytes free in data pages.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public uint DataPagesBytesFree { get; }

Visual Basic (Declaration)

Public ReadOnly Property DataPagesBytesFree As UInteger

Visual C++

public:
property unsigned int DataPagesBytesFree {
    unsigned int get ();
}

See Also

QueueStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
QueueStats..:::FirstRecordNumber Property

First not deleted record.

Namespace: BerkeleyDB
Syntax

C#

public uint FirstRecordNumber { get; }

Visual Basic (Declaration)

Public ReadOnly Property FirstRecordNumber AsUInteger

Visual C++

public:
property unsigned int FirstRecordNumber {
    unsigned int get ();
}


See Also

QueueStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
QueueStats..:::MagicNumber Property

Magic number.

**Namespace:**  BerkeleyDB  
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll)  
**Version:** 4.8.24.0
Syntax

C#

```csharp
public uint MagicNumber { get; }
```

Visual Basic (Declaration)

```vbnet
Public ReadOnly Property MagicNumber As UInteger
```

Visual C++

```cpp
public:
property unsigned int MagicNumber {
    unsigned int get ();
}
```
See Also

QueueStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
QueueStats

MetadataFlags Property

Namespace: BerkeleyDB
Syntax

C#

public uint MetadataFlags { get; }

Visual Basic (Declaration)

Public ReadOnly Property MetadataFlags As UInteger

Visual C++

public:
property unsigned int MetadataFlags {
    unsigned int get ();
}

See Also

QueueStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
QueueStats..:::nData Property

QueueStats Class  See Also

Number of data items.

Namespace:  BerkeleyDB
Syntax

C#

public uint nData { get; }

Visual Basic (Declaration)

Public ReadOnly Property nData AsUInteger

Visual C++

public:
property unsigned int nData {
    unsigned int get ();
}

See Also

QueueStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
QueueStats..::.NextRecordNumber Property

Next available record number.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public uint NextRecordNumber { get; }

Visual Basic (Declaration)

Public ReadOnly Property NextRecordNumber As UInteger

Visual C++

public:
    property unsigned int NextRecordNumber { 
        unsigned int get ();
    }
See Also

QueueStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Number of unique keys.

Namespace: BerkeleyDB
Syntax

C#

public uint nKeys { get; }

Visual Basic (Declaration)

Public ReadOnly Property nKeys As UInteger

Visual C++

public:
    property unsigned int nKeys {
        unsigned int get ();
    }

See Also

QueueStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Page size.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public uint PageSize { get; }

Visual Basic (Declaration)

Public ReadOnly Property PageSize As UInteger

Visual C++

public:
property unsigned int PageSize {
    unsigned int get ();
}

See Also

QueueStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Pages per extent.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public uint PagesPerExtent { get; }

Visual Basic (Declaration)

Public ReadOnly Property PagesPerExtent AsUInteger

Visual C++

public:
property unsigned int PagesPerExtent {
    unsigned int get ();
}

See Also

QueueStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Fixed-length record length.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public uint RecordLength { get; }

Visual Basic (Declaration)

Public ReadOnly Property RecordLength As UInteger

Visual C++

public:
property unsigned int RecordLength {
    unsigned int get ();
}


See Also

QueueStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
QueueStats::RecordPadByte Property

QueueStats Class  See Also

Fixed-length record pad.

Namespace:  BerkeleyDB
### Syntax

**C#**

```csharp
public uint RecordPadByte { get; }
```

**Visual Basic (Declaration)**

Public ReadOnly Property RecordPadByte AsUInteger

**Visual C++**

```cpp
public:
    property unsigned int RecordPadByte {
        unsigned int get ();
    }
```

See Also

QueueStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
QueueStats Class

See Also

Version number.

Namespace:  BerkeleyDB
### Syntax

**C#**

```csharp
public uint Version { get; }
```

**Visual Basic (Declaration)**

```vbnet
Public ReadOnly Property Version AsUInteger
```

**Visual C++**

```cpp
public:
property unsigned int Version {
    unsigned int get ();
}
```
See Also

QueueStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
A class for traversing the records of a RecnoDatabase

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public class RecnoCursor : Cursor

Visual Basic (Declaration)

Public Class RecnoCursor _
    Inherits Cursor

Visual C++

public ref class RecnoCursor : public Cursor
Inheritance Hierarchy

System...:::Object
BerkeleyDB...:::BaseCursor
    BerkeleyDB...:::Cursor
    BerkeleyDB...:::RecnoCursor
See Also

RecnoCursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The RecnoCursor type exposes the following members.
# Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Add</strong></td>
<td>Stores the key/data pair in the database. (Inherited from <a href="#">Cursor</a>.)</td>
</tr>
<tr>
<td><strong>Close</strong></td>
<td>Discard the cursor. It is possible for the Close() method to throw a <a href="#">DeadlockException</a>, signaling that any enclosing transaction should be aborted. If the application is already intending to abort the transaction, this error should be ignored, and the application should proceed. After Close has been called, regardless of its result, the object may not be used again. (Inherited from <a href="#">BaseCursor</a>.)</td>
</tr>
<tr>
<td><strong>Compare</strong></td>
<td>Compare this cursor's position to another's. (Inherited from <a href="#">BaseCursor</a>.)</td>
</tr>
<tr>
<td><strong>Count</strong></td>
<td>Returns a count of the number of data items for the key to which the cursor refers. (Inherited from <a href="#">BaseCursor</a>.)</td>
</tr>
<tr>
<td><strong>Delete</strong></td>
<td>Delete the key/data pair to which the cursor refers. (Inherited from <a href="#">Cursor</a>.)</td>
</tr>
<tr>
<td><strong>Dispose</strong></td>
<td>Release the resources held by this object, and close the cursor if it's still open.</td>
</tr>
</tbody>
</table>
Duplicate

Create a new cursor that uses the same transaction and locker ID as the original cursor.

Equals

Determines whether the specified Object is equal to the current Object.

GetEnumerator

Returns an enumerator that iterates through the Cursor.

GetHashCode

Serves as a hash function for a particular type.

GetType

Gets the Type of the current instance.

Insert

Insert the data element as a duplicate element of the key to which the cursor refers.

Move

Overloaded.

MoveFirst

Overloaded.

MoveFirstMultiple

Overloaded.

MoveFirstMultipleKey

Overloaded.

MoveLast

Overloaded.

MoveDuplicate

Overloaded.

MoveDuplicateMultiple

Overloaded.

MoveDuplicateMultipleKey

Overloaded.

MoveNext

Overloaded.

MoveNextDuplicate

Overloaded.

MoveNextDuplicateMultiple

Overloaded.

MoveNextDuplicateMultipleKey

Overloaded.

MoveNextMultiple

Overloaded.

MoveNextMultipleKey

Overloaded.
<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MoveNextUnique</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>MoveNextUniqueMultiple</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>MoveNextUniqueMultipleKey</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>MovePrev</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>MovePrevDuplicate</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>MovePrevUnique</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>Overwrite</strong></td>
<td>Overwrite the data of the key/data pair to which the cursor refers with the specified data item. (Inherited from Cursor.)</td>
</tr>
<tr>
<td><strong>Refresh</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>RefreshMultiple</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>RefreshMultipleKey</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a String that represents the current Object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current</td>
<td>The key/data pair at which the cursor currently points.</td>
</tr>
<tr>
<td>CurrentMultiple</td>
<td>The key and multiple data items at which the cursor currently points.</td>
</tr>
<tr>
<td>CurrentMultipleKey</td>
<td>The multiple key and data items at which the cursor currently points.</td>
</tr>
<tr>
<td>Priority</td>
<td>The cache priority for pages referenced by the cursor.</td>
</tr>
</tbody>
</table>
See Also

RecnoCursor Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation
RecnoCursor Methods

The **RecnoCursor** type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add</td>
<td>Stores the key/data pair in the database. (Inherited from Cursor.)</td>
</tr>
<tr>
<td>Close</td>
<td>Discard the cursor. It is possible for the Close() method to throw a DeadlockException, signaling that any enclosing transaction should be aborted. If the application is already intending to abort the transaction, this error should be ignored, and the application should proceed. After Close has been called, regardless of its result, the object may not be used again. (Inherited from BaseCursor.)</td>
</tr>
<tr>
<td>Compare</td>
<td>Compare this cursor's position to another's. (Inherited from BaseCursor.)</td>
</tr>
<tr>
<td>Count</td>
<td>Returns a count of the number of data items for the key to which the cursor refers. (Inherited from BaseCursor.)</td>
</tr>
<tr>
<td>Delete</td>
<td>Delete the key/data pair to which the cursor refers. (Inherited from Cursor.)</td>
</tr>
<tr>
<td>Dispose</td>
<td>Release the resources held by this object, and close the cursor if it's still open.</td>
</tr>
</tbody>
</table>
Duplicate
Create a new cursor that uses the same transaction and locker ID as the original cursor.

Equals
Determines whether the specified Object is equal to the current Object.

GetEnumerator
Returns an enumerator that iterates through the Cursor.

GetHashCode
Serves as a hash function for a particular type.

GetType
Gets the Type of the current instance.

Insert
Insert the data element as a duplicate element of the key to which the cursor refers.

Move
Overloaded.

MoveFirst
Overloaded.

MoveFirstMultiple
Overloaded.

MoveFirstMultipleKey
Overloaded.

MoveLast
Overloaded.

MoveMultiple
Overloaded.

MoveMultipleKey
Overloaded.

MoveNext
Overloaded.

MoveNextDuplicate
Overloaded.

MoveNextDuplicateMultiple
Overloaded.

MoveNextDuplicateMultipleKey
Overloaded.

MoveNextMultiple
Overloaded.

MoveNextMultipleKey
Overloaded.
- **MoveNextUnique** Overloaded.
- **MoveNextUniqueMultiple** Overloaded.
- **MoveNextUniqueMultipleKey** Overloaded.
- **MovePrev** Overloaded.
- **MovePrevDuplicate** Overloaded.
- **MovePrevUnique** Overloaded.
- **Overwrite**
  Overwrite the data of the key/data pair to which the cursor refers with the specified data item.
  (Inherited from **Cursor**.)
- **Refresh** Overloaded.
- **RefreshMultiple** Overloaded.
- **RefreshMultipleKey** Overloaded.
- **ToString**
  Returns a **String** that represents the current **Object**.
  (Inherited from **Object**.)
See Also

RecnoCursor Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Create a new cursor that uses the same transaction and locker ID as the original cursor.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
### Syntax

#### C#

```csharp
public RecnoCursor Duplicate(
    bool keepPosition
)
```

#### Visual Basic (Declaration)

```vbnet
Public Function Duplicate ( _
    keepPosition As Boolean _
) As RecnoCursor
```

#### Visual C++

```cpp
public:
    RecnoCursor^ Duplicate(
        bool keepPosition
    )
```

### Parameters

`keepPosition`  
Type: `System::::Boolean`  
If true, the newly created cursor is initialized to refer to the same position in the database as the original cursor (if any) and hold the same locks (if any). If false, or the original cursor does not hold a database position and locks, the created cursor is uninitialized and will behave like a cursor newly created by `Cursor()()()`.  

### Return Value

A newly created cursor
Remarks

This is useful when an application is using locking and requires two or more cursors in the same thread of control.
See Also

RecnoCursor Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Insert the data element as a duplicate element of the key to which the cursor refers.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

```csharp
public void Insert(
    DatabaseEntry data,
    Cursor...::InsertLocation loc
)
```

Visual Basic (Declaration)

```vbnet
Public Sub Insert (_
    data As DatabaseEntry, _
    loc As Cursor...::InsertLocation _
)
```

Visual C++

```cpp
public:
    void Insert(
        DatabaseEntry^ data,
        Cursor...::InsertLocation loc
    )
```

Parameters

data
Type: BerkleyDB...::DatabaseEntry
The data element to insert

loc
Type: BerkleyDB...::Cursor...::InsertLocation
Specify whether to insert the data item immediately before or immediately after the cursor's current position.
See Also

RecnoCursor Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
RecnoCursor.

RecnoCursor Class  See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Move(DatabaseEntry, Boolean)</td>
<td>Set the cursor to refer to key, and store the datum associated with the given key in Current. In the presence of duplicate key values, the first data item in the set of duplicates is stored in Current. (Inherited from Cursor.)</td>
</tr>
<tr>
<td>Move(KeyValuePair(Of &lt;(DatabaseEntry, DatabaseEntry)&gt;, Boolean))</td>
<td>Move the cursor to the specified key/data pair of the database. The cursor is positioned to a key/data pair if both the key and data match the values provided on the key and data parameters. (Inherited from Cursor.)</td>
</tr>
<tr>
<td>Move(DatabaseEntry, Boolean, LockingInfo)</td>
<td>Set the cursor to refer to key, and store the datum associated with the given key in Current. In the presence of duplicate key values, the first data item in the set of duplicates is stored in Current. (Inherited from Cursor.)</td>
</tr>
<tr>
<td>Move(KeyValuePair(Of &lt;(DatabaseEntry, DatabaseEntry)&gt;, Boolean, LockingInfo))</td>
<td>Move the cursor to the specified key/data pair of the database. The cursor is positioned to a key/data pair if both the key and data match the values provided on the key and data parameters. (Inherited from Cursor.)</td>
</tr>
</tbody>
</table>
See Also

Recno.Cursor Class
Recno.Cursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
RecnoCursor Class  See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MoveFirst</td>
<td>Set the cursor to refer to the first key/data pair of the database, and store that pair in Current. If the first key has duplicate values, the first data item in the set of duplicates is stored in Current. (Inherited from Cursor.)</td>
</tr>
<tr>
<td>MoveFirst(LockingInfo)</td>
<td>Set the cursor to refer to the first key/data pair of the database, and store that pair in Current. If the first key has duplicate values, the first data item in the set of duplicates is stored in Current. (Inherited from Cursor.)</td>
</tr>
</tbody>
</table>
See Also

RecnoCursor Class
RecnoCursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If positioning the cursor fails, `CurrentMultiple` will contain an empty `KeyValuePair<(Of <(TKey, TValue)>)>`. 

**RecnoCursor**...:.MoveFirstMultiple Method

**RecnoCursor Class**  **See Also**
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="MoveFirstMultiple()()" /></td>
<td>Set the cursor to refer to the first key/data pair of the database, and store that key and as many duplicate data items that can fit in a buffer the size of one database page in <code>CurrentMultiple</code>. (Inherited from <code>Cursor</code>.)</td>
</tr>
<tr>
<td><img src="image" alt="MoveFirstMultiple(LockingInfo)" /></td>
<td>Set the cursor to refer to the first key/data pair of the database, and store that key and as many duplicate data items that can fit in a buffer the size of one database page in <code>CurrentMultiple</code>. (Inherited from <code>Cursor</code>.)</td>
</tr>
<tr>
<td><img src="image" alt="MoveFirstMultiple(Int32)" /></td>
<td>Set the cursor to refer to the first key/data pair of the database, and store that key and as many duplicate data items that can fit in a buffer the size of BufferSize in <code>CurrentMultiple</code>. (Inherited from <code>Cursor</code>.)</td>
</tr>
<tr>
<td><img src="image" alt="MoveFirstMultiple(Int32, LockingInfo)" /></td>
<td>Set the cursor to refer to the first key/data pair of the database, and store that key and as many duplicate data items that can fit in a buffer the size of BufferSize in <code>CurrentMultiple</code>. (Inherited from <code>Cursor</code>.)</td>
</tr>
</tbody>
</table>
See Also

RecnoCursor Class
RecnoCursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
RecnoCursor="..:.MoveFirstMultipleKey Method"

RecnoCursor Class  See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MoveFirstMultipleKey()()</td>
<td>Set the cursor to refer to the first key/data pair of the database, and store that pair and as many ensuing key/data pairs that can fit in a buffer the size of one database page in CurrentMultipleKey. (Inherited from Cursor.)</td>
</tr>
<tr>
<td>MoveFirstMultipleKey(LockingInfo)</td>
<td>Set the cursor to refer to the first key/data pair of the database, and store that pair and as many ensuing key/data pairs that can fit in a buffer the size of one database page in CurrentMultipleKey. (Inherited from Cursor.)</td>
</tr>
<tr>
<td>MoveFirstMultipleKey(Int32)</td>
<td>Set the cursor to refer to the first key/data pair of the database, and store that pair and as many ensuing key/data pairs that can fit in a buffer the size of BufferSize in CurrentMultipleKey. (Inherited from Cursor.)</td>
</tr>
<tr>
<td>MoveFirstMultipleKey(Int32, LockingInfo)</td>
<td>Set the cursor to refer to the first key/data pair of the database, and store that pair and as many ensuing key/data pairs that can fit in a buffer the size of BufferSize in CurrentMultipleKey. (Inherited from Cursor.)</td>
</tr>
</tbody>
</table>
See Also

RecnoCursor Class
RecnoCursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
RecnoCursor Class

See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MoveLast()()</td>
<td>Set the cursor to refer to the last key/data pair of the database, and store that pair in Current. (Inherited from Cursor.)</td>
</tr>
<tr>
<td></td>
<td>If the last key has duplicate values, the last data item in the set of duplicates is stored in Current.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Cursor.)</td>
</tr>
<tr>
<td>MoveLast(LockingInfo)</td>
<td>Set the cursor to refer to the last key/data pair of the database, and store that pair in Current.</td>
</tr>
<tr>
<td></td>
<td>If the last key has duplicate values, the last data item in the set of duplicates is stored in Current.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Cursor.)</td>
</tr>
</tbody>
</table>

See Also

RecnoCursor Class
RecnoCursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
RecnoCursor.

RecnoCursor Class  See Also
# Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MoveMultiple(DatabaseEntry, Boolean)</strong></td>
<td>Set the cursor to refer to key, and store that key and as many duplicate data items associated with the given key that can fit in a buffer the size of one database page in CurrentMultiple. (Inherited from Cursor.)</td>
</tr>
<tr>
<td><strong>MoveMultiple(KeyValuePair&lt;Of (DatabaseEntry, DatabaseEntry)&gt;, Boolean)</strong></td>
<td>Move the cursor to the specified key/data pair of the database, and store that key/data pair and as many duplicate data items associated with the given key that can fit in a buffer the size of one database page in CurrentMultiple. The cursor is positioned to a key/data pair if both the key and data match the values provided on the key and data parameters. (Inherited from Cursor.)</td>
</tr>
<tr>
<td><strong>MoveMultiple(DatabaseEntry, Boolean, LockingInfo)</strong></td>
<td>Set the cursor to refer to key, and store that key and as many duplicate data items associated with the given key that can fit in a buffer the size of one database page in CurrentMultiple. (Inherited from Cursor.)</td>
</tr>
<tr>
<td><strong>MoveMultiple(DatabaseEntry, Boolean, Int32)</strong></td>
<td>Set the cursor to refer to key, and store that key and as many duplicate data items associated with the given key that can fit in a buffer the size of BufferSize in CurrentMultiple. (Inherited from Cursor.)</td>
</tr>
</tbody>
</table>
MoveMultiple(KeyValuePair(Of DatabaseEntry, DatabaseEntry), Boolean, LockingInfo)

MoveMultiple(KeyValuePair(Of DatabaseEntry, DatabaseEntry), Boolean, Int32, LockingInfo)

MoveMultiple(DatabaseEntry, Boolean, Int32, LockingInfo)

MoveMultiple(KeyValuePair(Of DatabaseEntry, DatabaseEntry), Boolean, Int32, LockingInfo)

Move the cursor to the specified key/data pair of the database, and store that key/data pair and as many duplicate data items associated with the given key that can fit in a buffer the size of one database page in CurrentMultiple. The cursor is positioned to a key/data pair if both the key and data match the values provided on the key and data parameters. (Inherited from Cursor.)

Move the cursor to the specified key/data pair of the database, and store that key/data pair and as many duplicate data items associated with the given key that can fit in a buffer the size of BufferSize in CurrentMultiple. The cursor is positioned to a key/data pair if both the key and data match the values provided on the key and data parameters. (Inherited from Cursor.)

Set the cursor to refer to key, and store that key and as many duplicate data items associated with the given key that can fit in a buffer the size of BufferSize in CurrentMultiple. (Inherited from Cursor.)

Move the cursor to the specified key/data pair of the database, and store that key/data pair and as many duplicate data items associated with the given key that can fit in a buffer the size of BufferSize in CurrentMultiple. The cursor is positioned to a key/data pair if both
the key and data match the values provided on the key and data parameters.
(Inherited from Cursor.)
See Also

RecnoCursor Class
RecnoCursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
RecnoCursor.

::

MoveMultipleKey Method

RecnoCursor Class  See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MoveMultipleKey(DatabaseEntry, Boolean)</strong></td>
<td>Set the cursor to refer to key, and store that key and as many ensuing key/data pairs that can fit in a buffer the size of one database page in CurrentMultipleKey. (Inherited from Cursor.)</td>
</tr>
<tr>
<td><strong>MoveMultipleKey(KeyValuePair&lt;Of &lt;(DatabaseEntry, DatabaseEntry)&gt;&gt;, Boolean)</strong></td>
<td>Move the cursor to the specified key/data pair of the database, and store that key/data pair and as many ensuing key/data pairs that can fit in a buffer the size of one database page in CurrentMultipleKey. The cursor is positioned to a key/data pair if both the key and data match the values provided on the key and data parameters. (Inherited from Cursor.)</td>
</tr>
<tr>
<td><strong>MoveMultipleKey(DatabaseEntry, Boolean, LockingInfo)</strong></td>
<td>Set the cursor to refer to key, and store that key and as many ensuing key/data pairs that can fit in a buffer the size of one database page in CurrentMultipleKey. (Inherited from Cursor.)</td>
</tr>
<tr>
<td><strong>MoveMultipleKey(DatabaseEntry, Boolean, Int32)</strong></td>
<td>Set the cursor to refer to key, and store that key and as many ensuing key/data pairs that can fit in a buffer the size of BufferSize in CurrentMultipleKey.</td>
</tr>
</tbody>
</table>
MoveMultipleKey(KeyValuePair<
(DatabaseEntry, DatabaseEntry>),
Boolean, LockingInfo)

Move the cursor to the specified key/data pair of the database, and store that key/data pair and as many ensuing key/data pairs that can fit in a buffer the size of one database page in CurrentMultipleKey. The cursor is positioned to a key/data pair if both the key and data match the values provided on the key and data parameters.

(Inherited from Cursor.)

MoveMultipleKey(KeyValuePair<
(DatabaseEntry, DatabaseEntry>),
Boolean, Int32)

Move the cursor to the specified key/data pair of the database, and store that key/data pair and as many ensuing key/data pairs that can fit in a buffer the size of BufferSize in CurrentMultipleKey. The cursor is positioned to a key/data pair if both the key and data match the values provided on the key and data parameters.

(Inherited from Cursor.)

MoveMultipleKey(DatabaseEntry,
Boolean, Int32, LockingInfo)

Set the cursor to refer to key, and store that key and as many ensuing key/data pairs that can fit in a buffer the size of BufferSize in CurrentMultipleKey.

(Inherited from Cursor.)

MoveMultipleKey(KeyValuePair<
(DatabaseEntry, DatabaseEntry>),
Boolean, LockingInfo)

Move the cursor to the specified key/data pair of the database, and store that key/data pair and as many ensuing key/data pairs that can fit in a buffer the size of BufferSize in
<DatabaseEntry, DatabaseEntry>, CurrentMultipleKey. The cursor is positioned to a key/data pair if both the key and data match the values provided on the key and data parameters. (Inherited from Cursor.)
See Also

RecnoCursor Class
RecnoCursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
RecnoCursor::MoveNext Method

RecnoCursor Class  See Also
### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MoveNext()()</td>
<td>If the cursor is not yet initialized, MoveNext is identical to MoveFirst(). Otherwise, move the cursor to the next key/data pair of the database, and store that pair in Current. In the presence of duplicate key values, the value of Current.Key may not change. (Inherited from Cursor.)</td>
</tr>
<tr>
<td>MoveNext(LockingInfo)</td>
<td>If the cursor is not yet initialized, MoveNext is identical to MoveFirst(LockingInfo). Otherwise, move the cursor to the next key/data pair of the database, and store that pair in Current. In the presence of duplicate key values, the value of Current.Key may not change. (Inherited from Cursor.)</td>
</tr>
</tbody>
</table>
See Also

RecnoCursor Class
RecnoCursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
C#  Visual Basic
Visual C++
Include Protected Members
Include Inherited Members
Berkeley DB .NET API Documentation
RecnoCursor...::MoveNextDuplicate Method

RecnoCursor Class  See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MoveNextDuplicate</strong></td>
<td>If the next key/data pair of the database is a duplicate data record for the current key/data pair, move the cursor to the next key/data pair in the database, and store that pair in <strong>Current</strong>. MoveNextDuplicate will return false if the next key/data pair of the database is not a duplicate data record for the current key/data pair. (Inherited from <strong>Cursor</strong>.)</td>
</tr>
<tr>
<td><strong>MoveNextDuplicate(LockingInfo)</strong></td>
<td><strong>Current</strong>. MoveNextDuplicate will return false if the next key/data pair of the database is not a duplicate data record for the current key/data pair. (Inherited from <strong>Cursor</strong>.)</td>
</tr>
</tbody>
</table>
See Also

RecnoCursor Class  
RecnoCursor Members  
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation

RecnoCursor...:MoveNextDuplicateMultiple Method

RecnoCursor Class  See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MoveNextDuplicateMultiple()()()</td>
<td>If the next key/data pair of the database is a duplicate data record for the current key/data pair, move the cursor to the next key/data pair in the database, and store that pair and as many duplicate data items that can fit in a buffer the size of one database page in <code>CurrentMultiple</code>. MoveNextDuplicateMultiple will return false if the next key/data pair of the database is not a duplicate data record for the current key/data pair. (Inherited from <code>Cursor</code>.)</td>
</tr>
<tr>
<td>MoveNextDuplicateMultiple(LockingInfo)</td>
<td>If the next key/data pair of the database is a duplicate data record for the current key/data pair, move the cursor to the next key/data pair in the database, and store that pair and as many duplicate data items that can fit in a buffer the size of one database page in <code>CurrentMultiple</code>. MoveNextDuplicateMultiple will return false if the next key/data pair of the database is not a duplicate data record for the current key/data pair.</td>
</tr>
</tbody>
</table>
MoveNextDuplicateMultiple(Int32)

(Inherited from Cursor.)
If the next key/data pair of the database is a duplicate data record for the current key/data pair, then move cursor to the next key/data pair in the database, and store that pair and as many duplicate data items that can fit in a buffer the size of BufferSize in CurrentMultiple.
MoveNextDuplicateMultiple will return false if the next key/data pair of the database is not a duplicate data record for the current key/data pair.

MoveNextDuplicateMultiple(Int32, LockingInfo)

(Inherited from Cursor.)
If the next key/data pair of the database is a duplicate data record for the current key/data pair, move the cursor to the next key/data pair in the database, and store that pair and as many duplicate data items that can fit in a buffer the size of BufferSize in CurrentMultiple.
MoveNextDuplicateMultiple will return false if the next key/data pair of the database is not a duplicate data record for the current key/data pair.
See Also

RecnoCursor Class
RecnoCursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
RecnoCursor::MoveNextDuplicateMultipleKey Method

RecnoCursor Class  See Also
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MoveNextDuplicateMultipleKey()</strong></td>
<td>If the next key/data pair of the database is a duplicate data record for the current key/data pair, move the cursor to the next key/data pair in the database, and store that pair and as many duplicate data items that can fit in a buffer the size of one database page in <code>CurrentMultipleKey</code>. MoveNextDuplicateMultipleKey will return false if the next key/data pair of the database is not a duplicate data record for the current key/data pair. (Inherited from <strong>Cursor</strong>.)</td>
</tr>
<tr>
<td><strong>MoveNextDuplicateMultipleKey(LockingInfo)</strong></td>
<td>If the next key/data pair of the database is a duplicate data record for the current key/data pair, move the cursor to the next key/data pair in the database, and store that pair and as many duplicate data items that can fit in a buffer the size of one database page in <code>CurrentMultipleKey</code>. MoveNextDuplicateMultipleKey will return false if the next key/data pair of the database is not a duplicate data record for the current key/data pair. (Inherited from <strong>Cursor</strong>.)</td>
</tr>
</tbody>
</table>
MoveNextDuplicateMultipleKey(Int32) database is a duplicate data record for the current key/data pair, move the cursor to the next key/data pair in the database, and store that pair and as many duplicate data items that can fit in a buffer the size of BufferSize in CurrentMultipleKey. MoveNextDuplicateMultipleKey will return false if the next key/data pair of the database is not a duplicate data record for the current key/data pair. (Inherited from Cursor.)

MoveNextDuplicateMultipleKey(Int32, LockingInfo) If the next key/data pair of the database is a duplicate data record for the current key/data pair, move the cursor to the next key/data pair in the database, and store that pair and as many duplicate data items that can fit in a buffer the size of BufferSize in CurrentMultipleKey. MoveNextDuplicateMultipleKey will return false if the next key/data pair of the database is not a duplicate data record for the current key/data pair. (Inherited from Cursor.)
See Also

RecnoCursor Class
RecnoCursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
RecnoCursor Class  See Also
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MoveNextMultiple()()</td>
<td>If the cursor is not yet initialized, MoveNextMultiple is identical to MoveFirstMultiple(). Otherwise, move the cursor to the next key/data pair of the database, and store that pair and as many duplicate data items that can fit in a buffer the size of one database page in CurrentMultiple. In the presence of duplicate key values, the value of CurrentMultiple.Key may not change. (Inherited from Cursor.)</td>
</tr>
<tr>
<td>MoveNextMultiple(LockingInfo)</td>
<td>If the cursor is not yet initialized, MoveNextMultiple is identical to MoveFirstMultiple(LockingInfo). Otherwise, move the cursor to the next key/data pair of the database, and store that pair and as many duplicate data items that can fit in a buffer the size of one database page in CurrentMultiple. In the presence of duplicate key values, the value of CurrentMultiple.Key may not change. (Inherited from Cursor.)</td>
</tr>
<tr>
<td>MoveNextMultiple(Int32)</td>
<td>If the cursor is not yet initialized, MoveNextMultiple is identical to MoveFirstMultiple(Int32). Otherwise, move the cursor to the next key/data pair of the database, and store that pair and as many duplicate data items that can fit in a buffer the size of BufferSize in CurrentMultiple. In the presence of duplicate key values, the</td>
</tr>
</tbody>
</table>
MoveNextMultiple(Int32, LockingInfo)

value of `CurrentMultiple.Key` may not change.
(Inherited from `Cursor`.)

If the cursor is not yet initialized, MoveNextMultiple is identical to `MoveFirstMultiple(Int32, LockingInfo)`. Otherwise, move the cursor to the next key/data pair of the database, and store that pair and as many duplicate data items that can fit in a buffer the size of BufferSize in `CurrentMultiple`. In the presence of duplicate key values, the value of `CurrentMultiple.Key` may not change.
(Inherited from `Cursor`.)
See Also

RecnoCursor Class
RecnoCursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation
RecnoCursor..::.MoveNextMultipleKey Method

RecnoCursor Class  See Also
Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MoveNextMultipleKey()()</strong></td>
<td>If the cursor is not yet initialized, MoveNextMultipleKey is identical to MoveFirstMultipleKey(). Otherwise, move the cursor to the next key/data pair of the database, and store that pair and as many ensuing key/data pairs that can fit in a buffer the size of one database page in CurrentMultipleKey. In the presence of duplicate key values, the keys of CurrentMultipleKey may not change. (Inherited from Cursor.)</td>
</tr>
<tr>
<td><strong>MoveNextMultipleKey(LockingInfo)</strong></td>
<td>If the cursor is not yet initialized, MoveNextMultipleKey is identical to MoveFirstMultipleKey(LockingInfo). Otherwise, move the cursor to the next key/data pair of the database, and store that pair and as many ensuing key/data pairs that can fit in a buffer the size of one database page in CurrentMultipleKey. In the presence of duplicate key values, the keys of CurrentMultipleKey may not change. (Inherited from Cursor.)</td>
</tr>
<tr>
<td><strong>MoveNextMultipleKey(Int32)</strong></td>
<td>If the cursor is not yet initialized, MoveNextMultipleKey is identical to MoveFirstMultipleKey(Int32). Otherwise, move the cursor to the next key/data pair of the database, and store that pair and as many ensuing key/data pairs that can fit in a buffer the size of one database page in CurrentMultipleKey. In the presence of duplicate key values, the keys of CurrentMultipleKey may not change. (Inherited from Cursor.)</td>
</tr>
</tbody>
</table>
buffer the size of BufferSize in `CurrentMultipleKey`. In the presence of duplicate key values, the keys of `CurrentMultipleKey` may not change. (Inherited from `Cursor`.)

If the cursor is not yet initialized, `MoveNextMultipleKey` is identical to `MoveFirstMultipleKey(Int32, LockingInfo)`. Otherwise, move the cursor to the next key/data pair of the database, and store that pair and as many ensuing key/data pairs that can fit in a buffer the size of BufferSize in `CurrentMultipleKey`. In the presence of duplicate key values, the keys of `CurrentMultipleKey` may not change. (Inherited from `Cursor`.)
See Also

RecnoCursor Class
RecnoCursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
RecnoCursor.

::

MoveNextUnique Method

RecnoCursor Class  See Also
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MoveNextUnique()()()</td>
<td>If the cursor is not yet initialized, MoveNextUnique is identical to MoveFirst(). Otherwise, move the cursor to the next non-duplicate key in the database, and store that key and associated datum in <strong>Current</strong>. MoveNextUnique will return false if no non-duplicate key/data pairs exist after the cursor position in the database. (Inherited from <strong>Cursor</strong>.)</td>
</tr>
<tr>
<td>MoveNextUnique(LockingInfo)</td>
<td>If the cursor is not yet initialized, MoveNextUnique is identical to MoveFirst(LockingInfo). Otherwise, move the cursor to the next non-duplicate key in the database, and store that key and associated datum in <strong>Current</strong>. MoveNextUnique will return false if no non-duplicate key/data pairs exist after the cursor position in the database. (Inherited from <strong>Cursor</strong>.)</td>
</tr>
</tbody>
</table>
See Also

RecnoCursor Class
RecnoCursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
RecnoCursor Class  See Also
**Overload List**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MoveNextUniqueMultiple()()</td>
<td>If the cursor is not yet initialized, MoveNextUniqueMultiple is identical to <strong>MoveFirstMultiple()()</strong>. Otherwise, move the cursor to the next non-duplicate key in the database, and store that key and associated datum and as many duplicate data items that can fit in a buffer the size of one database page in <strong>CurrentMultiple</strong>. MoveNextUniqueMultiple will return false if no non-duplicate key/data pairs exist after the cursor position in the database. (Inherited from <strong>Cursor</strong>.)</td>
</tr>
<tr>
<td>MoveNextUniqueMultiple(LockingInfo)</td>
<td>If the cursor is not yet initialized, MoveNextUniqueMultiple is identical to <strong>MoveFirstMultiple(LockingInfo)</strong>. Otherwise, move the cursor to the next non-duplicate key in the database, and store that key and associated datum and as many duplicate data items that can fit in a buffer the size of one database page in <strong>CurrentMultiple</strong>. MoveNextUniqueMultiple will return false if no non-duplicate key/data pairs exist after the cursor position in the database. (Inherited from <strong>Cursor</strong>.)</td>
</tr>
</tbody>
</table>
**MoveNextUniqueMultiple(Int32)**

MoveNextUniqueMultiple is identical to `MoveFirstMultiple(Int32)`. Otherwise, move the cursor to the next non-duplicate key in the database, and store that key and associated datum and as many duplicate data items that can fit in a buffer the size of BufferSize in `CurrentMultiple`. MoveNextUniqueMultiple will return false if no non-duplicate key/data pairs exist after the cursor position in the database. (Inherited from `Cursor`.)

If the cursor is not yet initialized, MoveNextUniqueMultiple is identical to `MoveFirstMultiple(Int32, LockingInfo)`. Otherwise, move the cursor to the next non-duplicate key in the database, and store that key and associated datum and as many duplicate data items that can fit in a buffer the size of BufferSize in `CurrentMultiple`. MoveNextUniqueMultiple will return false if no non-duplicate key/data pairs exist after the cursor position in the database. (Inherited from `Cursor`.)
See Also

RecnoCursor Class
RecnoCursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
RecnoCursor....MoveNextUniqueMultipleKey Method

RecnoCursor Class  See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MoveNextUniqueMultipleKey()()</td>
<td>If the cursor is not yet initialized, <code>MoveNextUniqueMultipleKey</code> is identical to <code>MoveFirstMultipleKey()</code>. Otherwise, move the cursor to the next non-duplicate key in the database, and store that key and associated datum and as many ensuing key/data pairs that can fit in a buffer the size of one database page in <code>CurrentMultipleKey</code>. <code>MoveNextUniqueMultipleKey</code> will return false if no non-duplicate key/data pairs exist after the cursor position in the database. (Inherited from <code>Cursor</code>.)</td>
</tr>
<tr>
<td>MoveNextUniqueMultipleKey(LockingInfo)</td>
<td>If the cursor is not yet initialized, <code>MoveNextUniqueMultipleKey</code> is identical to <code>MoveFirstMultipleKey(LockingInfo)</code>. Otherwise, move the cursor to the next non-duplicate key in the database, and store that key and associated datum and as many ensuing key/data pairs that can fit in a buffer the size of one database page in <code>CurrentMultipleKey</code>. <code>MoveNextUniqueMultipleKey</code> will return false if no non-duplicate key/data pairs exist after the cursor position in the database. (Inherited from <code>Cursor</code>.)</td>
</tr>
</tbody>
</table>

MoveNextUniqueMultipleKey(Int32)

identical to MoveFirstMultipleKey(Int32).
Otherwise, move the cursor to the next non-duplicate key in the database, and store that key and associated datum and as many ensuing key/data pairs that can fit in a buffer the size of BufferSize in CurrentMultipleKey.

MoveNextUniqueMultipleKey will return false if no non-duplicate key/data pairs exist after the cursor position in the database.
(Inherited from Cursor.)

If the cursor is not yet initialized, MoveNextUniqueMultipleKey is identical to MoveFirstMultipleKey(Int32, LockingInfo). Otherwise, move the cursor to the next non-duplicate key in the database, and store that key and associated datum and as many ensuing key/data pairs that can fit in a buffer the size of BufferSize in CurrentMultipleKey.

MoveNextUniqueMultipleKey will return false if no non-duplicate key/data pairs exist after the cursor position in the database.
(Inherited from Cursor.)
See Also

RecnoCursor Class
RecnoCursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
RecnoCursor Class  See Also
### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MovePrev()()</td>
<td>If the cursor is not yet initialized, MovePrev is identical to <code>MoveLast()()</code>. Otherwise, move the cursor to the previous key/data pair of the database, and store that pair in <code>Current</code>. In the presence of duplicate key values, the value of <code>Current.Key</code> may not change. (Inherited from <code>Cursor</code>.)</td>
</tr>
<tr>
<td>MovePrev(LockingInfo)</td>
<td>If the cursor is not yet initialized, MovePrev is identical to <code>MoveLast(LockingInfo)</code>. Otherwise, move the cursor to the previous key/data pair of the database, and store that pair in <code>Current</code>. In the presence of duplicate key values, the value of <code>Current.Key</code> may not change. (Inherited from <code>Cursor</code>.)</td>
</tr>
</tbody>
</table>
See Also

RecnoCursor Class
RecnoCursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
RecnoCursor::MovePrevDuplicate Method

RecnoCursor Class  See Also
### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MovePrevDuplicate()</strong></td>
<td>If the previous key/data pair of the database is a duplicate data record for the current key/data pair, the cursor is moved to the previous key/data pair of the database, and that pair is stored in <strong>Current</strong>. (Inherited from <strong>Cursor</strong>.)</td>
</tr>
<tr>
<td><strong>MovePrevDuplicate(LockingInfo)</strong></td>
<td>MovePrevDuplicate will return false if the previous key/data pair of the database is not a duplicate data record for the current key/data pair. (Inherited from <strong>Cursor</strong>.)</td>
</tr>
</tbody>
</table>
See Also

RecnoCursor Class
RecnoCursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation
RecnoCursor...:MovePrevUnique Method

RecnoCursor Class  See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MovePrevUnique()()</td>
<td>If the cursor is not yet initialized, MovePrevUnique is identical to MoveLast(). Otherwise, move the cursor to the previous non-duplicate key in the database, and store that key and associated datum in Current. MovePrevUnique will return false if no non-duplicate key/data pairs exist after the cursor position in the database. (Inherited from Cursor.)</td>
</tr>
<tr>
<td>MovePrevUnique(LockingInfo)</td>
<td>If the cursor is not yet initialized, MovePrevUnique is identical to MoveLast(LockingInfo). Otherwise, move the cursor to the previous non-duplicate key in the database, and store that key and associated datum in Current. MovePrevUnique will return false if no non-duplicate key/data pairs exist after the cursor position in the database. (Inherited from Cursor.)</td>
</tr>
</tbody>
</table>
See Also

RecnoCursor Class
RecnoCursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
See Also
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Refresh()()</td>
<td>Store the key/data pair to which the cursor refers in Current. (Inherited from Cursor.)</td>
</tr>
<tr>
<td>Refresh(LockingInfo)</td>
<td>Store the key/data pair to which the cursor refers in Current. (Inherited from Cursor.)</td>
</tr>
</tbody>
</table>
See Also

RecnoCursor Class
RecnoCursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
RecnoCursor Class   See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RefreshMultiple()()()</td>
<td>Store the key/data pair to which the cursor refers and as many duplicate data items that can fit in a buffer the size of one database page in <code>CurrentMultiple</code>. (Inherited from <code>Cursor</code>.)</td>
</tr>
<tr>
<td>RefreshMultiple(LockingInfo)</td>
<td>Store the key/data pair to which the cursor refers and as many duplicate data items that can fit in a buffer the size of one database page in <code>CurrentMultiple</code>. (Inherited from <code>Cursor</code>.)</td>
</tr>
<tr>
<td>RefreshMultiple(Int32)</td>
<td>Store the key/data pair to which the cursor refers and as many duplicate data items that can fit in a buffer the size of BufferSize in <code>CurrentMultiple</code>. (Inherited from <code>Cursor</code>.)</td>
</tr>
<tr>
<td>RefreshMultiple(Int32, LockingInfo)</td>
<td>Store the key/data pair to which the cursor refers and as many duplicate data items that can fit in a buffer the size of BufferSize in <code>CurrentMultiple</code>. (Inherited from <code>Cursor</code>.)</td>
</tr>
</tbody>
</table>
See Also

RecnoCursor Class
RecnoCursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
RecnoCursor.

:.RefreshMultipleKey Method

RecnoCursor Class See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>RefreshMultipleKey()()()</strong></td>
<td>Store the key/data pair to which the cursor refers and as many ensuing key/data pairs that can fit in a buffer the size of one database page in CurrentMultipleKey. (Inherited from Cursor.)</td>
</tr>
<tr>
<td><strong>RefreshMultipleKey(LockingInfo)</strong></td>
<td>Store the key/data pair to which the cursor refers and as many ensuing key/data pairs that can fit in a buffer the size of one database page in CurrentMultipleKey. (Inherited from Cursor.)</td>
</tr>
<tr>
<td><strong>RefreshMultipleKey(Int32)</strong></td>
<td>Store the key/data pair to which the cursor refers and as many ensuing key/data pairs that can fit in a buffer the size of BufferSize in CurrentMultipleKey. (Inherited from Cursor.)</td>
</tr>
<tr>
<td><strong>RefreshMultipleKey(Int32, LockingInfo)</strong></td>
<td>Store the key/data pair to which the cursor refers and as many ensuing key/data pairs that can fit in a buffer the size of BufferSize in CurrentMultipleKey. (Inherited from Cursor.)</td>
</tr>
</tbody>
</table>
See Also

RecnoCursor Class
RecnoCursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `RecnoCursor` type exposes the following members.
# Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current</td>
<td>The key/data pair at which the cursor currently points. (Inherited from Cursor.)</td>
</tr>
<tr>
<td>CurrentMultiple</td>
<td>The key and multiple data items at which the cursor currently points. (Inherited from Cursor.)</td>
</tr>
<tr>
<td>CurrentMultipleKey</td>
<td>The multiple key and data items at which the cursor currently points. (Inherited from Cursor.)</td>
</tr>
<tr>
<td>Priority</td>
<td>The cache priority for pages referenced by the cursor. (Inherited from Cursor.)</td>
</tr>
</tbody>
</table>
See Also

RecnoCursor Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
A class representing a RecnoDatabase. The Recno format supports fixed- or variable-length records, accessed sequentially or by logical record number, and optionally backed by a flat text file.

**Namespace:** BerkeleyDB
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
## Syntax

**C#**

```csharp
public class RecnoDatabase : Database
```

**Visual Basic (Declaration)**

```vbnet
Public Class RecnoDatabase _
    Inherits Database
```

**Visual C++**

```cpp
public ref class RecnoDatabase : public Database
```
Inheritance Hierarchy

System::Object

BerkeleyDB::BaseDatabase

BerkeleyDB::Database

BerkeleyDB::RecnoDatabase
See Also

RecnoDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The **RecnoDatabase** type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Append</td>
<td>Overloaded. Overloaded.</td>
</tr>
</tbody>
</table>

Although closing a database will close any open cursors, it is recommended that applications explicitly close all their Cursor objects before closing the database. The reason why is that when the cursor is explicitly closed, the memory allocated for it is reclaimed; however, this will not happen if you close a database while cursors are still opened.

The same rule, for the same reasons, hold true for Transaction objects. Simply make sure you resolve all your transaction objects before closing your database handle.

Because key/data pairs are cached in memory, applications should make a point to always either close database handles or sync their data to disk (using Sync() before exiting, to ensure that any data cached in main memory are reflected in the underlying file system.

When called on a database that is the primary database for a secondary index, the primary database should be closed only after all secondary indices referencing it have been closed.

When multiple threads are using the object concurrently, only a single thread may call the Close method.
The object may not be accessed again after Close is called, regardless of its outcome.

- **Compact**  Overloaded.
- **Cursor**   Overloaded.
- **Delete**  Overloaded.
- **Dispose** Release the resources held by this object, and close the database if it's still open. (Inherited from **BaseDatabase**.)
- **Equals** Determines whether the specified **Object** is equal to the current **Object**. (Inherited from **Object**.)
- **Exists**  Overloaded.
- **FastStats** Among other things, this method makes it possible for applications to request key and record counts without incurring the performance penalty of traversing the entire database.
- **Get**  Overloaded.
- **GetBoth** Overloaded.
- **GetBothMultiple** Overloaded.
- **GetHashCode** Serves as a hash function for a particular type. (Inherited from **Object**.)
- **GetMultiple** Overloaded.
- **GetType**  Gets the **Type** of the current instance. (Inherited from **Object**.)
- **Join** Create a specialized join cursor for use in performing equality or natural joins on secondary indices. (Inherited from **Database**.)
- **Open**  Overloaded.
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PrintFastStats</strong></td>
<td>Overloaded. The statistical information is described by the BTreeStats, HashStats, QueueStats, and RecnoStats classes.</td>
</tr>
<tr>
<td><strong>PrintStats</strong></td>
<td>Overloaded. The statistical information is described by the BTreeStats, HashStats, QueueStats, and RecnoStats classes.</td>
</tr>
<tr>
<td><strong>Put</strong></td>
<td>Overloaded. If the database supports duplicates, add the new data value at the end of the duplicate set. If the database supports sorted duplicates, the new data value is inserted at the correct sorted location.</td>
</tr>
<tr>
<td><strong>PutNoOverwrite</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>Stats</strong></td>
<td>Overloaded. The statistical information is described by the BTreeStats.</td>
</tr>
<tr>
<td><strong>Sync</strong></td>
<td>Overloaded. Flush any cached information to disk. (Inherited from BaseDatabase.) Returns a String that represents the current Object.</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Overloaded. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>Truncate</strong></td>
<td>Overloaded. When called on a database configured with secondary indices, Truncate will truncate the primary database and all secondary indices. A count of the records discarded from the primary database is returned.</td>
</tr>
<tr>
<td><strong>TruncateUnusedPages</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>AppendCallback</td>
<td>A function to call after the record number has been selected but before the data has been stored into the database.</td>
</tr>
<tr>
<td>AutoCommit</td>
<td>If true, all database modification operations based on this object will be transactionally protected.</td>
</tr>
<tr>
<td>CacheSize</td>
<td>The size of the shared memory buffer pool -- that is, the cache.</td>
</tr>
<tr>
<td>Creation</td>
<td>The CreatePolicy with which this database was opened.</td>
</tr>
<tr>
<td>DatabaseName</td>
<td>The name of this database, if it has one.</td>
</tr>
<tr>
<td>DoChecksum</td>
<td>If true, do checksum verification of pages read into the cache from the backing filestore.</td>
</tr>
<tr>
<td>EncryptAlgorithm</td>
<td>The algorithm used by the Berkeley DB library to perform encryption and decryption.</td>
</tr>
<tr>
<td>Encrypted</td>
<td>If true, encrypt all data stored in the database.</td>
</tr>
<tr>
<td>Endianness</td>
<td>The database byte order.</td>
</tr>
<tr>
<td>ErrorFeedback</td>
<td>The mechanism for reporting detailed error messages to the application.</td>
</tr>
<tr>
<td>ErrorPrefix</td>
<td>The prefix string that appears before error messages issued by Berkeley DB.</td>
</tr>
</tbody>
</table>

Monitor progress within long running operations.
<table>
<thead>
<tr>
<th><strong>Feedback</strong></th>
<th>(Inherited from <a href="#">BaseDatabase</a>.)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>FileName</strong></td>
<td>The filename of this database, if it has one. (Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
<tr>
<td><strong>FreeThreaded</strong></td>
<td>If true, the object is free-threaded; that is, concurrently usable by multiple threads in the address space. (Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
<tr>
<td><strong>HasMultiple</strong></td>
<td>If true, the object references a physical file supporting multiple databases. (Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
<tr>
<td><strong>InHostOrder</strong></td>
<td>If true, the underlying database files were created on an architecture of the same byte order as the current one. This information may be used to determine whether application data needs to be adjusted for this architecture or not. (Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
<tr>
<td><strong>NoMMap</strong></td>
<td>If true, this database is not mapped into process memory.</td>
</tr>
<tr>
<td><strong>NonDurableTxns</strong></td>
<td>See <a href="#">MMapSize</a> for further information. (Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
<tr>
<td><strong>Pagesize</strong></td>
<td>If true, Berkeley DB will not write log records for this database. (Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
<tr>
<td><strong>Priority</strong></td>
<td>The database's current page size. (Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
<tr>
<td><strong>ReadOnly</strong></td>
<td>The cache priority for pages referenced by this object. (Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
<tr>
<td><strong>ReadOnly</strong></td>
<td>If true, this database has been opened for reading only. Any attempt to modify items in the database will fail, regardless of the actual permissions of any underlying files. (Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
<tr>
<td><strong>Pagesize</strong></td>
<td>If true, this database supports transactional read</td>
</tr>
<tr>
<td><strong>ReadUncommitted</strong></td>
<td>operations with degree 1 isolation. Read operations on the database may request the return of modified but not yet committed data. (Inherited from <code>BaseDatabase</code>.)</td>
</tr>
<tr>
<td><strong>RecordDelimiter</strong></td>
<td>The delimiting byte used to mark the end of a record in <code>SourceFile</code>.</td>
</tr>
<tr>
<td><strong>RecordLength</strong></td>
<td>If using fixed-length, not byte-delimited records, the length of the records.</td>
</tr>
<tr>
<td><strong>RecordPad</strong></td>
<td>The padding character for short, fixed-length records.</td>
</tr>
<tr>
<td><strong>Renumber</strong></td>
<td>If true, the logical record numbers are mutable, and change as records are added to and deleted from the database.</td>
</tr>
<tr>
<td><strong>Snapshot</strong></td>
<td>If true, any <code>SourceFile</code> file will be read in its entirety when <code>Open(String, RecnoDatabaseConfig)</code> is called. If false, <code>SourceFile</code> may be read lazily.</td>
</tr>
<tr>
<td><strong>SourceFile</strong></td>
<td>The underlying source file for the Recno access method.</td>
</tr>
<tr>
<td><strong>Transactional</strong></td>
<td>If true, this database has been opened in a transactional mode. (Inherited from <code>BaseDatabase</code>.)</td>
</tr>
<tr>
<td><strong>Truncated</strong></td>
<td>If true, the underlying file was physically truncated upon open, discarding all previous databases it might have held. (Inherited from <code>BaseDatabase</code>.)</td>
</tr>
<tr>
<td><strong>Type</strong></td>
<td>The type of the underlying access method (and file format). This value may be used to determine the type of the database after an <code>Open(String, DatabaseConfig)</code>. (Inherited from <code>BaseDatabase</code>.)</td>
</tr>
<tr>
<td><strong>UseMVCC</strong></td>
<td>If true, the database was opened with support for multiversion concurrency control. (Inherited from <code>BaseDatabase</code>.)</td>
</tr>
</tbody>
</table>
See Also

RecnoDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `RecnoDatabase` type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| Append | Overloaded. Overloaded. 

Although closing a database will close any open cursors, it is recommended that applications explicitly close all their Cursor objects before closing the database. The reason why is that when the cursor is explicitly closed, the memory allocated for it is reclaimed; however, this will not happen if you close a database while cursors are still opened.

The same rule, for the same reasons, hold true for Transaction objects. Simply make sure you resolve all your transaction objects before closing your database handle.

Because key/data pairs are cached in memory, applications should make a point to always either close database handles or sync their data to disk (using Sync()()) before exiting, to ensure that any data cached in main memory are reflected in the underlying file system.

When called on a database that is the primary database for a secondary index, the primary database should be closed only after all secondary indices referencing it have been closed.

When multiple threads are using the object concurrently, only a single thread may call the Close method.
The object may not be accessed again after Close is called, regardless of its outcome.

- **Compact** Overloaded.
- **Cursor** Overloaded.
- **Delete** Overloaded.
- **Dispose**
  Release the resources held by this object, and close the database if it's still open.
  (Inherited from [BaseDatabase](#)).
- **Equals**
  Determines whether the specified [Object](#) is equal to the current [Object](#).
  (Inherited from [Object](#)).
- **Exists** Overloaded.

Among other things, this method makes it possible for applications to request key and record counts without incurring the performance penalty of traversing the entire database.

- **FastStats**
  The statistical information is described by the [BTreeStats](#), [HashStats](#), [QueueStats](#), and [RecnoStats](#) classes.

- **Get** Overloaded.
- **GetBoth** Overloaded.
- **GetBothMultiple** Overloaded.
- **GetHashCode**
  Serves as a hash function for a particular type.
  (Inherited from [Object](#)).
- **GetMultiple** Overloaded.
- **GetType**
  Gets the [Type](#) of the current instance.
  (Inherited from [Object](#)).
- **Join**
  Create a specialized join cursor for use in performing equality or natural joins on secondary indices.
  (Inherited from [Database](#)).
- **Open** Overloaded.
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>PrintFastStats</code></td>
<td>Overloaded. The statistical information is described by the <code>BTreeStats</code>, <code>HashStats</code>, <code>QueueStats</code>, and <code>RecnoStats</code> classes.</td>
</tr>
<tr>
<td><code>PrintStats</code></td>
<td>Overloaded. The statistical information is described by the <code>BTreeStats</code>, <code>HashStats</code>, <code>QueueStats</code>, and <code>RecnoStats</code> classes.</td>
</tr>
<tr>
<td><code>Put</code></td>
<td>If the database supports duplicates, add the new data value at the end of the duplicate set. If the database supports sorted duplicates, the new data value is inserted at the correct sorted location.</td>
</tr>
<tr>
<td><code>PutNoOverwrite</code></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><code>Stats</code></td>
<td>Overloaded. The statistical information is described by <code>BTreeStats</code>.</td>
</tr>
<tr>
<td><code>Sync</code></td>
<td>Flush any cached information to disk. (Inherited from <code>BaseDatabase</code>.) Returns a <code>String</code> that represents the current</td>
</tr>
<tr>
<td><code>ToString</code></td>
<td>(Inherited from <code>Object</code>.) Overloaded.</td>
</tr>
<tr>
<td><code>Truncate</code></td>
<td>When called on a database configured with secondary indices, Truncate will truncate the primary database and all secondary indices. A count of the records discarded from the primary database is returned.</td>
</tr>
<tr>
<td><code>TruncateUnusedPages</code></td>
<td>Overloaded.</td>
</tr>
</tbody>
</table>
See Also

RecnoDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
# Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Append(DatabaseEntry)</code></td>
<td>Append the data item to the end of the database.</td>
</tr>
<tr>
<td><code>Append(DatabaseEntry, Transaction)</code></td>
<td>Append the data item to the end of the database.</td>
</tr>
</tbody>
</table>
See Also

RecnoDatabase Class
RecnoDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Append the data item to the end of the database.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

```csharp
public uint Append(
    DatabaseEntry data)
```

Visual Basic (Declaration)

```vbnet
Public Function Append ( _
    data As DatabaseEntry _
) AsUInteger
```

Visual C++

```cpp
public:
    unsigned int Append(
        DatabaseEntry^ data
    )
```

Parameters

data
   Type: BerkeleyDB::DatabaseEntry
   The data item to store in the database

Return Value

The record number allocated to the record
See Also

RecnoDatabase Class
Append Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Append the data item to the end of the database.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
### Syntax

#### C#

```csharp
public uint Append(
    DatabaseEntry data,
    Transaction txn
)
```

#### Visual Basic (Declaration)

```vbnet
Public Function Append (_
    data As DatabaseEntry, _
    txn As Transaction _
) As UInteger
```

#### Visual C++

```cpp
public:
    unsigned int Append(
        DatabaseEntry^ data,
        Transaction^ txn
    )
```

### Parameters

**data**

Type: `BerkeleyDB::DatabaseEntry`

The data item to store in the database

**txn**

Type: `BerkeleyDB::Transaction`

If the operation is part of an application-specified transaction, `txn` is a Transaction object returned from `BeginTransaction()`; if the operation is part of a Berkeley DB Concurrent Data Store group, `txn` is a handle returned from `BeginCDSGroup()`; otherwise null.

### Return Value
The record number allocated to the record
Remarks

There is a minor behavioral difference between Append(DatabaseEntry) and Append(DatabaseEntry). If a transaction enclosing an Append operation aborts, the record number may be reallocated in a subsequent Append(DatabaseEntry) operation, but it will not be reallocated in a subsequent Append(DatabaseEntry) operation.
See Also

RecnoDatabase Class
Append Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Although closing a database will close any open cursors, it is recommended that applications explicitly close all their Cursor objects before closing the database. The reason why is that when the cursor is explicitly closed, the memory allocated for it is reclaimed; however, this will not happen if you close a database while cursors are still opened.

The same rule, for the same reasons, hold true for Transaction objects. Simply make sure you resolve all your transaction objects before closing your database handle.

Because key/data pairs are cached in memory, applications should make a point to always either close database handles or sync their data to disk (using Sync() before exiting, to ensure that any data cached in main memory are reflected in the underlying file system.

When called on a database that is the primary database for a secondary index, the primary database should be closed only after all secondary indices referencing it have been closed.

When multiple threads are using the object concurrently, only a single thread may call the Close method.

The object may not be accessed again after Close is called, regardless of its outcome.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Close()()</td>
<td>Flush any cached database information to disk, close any open Cursor()() objects, free any allocated resources, and close any underlying files. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>Close(Boolean)</td>
<td>Optionally flush any cached database information to disk, close any open Cursor()() objects, free any allocated resources, and close any underlying files. (Inherited from BaseDatabase.)</td>
</tr>
</tbody>
</table>
See Also

RecnoDatabase Class
RecnoDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation

RecnoDatabase...:::Compact Method

RecnoDatabase Class  See Also
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compact(CompactConfig)</td>
<td>Compact the database, and optionally return unused database pages to the underlying filesystem.</td>
</tr>
<tr>
<td>Compact(CompactConfig, Transaction)</td>
<td>Compact the database, and optionally return unused database pages to the underlying filesystem.</td>
</tr>
</tbody>
</table>
See Also

RecnoDatabase Class
RecnoDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Compact the database, and optionally return unused database pages to the underlying filesystem.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public CompactData Compact(
    CompactConfig cdata
)

Visual Basic (Declaration)

Public Function Compact ( _
    cdata As CompactConfig _
) As CompactData

Visual C++

public:
    CompactData^ Compact(
        CompactConfig^ cdata
    )

Parameters

cdata
    Type: BerkeleyDB:::CompactConfig
    Compact configuration parameters

Return Value

Compact operation statistics
Remarks

If the operation occurs in a transactional database, the operation will be implicitly transaction protected using multiple transactions. These transactions will be periodically committed to avoid locking large sections of the tree. Any deadlocks encountered cause the compaction operation to be retried from the point of the last transaction commit.
See Also

RecnoDatabase Class
Compact Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
RecnoDatabase..::..Compact Method (CompactConfig, Transaction)

Compact the database, and optionally return unused database pages to the underlying filesystem.

Namespace: BerkeleyDB
Syntax

C#

```csharp
public CompactData Compact(
    CompactConfig cdata,
    Transaction txn
)
```

Visual Basic (Declaration)

```vbnet
Public Function Compact (
    cdata As CompactConfig,
    txn As Transaction
) As CompactData
```

Visual C++

```cpp
public:
CompactData^ Compact(
    CompactConfig^ cdata,
    Transaction^ txn
)
```

Parameters

cdata

Type: BerkeleyDB:::CompactConfig
Compact configuration parameters

txn

Type: BerkeleyDB:::Transaction
If the operation is part of an application-specified transaction, txn is a Transaction object returned from BeginTransaction(); if the operation is part of a Berkeley DB Concurrent Data Store group, txn is a handle returned from BeginCDSGroup(); otherwise null.

Return Value
Compact operation statistics
Remarks

If txn is non-null, then the operation is performed using that transaction. In this event, large sections of the tree may be locked during the course of the transaction.

If txn is null, but the operation occurs in a transactional database, the operation will be implicitly transaction protected using multiple transactions. These transactions will be periodically committed to avoid locking large sections of the tree. Any deadlocks encountered cause the compaction operation to be retried from the point of the last transaction commit.
See Also

RecnoDatabase Class
Compact Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation
RecnoDatabase::Cursor Method

See Also

RecnoDatabase Class
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cursor()()</td>
<td>Create a database cursor.</td>
</tr>
<tr>
<td>Cursor(CursorConfig)</td>
<td>Create a database cursor with the given configuration.</td>
</tr>
<tr>
<td>Cursor(Transaction)</td>
<td>Create a transactionally protected database cursor.</td>
</tr>
<tr>
<td>Cursor(CursorConfig, Transaction)</td>
<td>Create a transactionally protected database cursor with the given configuration.</td>
</tr>
</tbody>
</table>
See Also

RecnoDatabase Class
RecnoDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation
RecnoDatabase..::.Cursor Method

RecnoDatabase Class  See Also

Create a database cursor.

Namespace:  BerkeleyDB
Syntax

**C#**

```csharp
public RecnoCursor Cursor()
```

**Visual Basic (Declaration)**

```vbnet
Public Function Cursor As RecnoCursor
```

**Visual C++**

```c++
public:
RecnoCursor^ Cursor()
```

**Return Value**

A newly created cursor
See Also

RecnoDatabase Class
Cursor Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Create a database cursor with the given configuration.

**Namespace:** BerkeleyDB
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
## Syntax

### C#

```csharp
public RecnoCursor Cursor(CursorConfig cfg)
```

### Visual Basic (Declaration)

```vbnet
Public Function Cursor(_
    cfg As CursorConfig _
) As RecnoCursor
```

### Visual C++

```cpp
public:
RecnoCursor^ Cursor(
    CursorConfig^ cfg
)
```

## Parameters

### `cfg`

- **Type**: BerkeleyDB::CursorConfig
- The configuration properties for the cursor.

## Return Value

A newly created cursor
See Also

RecnoDatabase Class
Cursor Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Create a transactionally protected database cursor.

**Namespace:** BerkeleyDB
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public RecnoCursor Cursor(
    Transaction txn
)

Visual Basic (Declaration)

Public Function Cursor ( _
    txn As Transaction _
) As RecnoCursor

Visual C++

public:
RecnoCursor^ Cursor(
    Transaction^ txn
)

Parameters

txn
    Type: BerkeleyDB::::Transaction
    The transaction context in which the cursor may be used.

Return Value

A newly created cursor
See Also

RecnoDatabase Class
Cursor Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Create a transactionally protected database cursor with the given configuration.

**Namespace:**  BerkeleyDB  
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
### Syntax

**C#**

```csharp
public RecnoCursor Cursor(
    CursorConfig cfg,
    Transaction txn
)
```

**Visual Basic (Declaration)**

```vbnet
Public Function Cursor ( _
    cfg As CursorConfig, _
    txn As Transaction _
) As RecnoCursor
```

**Visual C++**

```cpp
public:
RecnoCursor^ Cursor(
    CursorConfig^ cfg,
    Transaction^ txn
)
```

### Parameters

**cfg**

Type: `BerkeleyDB::{:}CursorConfig`

The configuration properties for the cursor.

**txn**

Type: `BerkeleyDB::{:}Transaction`

The transaction context in which the cursor may be used.

### Return Value

A newly created cursor
See Also

RecnoDatabase Class
Cursor Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delete(DatabaseEntry)</td>
<td>Remove key/data pairs from the database. The key/data pair associated with key is discarded from the database. In the presence of duplicate key values, all records associated with the designated key will be discarded. (Inherited from BaseDatabase.)</td>
</tr>
</tbody>
</table>
See Also

RecnoDatabase Class
RecnoDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
RecnoDatabase Exists Method

RecnoDatabase Class  See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exists(DatabaseEntry)</td>
<td>Check whether key appears in the database. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>Exists(DatabaseEntry, Transaction)</td>
<td>Check whether key appears in the database. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>Exists(DatabaseEntry, Transaction, LockingInfo)</td>
<td>Check whether key appears in the database. (Inherited from BaseDatabase.)</td>
</tr>
</tbody>
</table>
See Also

RecnoDatabase Class
RecnoDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Among other things, this method makes it possible for applications to request key and record counts without incurring the performance penalty of traversing the entire database.

The statistical information is described by the BTreeStats, HashStats, QueueStats, and RecnoStats classes.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FastStats()()()</td>
<td>Return the database statistical information which does not require traversal of the database.</td>
</tr>
<tr>
<td>FastStats(Transaction)</td>
<td>Return the database statistical information which does not require traversal of the database.</td>
</tr>
<tr>
<td>FastStats(Transaction, Isolation)</td>
<td>Return the database statistical information which does not require traversal of the database.</td>
</tr>
</tbody>
</table>
See Also

RecnoDatabase Class
RecnoDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
RecnoDatabase..::.FastStats Method

RecnoDatabase Class  See Also

Return the database statistical information which does not require traversal of the database.

Namespace:  BerkeleyDB
### Syntax

**C#**

public `RecnoStats` FastStats()

**Visual Basic (Declaration)**

Public Function FastStats As `RecnoStats`

**Visual C++**

public:

`RecnoStats` FastStats()

### Return Value

The database statistical information which does not require traversal of the database.
See Also

RecnoDatabase Class
FastStats Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Return the database statistical information which does not require traversal of the database.

Namespace: BerkeleyDB
Syntax

C#

```csharp
public RecnoStats FastStats(
    Transaction txn
)
```

Visual Basic (Declaration)

```vbnet
Public Function FastStats ( _
    txn As Transaction _
) As RecnoStats
```

Visual C++

```cpp
public:
    RecnoStats^ FastStats(
        Transaction^ txn
    )
```

Parameters

txn

Type: BerkeleyDB::Transaction
If the operation is part of an application-specified transaction, txn is a Transaction object returned from BeginTransaction(); if the operation is part of a Berkeley DB Concurrent Data Store group, txn is a handle returned from BeginCDSGroup(); otherwise null.

Return Value

The database statistical information which does not require traversal of the database.
See Also

RecnoDatabase Class
FastStats Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
RecnoDatabase..:::FastStats Method (Transaction, Isolation)

Return the database statistical information which does not require traversal of the database.

Namespace: BerkeleyDB
Syntax

**C#**

```
public RecnoStats FastStats(
    Transaction txn,
    Isolation isoDegree
)
```

**Visual Basic (Declaration)**

```
Public Function FastStats (_
    txn As Transaction, _
    isoDegree As Isolation _
) As RecnoStats
```

**Visual C++**

```
public:
RecnoStats^ FastStats(
    Transaction^ txn,
    Isolation isoDegree
)
```

**Parameters**

- **txn**
  Type: *BerkeleyDB...:::Transaction*
  If the operation is part of an application-specified transaction, txn is a Transaction object returned from `BeginTransaction()`; if the operation is part of a Berkeley DB Concurrent Data Store group, txn is a handle returned from `BeginCDSGroup()`; otherwise null.

- **isoDegree**
  Type: *BerkeleyDB...:::Isolation*
  The level of isolation for database reads. **DEGREE_ONE** will be silently ignored for databases which did not specify `ReadUncommitted`. 
Return Value

The database statistical information which does not require traversal of the database.
See Also

RecnoDatabase Class
FastStats Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
RecnoDatabase Class  See Also
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Get(DatabaseEntry)</td>
<td>Retrieve a key/data pair from the database. In the presence of duplicate key values, Get will return the first data item for key. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>Get(DatabaseEntry, Transaction)</td>
<td>Retrieve a key/data pair from the database. In the presence of duplicate key values, Get will return the first data item for key. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>Get(DatabaseEntry, Transaction, LockingInfo)</td>
<td>Retrieve a key/data pair from the database. In the presence of duplicate key values, Get will return the first data item for key. (Inherited from BaseDatabase.)</td>
</tr>
</tbody>
</table>
See Also

RecnoDatabase Class
RecnoDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
RecnoDatabase::GetBoth Method

RecnoDatabase Class   See Also
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetBoth(DatabaseEntry, DatabaseEntry)</td>
<td>Retrieve a key/data pair from the database which matches key and data. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>GetBoth(DatabaseEntry, DatabaseEntry, Transaction)</td>
<td>Retrieve a key/data pair from the database which matches key and data. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>GetBoth(DatabaseEntry, DatabaseEntry, Transaction, LockingInfo)</td>
<td>Retrieve a key/data pair from the database which matches key and data. (Inherited from BaseDatabase.)</td>
</tr>
</tbody>
</table>
See Also

RecnoDatabase Class
RecnoDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
RecnoDatabase Class  See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetBothMultiple(DatabaseEntry, DatabaseEntry)</td>
<td>If a key/data pair in the database matches key and data, return the key and all duplicate data items. (Inherited from Database.)</td>
</tr>
<tr>
<td>GetBothMultiple(DatabaseEntry, DatabaseEntry, Int32)</td>
<td>If a key/data pair in the database matches key and data, return the key and all duplicate data items. (Inherited from Database.)</td>
</tr>
<tr>
<td>GetBothMultiple(DatabaseEntry, DatabaseEntry, Int32, Transaction)</td>
<td>If a key/data pair in the database matches key and data, return the key and all duplicate data items. (Inherited from Database.)</td>
</tr>
<tr>
<td>GetBothMultiple(DatabaseEntry, DatabaseEntry, Int32, Transaction, LockingInfo)</td>
<td>If a key/data pair in the database matches key and data, return the key and all duplicate data items. (Inherited from Database.)</td>
</tr>
</tbody>
</table>
See Also

RecnoDatabase Class
RecnoDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
RecnoDatabase..::.GetMultiple Method

RecnoDatabase Class  See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>GetMultiple(DatabaseEntry)</code></td>
<td>Retrieve a key and all duplicate data items from the database.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Database</a>.)</td>
</tr>
<tr>
<td><code>GetMultiple(DatabaseEntry, Int32)</code></td>
<td>Retrieve a key and all duplicate data items from the database.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Database</a>.)</td>
</tr>
<tr>
<td><code>GetMultiple(DatabaseEntry, Int32, Transaction)</code></td>
<td>Retrieve a key and all duplicate data items from the database.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Database</a>.)</td>
</tr>
<tr>
<td><code>GetMultiple(DatabaseEntry, Int32, Transaction, LockingInfo)</code></td>
<td>Retrieve a key and all duplicate data items from the database.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">Database</a>.)</td>
</tr>
</tbody>
</table>
See Also

RecnoDatabase Class
RecnoDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation
RecnoDatabase...::Open Method
RecnoDatabase Class  See Also
# Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open(String, RecnoDatabaseConfig)</td>
<td>Instantiate a new RecnoDatabase object and open the database represented by Filename.</td>
</tr>
<tr>
<td>Open(String, RecnoDatabaseConfig, Transaction)</td>
<td>Instantiate a new RecnoDatabase object and open the database represented by Filename.</td>
</tr>
<tr>
<td>Open(String, String, RecnoDatabaseConfig)</td>
<td>Instantiate a new RecnoDatabase object and open the database represented by Filename and DatabaseName.</td>
</tr>
<tr>
<td>Open(String, String, RecnoDatabaseConfig, Transaction)</td>
<td>Instantiate a new RecnoDatabase object and open the database represented by Filename and DatabaseName.</td>
</tr>
</tbody>
</table>
See Also

RecnoDatabase Class
RecnoDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Instantiate a new RecnoDatabase object and open the database represented by Filename.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public static RecnoDatabase Open(
    string Filename,
    RecnoDatabaseConfig cfg
)

Visual Basic (Declaration)

Public Shared Function Open ( _
    Filename As String, _
    cfg As RecnoDatabaseConfig _
) As RecnoDatabase

Visual C++

public:
    static RecnoDatabase^ Open(
        String^ Filename,
        RecnoDatabaseConfig^ cfg
    )

Parameters

Filename
Type: System::String
The name of an underlying file that will be used to back the database. In-memory databases never intended to be preserved on disk may be created by setting this parameter to null.

cfg
Type: BerkeleyDB::RecnoDatabaseConfig
The database's configuration

Return Value
A new, open database object
Remarks

If Filename is null, the database is strictly temporary and cannot be opened by any other thread of control, thus the database can only be accessed by sharing the single database object that created it, in circumstances where doing so is safe.

If AutoCommit is set, the operation will be implicitly transaction protected. Note that transactionally protected operations on a database object requires the object itself be transactionally protected during its open.
See Also

RecnoDatabase Class
Open Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
C#  Visual Basic  Visual C++

Berkeley DB .NET API Documentation

RecnoDatabase..:::.Open Method (String, RecnoDatabaseConfig, Transaction)

RecnoDatabase Class   See Also

Instantiate a new RecnoDatabase object and open the database represented by Filename.

Namespace:  BerkeleyDB
## Syntax

### C#

```csharp
public static RecnoDatabase Open(
    string Filename,
    RecnoDatabaseConfig cfg,
    Transaction txn
)
```

### Visual Basic (Declaration)

```vbnet
Public Shared Function Open ( _
    Filename As String,
    cfg As RecnoDatabaseConfig, _
    txn As Transaction _
) As RecnoDatabase
```

### Visual C++

```cpp
public:
static RecnoDatabase^ Open(
    String^ Filename,
    RecnoDatabaseConfig^ cfg,
    Transaction^ txn
)
```

## Parameters

### Filename

- **Type:** `System::String`
- The name of an underlying file that will be used to back the database. In-memory databases never intended to be preserved on disk may be created by setting this parameter to null.

### cfg

- **Type:** `BerkeleyDB::RecnoDatabaseConfig`
- The database's configuration
txn

Type: BerkeleyDB::::Transaction
If the operation is part of an application-specified transaction, txn is a
Transaction object returned from BeginTransaction(); if the operation is
part of a Berkeley DB Concurrent Data Store group, txn is a handle
returned from BeginCDSGroup(); otherwise null.

Return Value

A new, open database object
Remarks

If Filename is null, the database is strictly temporary and cannot be opened by any other thread of control, thus the database can only be accessed by sharing the single database object that created it, in circumstances where doing so is safe.

If txn is null, but AutoCommit is set, the operation will be implicitly transaction protected. Note that transactionally protected operations on a database object requires the object itself be transactionally protected during its open. Also note that the transaction must be committed before the object is closed.
See Also

RecnoDatabase Class
Open Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
RecnoDatabase..:::Open Method (String, String, RecnoDatabaseConfig)

RecnoDatabase Class  See Also

Instantiate a new RecnoDatabase object and open the database represented by Filename and DatabaseName.

Namespace:  BerkeleyDB
Syntax

C#

```csharp
public static RecnoDatabase Open(
    string Filename,
    string DatabaseName,
    RecnoDatabaseConfig cfg
)
```

**Visual Basic (Declaration)**

```vbnet
Public Shared Function Open ( _
    Filename As String, _
    DatabaseName As String, _
    cfg As RecnoDatabaseConfig _
) As RecnoDatabase
```

**Visual C++**

```cpp
public:
static RecnoDatabase^ Open(
    String^ Filename, _
    String^ DatabaseName, _
    RecnoDatabaseConfig^ cfg
)
```

**Parameters**

**Filename**

Type: `System::::String`

The name of an underlying file that will be used to back the database. In-memory databases never intended to be preserved on disk may be created by setting this parameter to null.

**DatabaseName**

Type: `System::::String`

This parameter allows applications to have multiple databases in a single file. Although no DatabaseName needs to be specified, it is an error to
attempt to open a second database in a file that was not initially created using a database name.

cfg
  Type: BerkeleyDB::RecnoDatabaseConfig
  The database's configuration

**Return Value**

A new, open database object
Remarks

If both Filename and DatabaseName are null, the database is strictly temporary and cannot be opened by any other thread of control, thus the database can only be accessed by sharing the single database object that created it, in circumstances where doing so is safe. If Filename is null and DatabaseName is non-null, the database can be opened by other threads of control and will be replicated to client sites in any replication group.

If AutoCommit is set, the operation will be implicitly transaction protected. Note that transactionally protected operations on a database object requires the object itself be transactionally protected during its open.
See Also

RecnoDatabase Class
Open Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Instantiate a new RecnoDatabase object and open the database represented by Filename and DatabaseName.

**Namespace:**  BerkeleyDB  
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
### Syntax

**C#**

```csharp
public static RecnoDatabase Open(
    string Filename,
    string DatabaseName,
    RecnoDatabaseConfig cfg,
    Transaction txn)
```

**Visual Basic (Declaration)**

```vbnet
Public Shared Function Open (_
    Filename As String, _
    DatabaseName As String, _
    cfg As RecnoDatabaseConfig, _
    txn As Transaction _
) As RecnoDatabase
```

**Visual C++**

```csharp
public:
static RecnoDatabase^ Open(
    String^ Filename,
    String^ DatabaseName,
    RecnoDatabaseConfig^ cfg,
    Transaction^ txn
)
```

### Parameters

**Filename**

Type: `System::::String`

The name of an underlying file that will be used to back the database. In-memory databases never intended to be preserved on disk may be created by setting this parameter to null.

**DatabaseName**

Type: `System::::String`
This parameter allows applications to have multiple databases in a single file. Although no DatabaseName needs to be specified, it is an error to attempt to open a second database in a file that was not initially created using a database name.

**cfg**

Type: BerkeleyDB::RecnoDatabaseConfig

The database's configuration

**txn**

Type: BerkeleyDB::Transaction

If the operation is part of an application-specified transaction, txn is a Transaction object returned from `BeginTransaction()()`; if the operation is part of a Berkeley DB Concurrent Data Store group, txn is a handle returned from `BeginCDSGroup();` otherwise null.

**Return Value**

A new, open database object
Remarks

If both Filename and DatabaseName are null, the database is strictly temporary and cannot be opened by any other thread of control, thus the database can only be accessed by sharing the single database object that created it, in circumstances where doing so is safe. If Filename is null and DatabaseName is non-null, the database can be opened by other threads of control and will be replicated to client sites in any replication group.

If txn is null, but AutoCommit is set, the operation will be implicitly transaction protected. Note that transactionally protected operations on a database object requires the object itself be transactionally protected during its open. Also note that the transaction must be committed before the object is closed.
See Also

RecnoDatabase Class
Open Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
RecnoDatabase Class  See Also

The statistical information is described by the **BTreeStats**, **HashStats**, **QueueStats**, and **RecnoStats** classes.
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>PrintFastStats()</code></td>
<td>Display the database statistical information which does not require traversal of the database. (Inherited from <code>BaseDatabase</code>.)</td>
</tr>
<tr>
<td><code>PrintFastStats(Boolean)</code></td>
<td>Display the database statistical information which does not require traversal of the database. (Inherited from <code>BaseDatabase</code>.)</td>
</tr>
</tbody>
</table>
See Also

RecnoDatabase Class
RecnoDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
RecnoDatabase:::PrintStats Method

The statistical information is described by the BTreeStats, HashStats, QueueStats, and RecnoStats classes.
### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>PrintStats()</code></td>
<td>Display the database statistical information. (Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
<tr>
<td><code>PrintStats(Boolean)</code></td>
<td>Display the database statistical information. (Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
</tbody>
</table>
See Also

RecnoDatabase Class
RecnoDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If the database supports duplicates, add the new data value at the end of the duplicate set. If the database supports sorted duplicates, the new data value is inserted at the correct sorted location.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Put(DatabaseEntry, DatabaseEntry)</code></td>
<td>Store the key/data pair in the database, replacing any previously existing key if duplicates are disallowed, or adding a duplicate data item if duplicates are allowed. (Inherited from <a href="#">Database</a>).</td>
</tr>
<tr>
<td><code>Put(DatabaseEntry, DatabaseEntry, Transaction)</code></td>
<td>Store the key/data pair in the database, replacing any previously existing key if duplicates are disallowed, or adding a duplicate data item if duplicates are allowed. (Inherited from <a href="#">Database</a>).</td>
</tr>
</tbody>
</table>
See Also

RecnoDatabase Class
RecnoDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
RecnoDatabase Class  See Also
### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>PutNoOverwrite(DatabaseEntry, DatabaseEntry)</code></td>
<td>Store the key/data pair in the database, only if the key does not already appear in the database. (Inherited from <code>Database</code>.)</td>
</tr>
<tr>
<td><code>PutNoOverwrite(DatabaseEntry, DatabaseEntry, Transaction)</code></td>
<td>Store the key/data pair in the database, only if the key does not already appear in the database. (Inherited from <code>Database</code>.)</td>
</tr>
</tbody>
</table>
See Also

RecnoDatabase Class
RecnoDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The statistical information is described by BTreeStats.
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stats()()()</td>
<td>Return the database statistical information for this database.</td>
</tr>
<tr>
<td>Stats(Transaction)</td>
<td>Return the database statistical information for this database.</td>
</tr>
<tr>
<td>Stats(Transaction, Isolation)</td>
<td>Return the database statistical information for this database.</td>
</tr>
</tbody>
</table>
See Also

RecnoDatabase Class
RecnoDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Return the database statistical information for this database.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public RecnoStats Stats()

Visual Basic (Declaration)

Public Function Stats As RecnoStats

Visual C++

public:
RecnoStats^ Stats()

Return Value

Database statistical information.
See Also

RecnoDatabase Class
Stats Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Return the database statistical information for this database.

**Namespace:**  [BerkeleyDB](https://www.oracle.com/database/berkeley-db/index.html)

**Assembly:**  `libdb_dotnet48` (in `libdb_dotnet48.dll`) Version: 4.8.24.0
Syntax

C#

```csharp
public RecnoStats Stats(
    Transaction txn
)
```

Visual Basic (Declaration)

```vbnet
Public Function Stats ( _
    txn As Transaction _
) As RecnoStats
```

Visual C++

```cpp
public:
    RecnoStats^ Stats(
        Transaction^ txn
    )
```

Parameters

txn

Type: BerkeleyDB::<Transaction>

If the operation is part of an application-specified transaction, txn is a Transaction object returned from BeginTransaction(); if the operation is part of a Berkeley DB Concurrent Data Store group, txn is a handle returned from BeginCDSGroup(); otherwise null.

Return Value

Database statistical information.
See Also

RecnoDatabase Class
Stats Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Return the database statistical information for this database.

**Namespace:**  BerkeleyDB  
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
### Syntax

**C#**

```csharp
public RecnoStats Stats(
    Transaction txn,
    Isolation isoDegree
)
```

**Visual Basic (Declaration)**

```vbnet
Public Function Stats ( _
    txn As Transaction, _
    isoDegree As Isolation _
) As RecnoStats
```

**Visual C++**

```cpp
public:
RecnoStats^ Stats(
    Transaction^ txn,
    Isolation isoDegree
)
```

### Parameters

**txn**

- **Type:** `BerkeleyDB::::Transaction`
- If the operation is part of an application-specified transaction, `txn` is a `Transaction` object returned from `BeginTransaction()()();` if the operation is part of a Berkeley DB Concurrent Data Store group, `txn` is a handle returned from `BeginCDSGroup()()();` otherwise null.

**isoDegree**

- **Type:** `BerkeleyDB::::Isolation`
- The level of isolation for database reads. `DEGREE_ONE` will be silently ignored for databases which did not specify `ReadUncommitted`.
Return Value

Database statistical information.
See Also

RecnoDatabase Class
Stats Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
RecnoDatabase:::Truncate Method

When called on a database configured with secondary indices, Truncate will truncate the primary database and all secondary indices. A count of the records discarded from the primary database is returned.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Truncate()()</td>
<td>Empty the database, discarding all records it contains.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
<tr>
<td>Truncate(Transaction)</td>
<td>Empty the database, discarding all records it contains.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
</tbody>
</table>
See Also

RecnoDatabase Class
RecnoDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
RecnoDatabase Class  See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>TruncateUnusedPages()</strong></td>
<td>Return pages to the filesystem that are already free and at the end of the file.</td>
</tr>
<tr>
<td><strong>TruncateUnusedPages(Transaction)</strong></td>
<td>Return pages to the filesystem that are already free and at the end of the file.</td>
</tr>
</tbody>
</table>
See Also

RecnoDatabase Class
RecnoDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
RecnoDatabase..:::TruncateUnusedPages Method

Return pages to the filesystem that are already free and at the end of the file.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

class TruncateUnusedPages;

public uint TruncateUnusedPages()

Visual Basic (Declaration)

Public Function TruncateUnusedPages AsUInteger

Visual C++

public:

unsigned int TruncateUnusedPages()

Return Value

The number of database pages returned to the filesystem
See Also

RecnoDatabase Class
TruncateUnusedPages Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
RecnoDatabase...:::TruncateUnusedPages Method (Transaction)

RecnoDatabase Class  See Also

Return pages to the filesystem that are already free and at the end of the file.

Namespace:  BerkeleyDB
Syntax

C#

```csharp
public uint TruncateUnusedPages(
    Transaction txn
)
```

Visual Basic (Declaration)

```vbnet
Public Function TruncateUnusedPages ( _
    As Transaction _
) As UInteger
```

Visual C++

```cpp
public:
    unsigned int TruncateUnusedPages(
        Transaction^ txn
    )
```

Parameters

txn

Type: `BerkeleyDB..::.Transaction`

If the operation is part of an application-specified transaction, txn is a Transaction object returned from `BeginTransaction()`; if the operation is part of a Berkeley DB Concurrent Data Store group, txn is a handle returned from `BeginCDSGroup()`; otherwise null.

Return Value

The number of database pages returned to the filesystem
See Also

RecnoDatabase Class
TruncateUnusedPages Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `RecnoDatabase` type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AppendCallback</td>
<td>A function to call after the record number has been selected but before the data has been stored into the database.</td>
</tr>
<tr>
<td>AutoCommit</td>
<td>If true, all database modification operations based on this object will be transactionally protected. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>CacheSize</td>
<td>The size of the shared memory buffer pool -- that is, the cache. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>Creation</td>
<td>The CreatePolicy with which this database was opened. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>DatabaseName</td>
<td>The name of this database, if it has one. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>DoChecksum</td>
<td>If true, do checksum verification of pages read into the cache from the backing filestore. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>EncryptAlgorithm</td>
<td>The algorithm used by the Berkeley DB library to perform encryption and decryption. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>Encrypted</td>
<td>If true, encrypt all data stored in the database. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>Endianness</td>
<td>The database byte order. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>ErrorFeedback</td>
<td>The mechanism for reporting detailed error messages to the application. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>ErrorPrefix</td>
<td>The prefix string that appears before error messages issued by Berkeley DB. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td></td>
<td>Monitor progress within long running operations.</td>
</tr>
<tr>
<td><strong>Feedback</strong></td>
<td>(Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
<tr>
<td><strong>FileName</strong></td>
<td>The filename of this database, if it has one. (Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
<tr>
<td><strong>FreeThreaded</strong></td>
<td>If true, the object is free-threaded; that is, concurrently usable by multiple threads in the address space. (Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
<tr>
<td><strong>HasMultiple</strong></td>
<td>If true, the object references a physical file supporting multiple databases. (Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
<tr>
<td><strong>InHostOrder</strong></td>
<td>If true, the underlying database files were created on an architecture of the same byte order as the current one. This information may be used to determine whether application data needs to be adjusted for this architecture or not. (Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
<tr>
<td><strong>NoMMap</strong></td>
<td>If true, this database is not mapped into process memory.</td>
</tr>
<tr>
<td><strong>NonDurableTxns</strong></td>
<td>See <a href="#">MMapSize</a> for further information. (Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
<tr>
<td><strong>Pagesize</strong></td>
<td>The database's current page size. (Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
<tr>
<td><strong>Priority</strong></td>
<td>The cache priority for pages referenced by this object. (Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
<tr>
<td><strong>ReadOnly</strong></td>
<td>If true, this database has been opened for reading only. Any attempt to modify items in the database will fail, regardless of the actual permissions of any underlying files. (Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
<tr>
<td><strong>ReadOnly</strong></td>
<td>If true, this database supports transactional read</td>
</tr>
<tr>
<td><strong>ReadUncommitted</strong></td>
<td>Operations with degree 1 isolation. Read operations on the database may request the return of modified but not yet committed data. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>---------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>RecordDelimiter</strong></td>
<td>The delimiting byte used to mark the end of a record in SourceFile.</td>
</tr>
<tr>
<td><strong>RecordLength</strong></td>
<td>If using fixed-length, not byte-delimited records, the length of the records.</td>
</tr>
<tr>
<td><strong>RecordPad</strong></td>
<td>The padding character for short, fixed-length records.</td>
</tr>
<tr>
<td><strong>Renumber</strong></td>
<td>If true, the logical record numbers are mutable, and change as records are added to and deleted from the database.</td>
</tr>
<tr>
<td><strong>Snapshot</strong></td>
<td>If true, any SourceFile file will be read in its entirety when Open(String, RecnoDatabaseConfig) is called. If false, SourceFile may be read lazily.</td>
</tr>
<tr>
<td><strong>SourceFile</strong></td>
<td>The underlying source file for the Recno access method.</td>
</tr>
<tr>
<td><strong>Transactional</strong></td>
<td>If true, this database has been opened in a transactional mode. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>Truncated</strong></td>
<td>If true, the underlying file was physically truncated upon open, discarding all previous databases it might have held. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>Type</strong></td>
<td>The type of the underlying access method (and file format). This value may be used to determine the type of the database after an Open(String, DatabaseConfig). (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>UseMVCC</strong></td>
<td>If true, the database was opened with support for multiversion concurrency control. (Inherited from BaseDatabase.)</td>
</tr>
</tbody>
</table>
See Also

RecnoDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
A function to call after the record number has been selected but before the data has been stored into the database.

**Namespace:**  BerkeleyDB

**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public AppendRecordDelegate AppendCallback { get; set; }

Visual Basic (Declaration)

Public Property AppendCallback As AppendRecordDelegate

Visual C++

public:
property AppendRecordDelegate^ AppendCallback {
AppendRecordDelegate^ get ();
void set (AppendRecordDelegate^ value);
}

Remarks

When using `Append(DatabaseEntry)`, it may be useful to modify the stored data based on the generated key. If a delegate is specified, it will be called after the record number has been selected, but before the data has been stored.
See Also

RecnoDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The delimiting byte used to mark the end of a record in *SourceFile*.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
### Syntax

**C#**

```csharp
public int RecordDelimiter { get; private set; }
```

**Visual Basic (Declaration)**

```vbnet
Public Property RecordDelimiter As Integer
```

**Visual C++**

```cpp
public:
property int RecordDelimiter {
    int get ();
    void set (int value);
}
```
See Also

RecnoDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If using fixed-length, not byte-delimited records, the length of the records.

**Namespace:**  [BerkeleyDB](https://github.com/berkeley-db-net)

**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
**Syntax**

**C#**

```csharp
public uint RecordLength { get; private set; }
```

**Visual Basic (Declaration)**

```vbnet
Public Property RecordLength AsUInteger
```

**Visual C++**

```cpp
public:
property unsigned int RecordLength {
    unsigned int get();
    void set (unsigned int value);
}
```
See Also

RecnoDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The padding character for short, fixed-length records.

_Namespace:_  [BerkeleyDB](#)  
_Assembly:_  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public int RecordPad { get; private set; }

Visual Basic (Declaration)

Public Property RecordPad As Integer

Visual C++

public:
property int RecordPad {
    int get ();
    void set (int value);
}
See Also

RecnoDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, the logical record numbers are mutable, and change as records are added to and deleted from the database.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#
public bool Renumber { get; }

Visual Basic (Declaration)
Public ReadOnly Property Renumber As Boolean

Visual C++
public:
property bool Renumber {
    bool get();
}
See Also

RecnoDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
RecnoDatabase::Snapshot Property

If true, any SourceFile file will be read in its entirety when Open(String, RecnoDatabaseConfig) is called. If false, SourceFile may be read lazily.

Namespace: BerkeleyDB
Syntax

C#

```csharp
public bool Snapshot { get; }
```

Visual Basic (Declaration)

```vbnet
Public ReadOnly Property Snapshot As Boolean
```

Visual C++

```cpp
public:
property bool Snapshot {
    bool get ();
}
```
See Also

RecnoDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The underlying source file for the Recno access method.

**Namespace:** [BerkeleyDB](https://www.berdex.org)
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public string SourceFile { get; private set; }

Visual Basic (Declaration)

Public Property SourceFile As String

Visual C++

public:
property String^ SourceFile {
    String^ get ();
    void set (String^ value);
}
See Also

RecnoDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation
RecnoDatabaseConfig Class

A class representing configuration parameters for RecnoDatabase

**Namespace:**  BerkeleyDB
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
### Syntax

**C#**

public class RecnoDatabaseConfig : DatabaseConfig

**Visual Basic (Declaration)**

Public Class RecnoDatabaseConfig
    Inherits DatabaseConfig

**Visual C++**

public ref class RecnoDatabaseConfig : public DatabaseConfig
Inheritance Hierarchy

System::Object
BerkeleyDB::DatabaseConfig
BerkeleyDB::RecnoDatabaseConfig
See Also

RecnoDatabaseConfig Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The *RecnoDatabaseConfig* type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RecnoDatabaseConfig</td>
<td>Instantiate a new RecnoDatabaseConfig object</td>
</tr>
</tbody>
</table>
### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="#">Object</a> is equal to the current Object. (Inherited from <a href="#">Object</a>)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="#">Object</a>)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the Type of the current instance. (Inherited from <a href="#">Object</a>)</td>
</tr>
<tr>
<td><strong>SetEncryption</strong></td>
<td>Set the password and algorithm used by the Berkeley DB library to perform encryption and decryption. (Inherited from <a href="#">DatabaseConfig</a>)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a String that represents the current Object. (Inherited from <a href="#">Object</a>)</td>
</tr>
</tbody>
</table>
# Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Append</td>
<td>A function to call after the record number has been selected but before the data has been stored into the database.</td>
</tr>
<tr>
<td></td>
<td>Enclose the open call within a transaction. If the call succeeds, the open operation will be recoverable and all subsequent database modification operations based on this handle will be transactionally protected. If the call fails, no database will have been created. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td>AutoCommit</td>
<td></td>
</tr>
<tr>
<td>BackingFile</td>
<td>The underlying source file for the Recno access method.</td>
</tr>
<tr>
<td></td>
<td>The byte order for integers in the stored database metadata. The host byte order of the machine where the Berkeley DB library was compiled is the default value. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td>ByteOrder</td>
<td></td>
</tr>
<tr>
<td>CacheSize</td>
<td>The size of the shared memory buffer pool -- that is, the cache. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td>Creation</td>
<td>The policy for how to handle database creation. If true, do checksum verification of pages read into the cache from the backing filestore. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td>DoChecksum</td>
<td>The Berkeley DB environment within which to create a database. If null, the database will be created stand-alone; that is, it is not part of any Berkeley DB environment. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td>Env</td>
<td>The mechanism for reporting error messages to the application. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td>ErrorFeedback</td>
<td></td>
</tr>
</tbody>
</table>
- **ErrorPrefix**: The prefix string that appears before error messages issued by Berkeley DB. (Inherited from [DatabaseConfig](#)).
- **Feedback**: Cause the database object to be free-threaded; that is, concurrently usable by multiple threads in the address space. (Inherited from [DatabaseConfig](#)).
- **FreeThreaded**: Do not map this database into process memory. (Inherited from [DatabaseConfig](#)).
- **NoMMap**: If true, Berkeley DB will not write log records for this database. (Inherited from [DatabaseConfig](#)).
- **NonDurableTxns**: The cache priority for pages referenced by the database. (Inherited from [DatabaseConfig](#)).
- **Priority**: Open the database for reading only. Any attempt to modify items in the database will fail, regardless of the actual permissions of any underlying files. (Inherited from [DatabaseConfig](#)).
- **ReadOnly**: Support transactional read operations with degree 1 isolation. (Inherited from [DatabaseConfig](#)).
- **ReadUncommitted**: Cause the logical record numbers to be mutable, and change as records are added to and deleted from the database. (Inherited from [DatabaseConfig](#)).
- **Renumber**: Physically truncate the underlying file, discarding all previous databases it might have held. (Inherited from [DatabaseConfig](#)).
- **Snapshot**: Open the database with support for multiversion concurrency control. (Inherited from [DatabaseConfig](#)).
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Delimiter</strong></td>
<td>The delimiting byte used to mark the end of a record in <a href="#">BackingFile</a>.</td>
</tr>
<tr>
<td><strong>EncryptAlgorithm</strong></td>
<td>The algorithm used to perform encryption and decryption. (Inherited from <a href="#">DatabaseConfig</a>.</td>
</tr>
<tr>
<td><strong>EncryptionPassword</strong></td>
<td>The password used to perform encryption and decryption. (Inherited from <a href="#">DatabaseConfig</a>.</td>
</tr>
<tr>
<td><strong>Length</strong></td>
<td>Specify that the records are fixed-length, not byte-delimited, and are of length Length.</td>
</tr>
<tr>
<td><strong>PadByte</strong></td>
<td>The padding character for short, fixed-length records.</td>
</tr>
<tr>
<td><strong>PageSize</strong></td>
<td>The size of the pages used to hold items in the database, in bytes. (Inherited from <a href="#">DatabaseConfig</a>.</td>
</tr>
</tbody>
</table>
See Also

RecnoDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Instantiate a new RecnoDatabaseConfig object

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public RecnoDatabaseConfig()

Visual Basic (Declaration)

Public Sub New

Visual C++

public:
RecnoDatabaseConfig()}
See Also

RecnoDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `RecnoDatabaseConfig` type exposes the following members.
# Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Append</strong></td>
<td>A function to call after the record number has been selected but before the data has been stored into the database. Enclose the open call within a transaction. If the call succeeds, the open operation will be recoverable and all subsequent database modification operations based on this handle will be transactionally protected. If the call fails, no database will have been created. (Inherited from <code>DatabaseConfig</code>.)</td>
</tr>
<tr>
<td><strong>AutoCommit</strong></td>
<td>The underlying source file for the Recno access method. The byte order for integers in the stored database metadata. The host byte order of the machine where the Berkeley DB library was compiled is the default value. (Inherited from <code>DatabaseConfig</code>.)</td>
</tr>
<tr>
<td><strong>BackingFile</strong></td>
<td>The byte order for integers in the stored database metadata. The host byte order of the machine where the Berkeley DB library was compiled is the default value. (Inherited from <code>DatabaseConfig</code>.)</td>
</tr>
<tr>
<td><strong>ByteOrder</strong></td>
<td>The policy for how to handle database creation. If true, do checksum verification of pages read into the cache from the backing filestore. (Inherited from <code>DatabaseConfig</code>.)</td>
</tr>
<tr>
<td><strong>CacheSize</strong></td>
<td>The Berkeley DB environment within which to create a database. If null, the database will be created stand-alone; that is, it is not part of any Berkeley DB environment. (Inherited from <code>DatabaseConfig</code>.)</td>
</tr>
<tr>
<td><strong>Creation</strong></td>
<td>The mechanism for reporting error messages to the application. (Inherited from <code>DatabaseConfig</code>.)</td>
</tr>
<tr>
<td><strong>DoChecksum</strong></td>
<td>The Berkeley DB environment within which to create a database. If null, the database will be created stand-alone; that is, it is not part of any Berkeley DB environment. (Inherited from <code>DatabaseConfig</code>.)</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ErrorPrefix</td>
<td>The prefix string that appears before error messages issued by Berkeley DB. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td>Feedback</td>
<td>Cause the database object to be free-threaded; that is, concurrently usable by multiple threads in the address space. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td>FreeThreaded</td>
<td>Do not map this database into process memory. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td>NonMMap</td>
<td>If true, Berkeley DB will not write log records for this database. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td>NonDurableTxns</td>
<td>The cache priority for pages referenced by the database. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td>Priority</td>
<td>Open the database for reading only. Any attempt to modify items in the database will fail, regardless of the actual permissions of any underlying files. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td>ReadOnly</td>
<td>Support transactional read operations with degree 1 isolation. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td>ReadUncommitted</td>
<td>Cause the logical record numbers to be mutable, and change as records are added to and deleted from the database. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td>Renumber</td>
<td>If true, any BackingFile file will be read in its entirety when Open(String, RecnoDatabaseConfig) is called. If false, BackingFile may be read lazily.</td>
</tr>
<tr>
<td>Snapshot</td>
<td>Physically truncate the underlying file, discarding all previous databases it might have held. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td>Truncate</td>
<td>Open the database with support for multiversion concurrency control. (Inherited from DatabaseConfig.)</td>
</tr>
</tbody>
</table>
See Also

RecnoDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
A function to call after the record number has been selected but before the data has been stored into the database.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public AppendRecordDelegate Append

Visual Basic (Declaration)

Public Append As AppendRecordDelegate

Visual C++

public:
AppendRecordDelegate^ Append
Remarks

When using `Append(DatabaseEntry)`, it may be useful to modify the stored data based on the generated key. If a delegate is specified, it will be called after the record number has been selected, but before the data has been stored.
See Also

RecnoDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The underlying source file for the Recno access method.

**Namespace:**  BerkeleyDB  
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll)  Version: 4.8.24.0
Syntax

C#

public string BackingFile

Visual Basic (Declaration)

Public BackingFile As String

Visual C++

public:
String^ BackingFile
Remarks

The purpose of the source file is to provide fast access and modification to databases that are normally stored as flat text files.

The source parameter specifies an underlying flat text database file that is read to initialize a transient record number index. In the case of variable length records, the records are separated, as specified by Delimiter. For example, standard UNIX byte stream files can be interpreted as a sequence of variable length records separated by newline characters.

In addition, when cached data would normally be written back to the underlying database file (for example, Close() or Sync()), the in-memory copy of the database will be written back to the source file.

By default, the backing source file is read lazily; that is, records are not read from the file until they are requested by the application. If multiple processes (not threads) are accessing a Recno database concurrently, and are either inserting or deleting records, the backing source file must be read in its entirety before more than a single process accesses the database, and only that process should specify the backing source file as part of the Open(String, RecnoDatabaseConfig) call. See Snapshot for more information.

Reading and writing the backing source file specified by source cannot be transaction-protected because it involves filesystem operations that are not part of the Db transaction methodology. For this reason, if a temporary database is used to hold the records, it is possible to lose the contents of the source file, for example, if the system crashes at the right instant. If a file is used to hold the database, normal database recovery on that file can be used to prevent information loss, although it is still possible that the contents of source will be lost if the system crashes.

The source file must already exist (but may be zero-length) when Open(String, RecnoDatabaseConfig) is called.

It is not an error to specify a read-only source file when creating a database, nor is it an error to modify the resulting database. However, any attempt to write the
changes to the backing source file using either the `Sync()` or `Close()` methods will fail, of course. Use `Close(Boolean)` to stop it from attempting to write the changes to the backing file; instead, they will be silently discarded.

For all of the previous reasons, the source file is generally used to specify databases that are read-only for Berkeley DB applications; and that are either generated on the fly by software tools or modified using a different mechanism — for example, a text editor.

If the database already exists, BackingFile must be the same as that historically used to create the database or corruption can occur.
See Also

RecnoDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The policy for how to handle database creation.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

**C#**

```csharp
public CreatePolicy Creation
```

**Visual Basic (Declaration)**

```vbnet
Public Creation As CreatePolicy
```

**Visual C++**

```c++
public:
CreatePolicy Creation
```
Remarks

If the database does not already exist and NEVER is set, Open(String, RecnoDatabaseConfig) will fail.
See Also

RecnoDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
RecnoDatabaseConfig::Renumber Field

Cause the logical record numbers to be mutable, and change as records are added to and deleted from the database.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool Renumber

Visual Basic (Declaration)

Public Renumber As Boolean

Visual C++

public:

bool Renumber
Remarks

Using Put(DatabaseEntry, DatabaseEntry) or Put(DatabaseEntry, DatabaseEntry, UInt32) to create new records will cause the creation of multiple records if the record number is more than one greater than the largest record currently in the database. For example, creating record 28, when record 25 was previously the last record in the database, will create records 26 and 27 as well as 28. Attempts to retrieve records that were created in this manner will throw a KeyEmptyException.

If a created record is not at the end of the database, all records following the new record will be automatically renumbered upward by one. For example, the creation of a new record numbered 8 causes records numbered 8 and greater to be renumbered upward by one. If a cursor was positioned to record number 8 or greater before the insertion, it will be shifted upward one logical record, continuing to refer to the same record as it did before.

If a deleted record is not at the end of the database, all records following the removed record will be automatically renumbered downward by one. For example, deleting the record numbered 8 causes records numbered 9 and greater to be renumbered downward by one. If a cursor was positioned to record number 9 or greater before the removal, it will be shifted downward one logical record, continuing to refer to the same record as it did before.

If a record is deleted, all cursors that were positioned on that record prior to the removal will no longer be positioned on a valid entry. This includes cursors used to delete an item. For example, if a cursor was positioned to record number 8 before the removal of that record, subsequent calls to Refresh() will return false until the cursor is moved to another record. A call to MoveNext() will return the new record numbered 8 - which is the record that was numbered 9 prior to the delete (if such a record existed).

For these reasons, concurrent access to a RecnoDatabase with this setting specified may be largely meaningless, although it is supported.

If the database already exists, this setting must be the same as the existing database or an exception will be thrown.
See Also

RecnoDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
RecnoDatabaseConfig Class  See Also

If true, any BackingFile file will be read in its entirety when Open(String, RecnoDatabaseConfig) is called. If false, BackingFile may be read lazily.

Namespace: BerkeleyDB
Syntax

C#

public bool Snapshot

Visual Basic (Declaration)

Public Snapshot As Boolean

Visual C++

public:
    bool Snapshot
See Also

RecnoDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `RecnoDatabaseConfig` type exposes the following members.
### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>SetEncryption</strong></td>
<td>Set the password and algorithm used by the Berkeley DB library to perform encryption and decryption. (Inherited from <a href="#">DatabaseConfig</a>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a <a href="#">String</a> that represents the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
</tbody>
</table>

[Object](#): The keyword `Object` refers to the `Object` class in C#. It is used in C# code to indicate an object, which can be any type of object supported by the language.
See Also

RecnoDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The **RecnoDatabaseConfig** type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Delimiter</strong></td>
<td>The delimiting byte used to mark the end of a record in <a href="#">BackingFile</a>.</td>
</tr>
<tr>
<td></td>
<td>The algorithm used to perform encryption and decryption.</td>
</tr>
<tr>
<td><strong>EncryptAlgorithm</strong></td>
<td>(Inherited from <a href="#">DatabaseConfig</a>.</td>
</tr>
<tr>
<td></td>
<td>The password used to perform encryption and decryption.</td>
</tr>
<tr>
<td><strong>EncryptionPassword</strong></td>
<td>(Inherited from <a href="#">DatabaseConfig</a>.</td>
</tr>
<tr>
<td><strong>Length</strong></td>
<td>Specify that the records are fixed-length, not byte-delimited, and are of length <strong>Length</strong>.</td>
</tr>
<tr>
<td><strong>PadByte</strong></td>
<td>The padding character for short, fixed-length records.</td>
</tr>
<tr>
<td><strong>PageSize</strong></td>
<td>The size of the pages used to hold items in the database, in bytes.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">DatabaseConfig</a>.</td>
</tr>
</tbody>
</table>
See Also

RecnoDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
RecnoDatabaseConfig.Delimiter Property

The delimiting byte used to mark the end of a record in BackingFile.

**Namespace:** BerkeleyDB
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public int Delimiter { get; set; }

Visual Basic (Declaration)

Public Property Delimiter As Integer

Visual C++

public:
property int Delimiter {
    int get ();
    void set (int value);
}
Remarks

This byte is used for variable length records if BackingFile is set. If BackingFile is specified and no delimiting byte was specified, newline characters (that is, ASCII 0x0a) are interpreted as end-of-record markers.

If the database already exists, this setting will be ignored.
See Also

RecnoDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Specify that the records are fixed-length, not byte-delimited, and are of length Length.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
### Syntax

**C#**

```csharp
public uint Length { get; set; }
```

**Visual Basic (Declaration)**

```vbnet
Public Property Length AsUInteger
```

**Visual C++**

```cpp
public:
property unsigned int Length {
    unsigned int get ();
    void set (unsigned int value);
}
```
Remarks

Any records added to the database that are less than Length bytes long are automatically padded (see PadByte for more information).

Any attempt to insert records into the database that are greater than Length bytes long will cause the call to fail immediately and return an error.

If the database already exists, this setting will be ignored.
See Also

RecnoDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The padding character for short, fixed-length records.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public int PadByte { get; set; }

Visual Basic (Declaration)

Public Property PadByte As Integer

Visual C++

public:
property int PadByte {
    int get ();
    void set (int value);
}
Remarks

If no pad character is specified, space characters (that is, ASCII 0x20) are used for padding.

If the database already exists, this setting will be ignored.
See Also

RecnoDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Statistical information about a RecnoDatabase

**Namespace:**  BerkeleyDB  
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll)  
Version: 4.8.24.0
Syntax

C#

public class RecnoStats

Visual Basic (Declaration)

Public Class RecnoStats

Visual C++

public ref class RecnoStats
Inheritance Hierarchy

System:::Object

BerkeleyDB:::RecnoStats
See Also

RecnoStats Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The **RecnoStats** type exposes the following members.
**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>equals</strong></td>
<td>Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a>.</td>
</tr>
<tr>
<td><strong>getHashCode</strong></td>
<td>Serves as a hash function for a particular type.</td>
</tr>
<tr>
<td><strong>getType</strong></td>
<td>Gets the <a href="#">Type</a> of the current instance.</td>
</tr>
<tr>
<td><strong>toString</strong></td>
<td>Returns a <a href="#">String</a> that represents the current <a href="#">Object</a>.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>---------------------------------</td>
</tr>
<tr>
<td>DuplicatePages</td>
<td>Duplicate pages.</td>
</tr>
<tr>
<td>DuplicatePagesFreeBytes</td>
<td>Bytes free in duplicate pages.</td>
</tr>
<tr>
<td>EmptyPages</td>
<td>Empty pages.</td>
</tr>
<tr>
<td>FreePages</td>
<td>Pages on the free list.</td>
</tr>
<tr>
<td>InternalPages</td>
<td>Internal pages.</td>
</tr>
<tr>
<td>InternalPagesFreeBytes</td>
<td>Bytes free in internal pages.</td>
</tr>
<tr>
<td>LeafPages</td>
<td>Leaf pages.</td>
</tr>
<tr>
<td>LeafPagesFreeBytes</td>
<td>Bytes free in leaf pages.</td>
</tr>
<tr>
<td>Levels</td>
<td>Tree levels.</td>
</tr>
<tr>
<td>MagicNumber</td>
<td>Magic number.</td>
</tr>
<tr>
<td>MetadataFlags</td>
<td>Metadata flags.</td>
</tr>
<tr>
<td>MinKey</td>
<td>Minkey value.</td>
</tr>
<tr>
<td>nData</td>
<td>Number of data items.</td>
</tr>
<tr>
<td>nKeys</td>
<td>Number of unique keys.</td>
</tr>
<tr>
<td>nPages</td>
<td>Page count.</td>
</tr>
<tr>
<td>OverflowPages</td>
<td>Overflow pages.</td>
</tr>
<tr>
<td>OverflowPagesFreeBytes</td>
<td>Bytes free in overflow pages.</td>
</tr>
<tr>
<td>PageSize</td>
<td>Page size.</td>
</tr>
<tr>
<td>RecordLength</td>
<td>Fixed-length record length.</td>
</tr>
<tr>
<td>RecordPadByte</td>
<td>Fixed-length record pad.</td>
</tr>
<tr>
<td>Version</td>
<td>Version number.</td>
</tr>
</tbody>
</table>
See Also

RecnoStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The **RecnoStats** type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a> is equal to the current <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <a href="https://docs.microsoft.com/en-us/dotnet/api/system.type">Type</a> of the current instance. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a <a href="https://docs.microsoft.com/en-us/dotnet/api/system.string">String</a> that represents the current <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
</tbody>
</table>
See Also

RecnoStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The **RecnoStats** type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DuplicatePages</td>
<td>Duplicate pages.</td>
</tr>
<tr>
<td>DuplicatePagesFreeBytes</td>
<td>Bytes free in duplicate pages.</td>
</tr>
<tr>
<td>EmptyPages</td>
<td>Empty pages.</td>
</tr>
<tr>
<td>FreePages</td>
<td>Pages on the free list.</td>
</tr>
<tr>
<td>InternalPages</td>
<td>Internal pages.</td>
</tr>
<tr>
<td>InternalPagesFreeBytes</td>
<td>Bytes free in internal pages.</td>
</tr>
<tr>
<td>LeafPages</td>
<td>Leaf pages.</td>
</tr>
<tr>
<td>LeafPagesFreeBytes</td>
<td>Bytes free in leaf pages.</td>
</tr>
<tr>
<td>Levels</td>
<td>Tree levels.</td>
</tr>
<tr>
<td>MagicNumber</td>
<td>Magic number.</td>
</tr>
<tr>
<td>MetadataFlags</td>
<td>Metadata flags.</td>
</tr>
<tr>
<td>MinKey</td>
<td>Minkey value.</td>
</tr>
<tr>
<td>nData</td>
<td>Number of data items.</td>
</tr>
<tr>
<td>nKeys</td>
<td>Number of unique keys.</td>
</tr>
<tr>
<td>nPages</td>
<td>Page count.</td>
</tr>
<tr>
<td>OverflowPages</td>
<td>Overflow pages.</td>
</tr>
<tr>
<td>OverflowPagesFreeBytes</td>
<td>Bytes free in overflow pages.</td>
</tr>
<tr>
<td>PageSize</td>
<td>Page size.</td>
</tr>
<tr>
<td>RecordLength</td>
<td>Fixed-length record length.</td>
</tr>
<tr>
<td>RecordPadByte</td>
<td>Fixed-length record pad.</td>
</tr>
<tr>
<td>Version</td>
<td>Version number.</td>
</tr>
</tbody>
</table>
See Also

RecnoStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Duplicate pages.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
## Syntax

### C#

```csharp
public uint DuplicatePages { get; }
```

### Visual Basic (Declaration)

```vbnet
Public ReadOnly Property DuplicatePages AsUInteger
```

### Visual C++

```cpp
public:
    property unsigned int DuplicatePages {
        unsigned int get ();
    }
```

See Also

RecnoStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
RecnoStats.DuplicatePagesFreeBytes Property

Bytes free in duplicate pages.

Namespace: BerkeleyDB
**Syntax**

**C#**

```csharp
public ulong DuplicatePagesFreeBytes { get; }
```

**Visual Basic (Declaration)**

```vbnet
Public ReadOnly Property DuplicatePagesFreeBytes As ULong
```

**Visual C++**

```cpp
public:
    property unsigned long long DuplicatePagesFreeBytes {
        unsigned long long get ()
    }
```

See Also

RecnoStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
RecnoStats Class  See Also

Empty pages.

Namespace:  BerkeleyDB
Syntax

C#

public uint EmptyPages { get; }

Visual Basic (Declaration)

Public ReadOnly Property EmptyPages AsUInteger

Visual C++

public:
property unsigned int EmptyPages {
    unsigned int get ();
}
See Also

RecnoStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Pages on the free list.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll)  
**Version:** 4.8.24.0
**Syntax**

**C#**

```csharp
public uint FreePages { get; }
```

**Visual Basic (Declaration)**

```vbnet
Public ReadOnly Property FreePages AsUInteger
```

**Visual C++**

```cpp
public:
property unsigned int FreePages {
    unsigned int get ();
}
```
See Also

RecnoStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Internal pages.

**Namespace:**  [BerkeleyDB](https://www.berkeleydb.com)

**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public uint InternalPages { get; }

Visual Basic (Declaration)

Public ReadOnly Property InternalPages AsUInteger

Visual C++

public:
property unsigned int InternalPages {
    unsigned int get ();
}

See Also

RecnoStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Bytes free in internal pages.

**Namespace:**  BerkeleyDB  
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
### Syntax

**C#**

```csharp
public ulong InternalPagesFreeBytes { get; }
```

**Visual Basic (Declaration)**

```vbnet
Public Readonly Property InternalPagesFreeBytes As ULong
```

**Visual C++**

```cpp
public:
    property unsigned long long InternalPagesFreeBytes {
        unsigned long long get();
    }
```
See Also

RecnoStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation
RecnoStats.::.LeafPages Property

RecnoStats Class

See Also

Leaf pages.

Namespace: BerkeleyDB
Syntax

**C#**

```csharp
public uint LeafPages { get; }
```

**Visual Basic (Declaration)**

```vbnet
Public ReadOnly Property LeafPages As UInteger
```

**Visual C++**

```cpp
public:
property unsigned int LeafPages {
    unsigned int get ();
}
```
See Also

RecnoStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
RecnoStats Class  See Also

Bytes free in leaf pages.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public ulong LeafPagesFreeBytes { get; }

Visual Basic (Declaration)

Public ReadOnly Property LeafPagesFreeBytes As ULong

Visual C++

public:
property unsigned long long LeafPagesFreeBytes {
     unsigned long long get ();
}
See Also

RecnoStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
RecnoStats...:::Levels Property

RecnoStats Class  See Also

Tree levels.

Namespace:  BerkeleyDB
Syntax

C#

public uint Levels { get; }

Visual Basic (Declaration)

Public ReadOnly Property Levels AsUInteger

Visual C++

public:
property unsigned int Levels {
    unsigned int get ();
}

See Also

RecnoStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Magic number.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public uint MagicNumber { get; }

Visual Basic (Declaration)

Public ReadOnly Property MagicNumber As UInteger

Visual C++

public:
property unsigned int MagicNumber {
    unsigned int get ();
}

See Also

RecnoStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Metadata flags.

**Namespace:**  [BerkeleyDB](https://www.berkeleydb.com/)

**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
### Syntax

**C#**

```csharp
public uint MetadataFlags { get; }
```

**Visual Basic (Declaration)**

```vbnet
Public ReadOnly Property MetadataFlags AsUInteger
```

**Visual C++**

```cpp
public:
property unsigned int MetadataFlags {
    unsigned int get ();
}
```
See Also

RecnoStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Minkey value.

**Namespace:**  [BerkeleyDB](https://www.berkeleydb.com)

**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public uint MinKey { get; }

Visual Basic (Declaration)

Public ReadOnly Property MinKey AsUInteger

Visual C++

public:
property unsigned int MinKey {
    unsigned int get ()
}

See Also

RecnoStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
RecnoStats Class  See Also

Number of data items.

Namespace:  BerkeleyDB
Syntax

C#

public uint nData { get; }

Visual Basic (Declaration)

Public ReadOnly Property nData AsUInteger

Visual C++

public:
property unsigned int nData {
    unsigned int get ();
}

See Also

RecnoStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
C# Visual Basic Visual C++

Berkeley DB .NET API Documentation

RecnoStats:::nKeys Property

RecnoStats Class  See Also

Number of unique keys.

Namespace:  BerkeleyDB
Syntax

C#

public uint nKeys { get; }

Visual Basic (Declaration)

Public ReadOnly Property nKeys AsUInteger

Visual C++

public:
property unsigned int nKeys {
    unsigned int get ();
}

See Also

RecnoStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
RecnoStats Class  See Also
Page count.

Namespace: BerkeleyDB
Syntax

C#

public uint nPages { get; }

Visual Basic (Declaration)

Public ReadOnly Property nPages As UInteger

Visual C++

public:
property unsigned int nPages {
    unsigned int get ();
}

See Also

RecnoStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
RecnoStats Class  See Also

Overflow pages.

Namespace:  BerkeleyDB
Syntax

C#

public uint OverflowPages { get; }

Visual Basic (Declaration)

Public ReadOnly Property OverflowPages AsUInteger

Visual C++

public:
property unsigned int OverflowPages {
    unsigned int get ();
}

See Also

RecnoStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Bytes free in overflow pages.

**Namespace:**  [BerkeleyDB](https://www.berkeleydb.com)

**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

**C#**

```csharp
public ulong OverflowPagesFreeBytes { get; }
```

**Visual Basic (Declaration)**

```vbnet
Public ReadOnly Property OverflowPagesFreeBytes As ULong
```

**Visual C++**

```cpp
public:
property unsigned long long OverflowPagesFreeBytes {
    unsigned long long get ();
}
```
See Also

RecnoStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Page size.

**Namespace:**  BerkeleyDB  
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public uint PageSize { get; }

Visual Basic (Declaration)

Public ReadOnly Property PageSize AsUInteger

Visual C++

public:
property unsigned int PageSize {
    unsigned int get ();
}
See Also

RecnoStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
RecnoStats Class

See Also

Fixed-length record length.

Namespace: BerkeleyDB
## Syntax

**C#**

```csharp
public uint RecordLength { get; }
```

**Visual Basic (Declaration)**

```vbnet
Public ReadOnly Property RecordLength As UInteger
```

**Visual C++**

```cpp
public:
    property unsigned int RecordLength {
        unsigned int get ();
    }
```


See Also

RecnoStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
RecnoStats Class

Fixed-length record pad.

Namespace: BerkeleyDB
Syntax

C#

public uint RecordPadByte { get; }

Visual Basic (Declaration)

Public ReadOnly Property RecordPadByte AsUInteger

Visual C++

public:
property unsigned int RecordPadByte {
    unsigned int get ();
}

See Also

RecnoStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
RecnoStats Class

Version number.

Namespace: BerkeleyDB
Syntax

C#

public uint Version { get; }

Visual Basic (Declaration)

Public ReadOnly Property Version AsUInteger

Visual C++

public:
property unsigned int Version {
  unsigned int get ();
}


See Also

RecnoStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
ReplicationConfig Class

A class representing configuration parameters for a DatabaseEnvironment's replication subsystem.

Namespace: BerkeleyDB
Syntax

C#

public class ReplicationConfig

Visual Basic (Declaration)

Public Class ReplicationConfig

Visual C++

public ref class ReplicationConfig
Inheritance Hierarchy

System..::..Object
BerkeleyDB..::..ReplicationConfig
See Also

ReplicationConfig Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation

ReplicationConfig Members

The ReplicationConfig type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ReplicationConfig</td>
<td>Instantiate a new ReplicationConfig object with default configuration values.</td>
</tr>
</tbody>
</table>
### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AddRemoteSite</strong></td>
<td>Add a new replication site to the replication manager's list of known sites. It is not necessary for all sites in a replication group to know about all other sites in the group. Set the clock skew ratio among replication group members based on the fastest and slowest measurements among the group for use with master leases. Determines whether the specified <strong>Object</strong> is equal to the current <strong>Object</strong>. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>Clockskew</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>Equals</strong></td>
<td>Gets the <strong>Type</strong> of the current instance. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Set a threshold for the minimum and maximum time that a client waits before requesting retransmission of a missing message. Returns a <strong>String</strong> that represents the current <strong>Object</strong>. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Set a byte-count limit on the amount of data that will be transmitted from a site in response to a single message processed by <strong>RepProcessMessage(DatabaseEntry, DatabaseEntry, Int32)</strong>. The limit is not a hard limit, and the record that exceeds the limit is the last record to be sent.</td>
</tr>
<tr>
<td><strong>RetransmissionRequest</strong></td>
<td></td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>BulkTransfer</td>
<td>If true, the replication master will send groups of records to the clients in a single network transfer.</td>
</tr>
<tr>
<td></td>
<td>If true, the client will delay synchronizing to a newly declared master (defaults to false). Clients configured in this way will remain unsynchronized until the application calls RepSync().</td>
</tr>
<tr>
<td>DelayClientSync</td>
<td>If true, the client will delay synchronizing to a newly declared master (defaults to false). Clients configured in this way will remain unsynchronized until the application calls RepSync().</td>
</tr>
<tr>
<td></td>
<td>If true, Berkeley DB method calls that would normally block while clients are in recovery will return errors immediately (defaults to false).</td>
</tr>
<tr>
<td></td>
<td>Specify how master and client sites will handle acknowledgment of replication messages which are necessary for &quot;permanent&quot; records. The current implementation requires all sites in a replication group configure the same acknowledgement policy.</td>
</tr>
<tr>
<td>NoAutoInit</td>
<td>If true, the replication master will not automatically re-initialize outdated clients (defaults to false).</td>
</tr>
<tr>
<td></td>
<td>If true, Berkeley DB method calls that would normally block while clients are in recovery will return errors immediately (defaults to false).</td>
</tr>
<tr>
<td>NoBlocking</td>
<td>normally block while clients are in recovery will return errors immediately (defaults to false). Specify how master and client sites will handle acknowledgment of replication messages which are necessary for &quot;permanent&quot; records. The current implementation requires all sites in a replication group configure the same acknowledgement policy.</td>
</tr>
<tr>
<td>RepMgrAckPolicy</td>
<td>The host information for the local system.</td>
</tr>
<tr>
<td>RepMgrLocalSite</td>
<td>If true, the Replication Manager will observe the strict &quot;majority&quot; rule in managing elections, even in a group with only 2 sites. This means the client in a 2-site group will be unable to take over as master if the original master fails or becomes disconnected. (See the Elections section in the Berkeley DB Reference Guide for more information.) Both sites in the replication group should have the same value for this parameter.</td>
</tr>
<tr>
<td>Strict2Site</td>
<td>If true, the replication master will send groups of records to the clients in a single network transfer.</td>
</tr>
<tr>
<td></td>
<td>If true, the client will delay synchronizing to a newly declared master (defaults to false). Clients configured in this way will remain unsynchronized until the application calls RepSync().</td>
</tr>
<tr>
<td>Transport</td>
<td>The delegate used to transmit data using the replication application's communication infrastructure.</td>
</tr>
<tr>
<td>UseMasterLeases</td>
<td>If true, master leases will be used for this site (defaults to false).</td>
</tr>
</tbody>
</table>
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AckTimeout</td>
<td>The amount of time the replication manager's transport function waits to collect enough acknowledgments from replication group clients, before giving up and returning a failure indication. The default wait time is 1 second.</td>
</tr>
<tr>
<td>CheckpointDelay</td>
<td>The amount of time a master site will delay between completing a checkpoint and writing a checkpoint record into the log.</td>
</tr>
<tr>
<td>ClockskewFast</td>
<td>The value, relative to ClockskewSlow, of the fastest clock in the group of sites.</td>
</tr>
<tr>
<td>ClockskewSlow</td>
<td>The value of the slowest clock in the group of sites.</td>
</tr>
<tr>
<td>ConnectionRetry</td>
<td>The amount of time the replication manager will wait before trying to re-establish a connection to another site after a communication failure. The default wait time is 30 seconds.</td>
</tr>
<tr>
<td>ElectionRetry</td>
<td>Configure the amount of time the replication manager will wait before retrying a failed election. The default wait time is 10 seconds.</td>
</tr>
<tr>
<td>ElectionTimeout</td>
<td>The timeout period for an election. The default timeout is 2 seconds.</td>
</tr>
<tr>
<td>FullElectionTimeout</td>
<td>An optional configuration timeout period to wait for full election participation the first time the replication group finds a master. By default this option is turned off and normal election timeouts are used. (See the Elections section in the Berkeley DB Reference Guide for more information.)</td>
</tr>
</tbody>
</table>
HeartbeatMonitor

The amount of time the replication manager, running at a client site, waits for some message activity on the connection from the master (heartbeats or other messages) before concluding that the connection has been lost. When 0 (the default), no monitoring is performed.

HeartbeatSend

The frequency at which the replication manager, running at a master site, broadcasts a heartbeat message in an otherwise idle system. When 0 (the default), no heartbeat messages will be sent.

LeaseTimeout

The amount of time a client grants its master lease to a master. When using master leases all sites in a replication group must use the same lease timeout value. There is no default value. If leases are desired, this method must be called prior to calling RepStartClient() or RepStartMaster().

NSites

The total number of sites in the replication group.

Priority

The database environment's priority in replication group elections. A special value of 0 indicates that this environment cannot be a replication group master. If not configured, then a default value of 100 is used.

RetransmissionRequestMax

The maximum number of microseconds a client waits before requesting retransmission.

RetransmissionRequestMin

The minimum number of microseconds a client waits before requesting retransmission.

The bytes component of the byte-count limit on the amount of data that will be
TransmitLimitBytes transmitted from a site in response to a single message processed by RepProcessMessage(DatabaseEntry, DatabaseEntry, Int32).

The gigabytes component of the byte-count limit on the amount of data that will be transmitted from a site in response to a single message processed by RepProcessMessage(DatabaseEntry, DatabaseEntry, Int32).
See Also

ReplicationConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Instantiate a new ReplicationConfig object with default configuration values.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public ReplicationConfig()

Visual Basic (Declaration)

Public Sub New

Visual C++

public:
ReplicationConfig()
See Also

ReplicationConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `ReplicationConfig` type exposes the following members.
### Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BulkTransfer</strong></td>
<td>If true, the replication master will send groups of records to the clients in a single network transfer. If true, the client will delay synchronizing to a newly declared master (defaults to false). Clients configured in this way will remain unsynchronized until the application calls <code>RepSync()()</code>.</td>
</tr>
<tr>
<td><strong>DelayClientSync</strong></td>
<td>If true, the client will delay synchronizing to a newly declared master (defaults to false). Clients configured in this way will remain unsynchronized until the application calls <code>RepSync()()</code>.</td>
</tr>
<tr>
<td><strong>NoAutoInit</strong></td>
<td>If true, the replication master will not automatically re-initialize outdated clients (defaults to false). If true, Berkeley DB method calls that would normally block while clients are in recovery will return errors immediately (defaults to false). Specify how master and client sites will handle acknowledgment of replication messages which are necessary for &quot;permanent&quot; records. The current implementation requires all sites in a replication group configure the same acknowledgement policy.</td>
</tr>
<tr>
<td><strong>NoBlocking</strong></td>
<td>If true, the Replication Manager will observe the strict &quot;majority&quot; rule in managing elections, even in a group with only 2 sites. This means the client in a 2-site group will be unable to take over as master if the original master fails or becomes disconnected. (See the Elections section in the Berkeley DB Reference Guide for more information.) Both sites in the replication group should have the same value for this parameter.</td>
</tr>
<tr>
<td><strong>RepMgrAckPolicy</strong></td>
<td>The host information for the local system.</td>
</tr>
<tr>
<td><strong>RepMgrLocalSite</strong></td>
<td>The delegate used to transmit data using the replication application's communication infrastructure.</td>
</tr>
<tr>
<td><strong>UseMasterLeases</strong></td>
<td>If true, master leases will be used for this site (defaults to false).</td>
</tr>
</tbody>
</table>
See Also

ReplicationConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, the replication master will send groups of records to the clients in a single network transfer

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool BulkTransfer

Visual Basic (Declaration)

Public BulkTransfer As Boolean

Visual C++

public:
bool BulkTransfer
See Also

ReplicationConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, the client will delay synchronizing to a newly declared master (defaults to false). Clients configured in this way will remain unsynchronized until the application calls `RepSync()`.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
**Syntax**

**C#**

public bool DelayClientSync

**Visual Basic (Declaration)**

Public DelayClientSync As Boolean

**Visual C++**

public:
bool DelayClientSync
See Also

ReplicationConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, the replication master will not automatically re-initialize outdated clients (defaults to false).

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool NoAutoInit

Visual Basic (Declaration)

Public NoAutoInit As Boolean

Visual C++

public:
  bool NoAutoInit
See Also

ReplicationConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation
ReplicationConfig...: NoBlocking Field

ReplicationConfig Class  See Also

If true, Berkeley DB method calls that would normally block while clients are in recovery will return errors immediately (defaults to false).

Namespace: BerkeleyDB
Syntax

C#

public bool NoBlocking

Visual Basic (Declaration)

Public NoBlocking As Boolean

Visual C++

public:
bool NoBlocking
See Also

ReplicationConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation

ReplicationConfig:::RepMgrAckPolicy Field

ReplicationConfig Class  See Also

Specify how master and client sites will handle acknowledgment of replication messages which are necessary for "permanent" records. The current implementation requires all sites in a replication group configure the same acknowledgement policy.

Namespace: BerkeleyDB
Syntax

**C#**

```csharp
public AckPolicy RepMgrAckPolicy
```

**Visual Basic (Declaration)**

```vbnet
Public RepMgrAckPolicy As AckPolicy
```

**Visual C++**

```cpp
public:
    AckPolicy^ RepMgrAckPolicy
```
See Also

ReplicationConfig Class
BerkeleyDB Namespace
ReplicationConfig::AckTimeout

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The host information for the local system.

Namespace: BerkeleyDB
Syntax

C#

public ReplicationHostAddress RepMgrLocalSite

Visual Basic (Declaration)

Public RepMgrLocalSite As ReplicationHostAddress

Visual C++

public:
ReplicationHostAddress^ RepMgrLocalSite
See Also

ReplicationConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, the Replication Manager will observe the strict "majority" rule in managing elections, even in a group with only 2 sites. This means the client in a 2-site group will be unable to take over as master if the original master fails or becomes disconnected. (See the Elections section in the Berkeley DB Reference Guide for more information.) Both sites in the replication group should have the same value for this parameter.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
## Syntax

### C#

```csharp
public bool Strict2Site
```

### Visual Basic (Declaration)

```vbnet
Public Strict2Site As Boolean
```

### Visual C++

```cpp
public:
    bool Strict2Site
```
See Also

ReplicationConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
ReplicationConfig...:::Transport Field

ReplicationConfig Class  See Also

The delegate used to transmit data using the replication application's communication infrastructure.

Namespace:  BerkeleyDB
Syntax

**C#**

```csharp
public ReplicationTransportDelegate Transport
```

**Visual Basic (Declaration)**

```vbnet
Public Transport As ReplicationTransportDelegate
```

**Visual C++**

```cpp
public: ReplicationTransportDelegate^ Transport
```
See Also

ReplicationConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, master leases will be used for this site (defaults to false).

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll)  
**Version:** 4.8.24.0
Syntax

C#

public bool UseMasterLeases

Visual Basic (Declaration)

Public UseMasterLeases As Boolean

Visual C++

public:
bool UseMasterLeases
Remarks

Configuring this option may result in a LeaseExpiredException when attempting to read entries from a database after the site's master lease has expired.
See Also

ReplicationConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `ReplicationConfig` type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AddRemoteSite</td>
<td>Add a new replication site to the replication manager's list of known sites. It is not necessary for all sites in a replication group to know about all other sites in the group. Set the clock skew ratio among replication group members based on the fastest and slowest measurements among the group for use with master leases. Determines whether the specified Object is equal to the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Clockskew</td>
<td>Set the clock skew ratio among replication group members based on the fastest and slowest measurements among the group for use with master leases.</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified Object is equal to the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as a hash function for a particular type. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>RetransmissionRequest</td>
<td>Set a threshold for the minimum and maximum time that a client waits before requesting retransmission of a missing message. Returns a String that represents the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Set a byte-count limit on the amount of data that will be transmitted from a site in response to a single message processed by RepProcessMessage(DatabaseEntry, DatabaseEntry, Int32). The limit is not a hard limit, and the record that exceeds the limit is the last record to be sent.</td>
</tr>
<tr>
<td>TransmitLimit</td>
<td></td>
</tr>
</tbody>
</table>
See Also

ReplicationConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Add a new replication site to the replication manager's list of known sites. It is not necessary for all sites in a replication group to know about all other sites in the group.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
## Syntax

### C#

```csharp
public void AddRemoteSite(
    ReplicationHostAddress host,
    bool isPeer
)
```

### Visual Basic (Declaration)

```vbnet
Public Sub AddRemoteSite (_
    host As ReplicationHostAddress, _
    isPeer As Boolean _
)
```

### Visual C++

```cpp
public:
void AddRemoteSite(
    ReplicationHostAddress^ host,
    bool isPeer
)
```

### Parameters

**host**
- **Type:** `BerkeleyDB::ReplicationHostAddress`
- The remote site's address

**isPeer**
- **Type:** `System::Boolean`
- If true, configure client-to-client synchronization with the specified remote site.
Remarks

Currently, the replication manager framework only supports a single client peer, and the last specified peer is used.
See Also

ReplicationConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Set the clock skew ratio among replication group members based on the fastest and slowest measurements among the group for use with master leases.

Namespace: BerkeleyDB
Syntax

C#

```csharp
public void Clockskew(
    uint fast,
    uint slow
)
```

Visual Basic (Declaration)

```vbnet
Public Sub Clockskew (_
    fast As UInteger, _
    slow As UInteger _
)
```

Visual C++

```cpp
public:
void Clockskew(
    unsigned int fast,
    unsigned int slow
)
```

Parameters

fast
Type: `System::::UInt32`  
The value, relative to slow, of the fastest clock in the group of sites.

slow
Type: `System::::UInt32`  
The value of the slowest clock in the group of sites.
Remarks

Calling this method is optional, the default values for clock skew assume no skew. The user must also configure leases via UseMasterLeases. Additionally, the user must also set the master lease timeout via LeaseTimeout and the number of sites in the replication group via NSites. These settings may be configured in any order. For a description of the clock skew values, see Clock skew in the Berkeley DB Programmer's Reference Guide. For a description of master leases, see Master leases in the Berkeley DB Programmer's Reference Guide.

These arguments can be used to express either raw measurements of a clock timing experiment or a percentage across machines. For instance a group of sites have a 2% variance, then fast should be set to 102, and slow should be set to 100. Or, for a 0.03% difference, you can use 10003 and 10000 respectively.
See Also

ReplicationConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
ReplicationConfig...::RetransmissionRequest Method

See Also

Set a threshold for the minimum and maximum time that a client waits before requesting retransmission of a missing message.

Namespace: BerkeleyDB
Syntax

C#

public void RetransmissionRequest(
    uint min,
    uint max
)

Visual Basic (Declaration)

Public Sub RetransmissionRequest (_
    min As UInteger, _
    max As UInteger _
)

Visual C++

public:
void RetransmissionRequest( 
    unsigned int min, 
    unsigned int max
)

Parameters

min
Type: System::::UInt32
The minimum number of microseconds a client waits before requesting retransmission.

max
Type: System::::UInt32
The maximum number of microseconds a client waits before requesting retransmission.
Remarks

If the client detects a gap in the sequence of incoming log records or database pages, Berkeley DB will wait for at least min micro seconds before requesting retransmission of the missing record. Berkeley DB will double that amount before requesting the same missing record again, and so on, up to a maximum threshold of max microseconds.

These values are thresholds only. Since Berkeley DB has no thread available in the library as a timer, the threshold is only checked when a thread enters the Berkeley DB library to process an incoming replication message. Any amount of time may have passed since the last message arrived and Berkeley DB only checks whether the amount of time since a request was made is beyond the threshold value or not.

By default the minimum is 40000 and the maximum is 1280000 (1.28 seconds). These defaults are fairly arbitrary and the application likely needs to adjust these. The values should be based on expected load and performance characteristics of the master and client host platforms and transport infrastructure as well as round-trip message time.
See Also

ReplicationConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
ReplicationConfig Class  See Also

Set a byte-count limit on the amount of data that will be transmitted from a site in response to a single message processed by RepProcessMessage(DatabaseEntry, DatabaseEntry, Int32). The limit is not a hard limit, and the record that exceeds the limit is the last record to be sent.

Namespace: BerkeleyDB
Syntax

C#

public void TransmitLimit(
    uint GBytes,
    uint Bytes
)

Visual Basic (Declaration)

Public Sub TransmitLimit (_
    GBytes As UInteger, _
    Bytes As UInteger _
)

Visual C++

public:
void TransmitLimit(
    unsigned int GBytes,
    unsigned int Bytes
)

Parameters

GBytes
Type: System::::UInt32
The number of gigabytes which, when added to Bytes, specifies the maximum number of bytes that will be sent in a single call to RepProcessMessage(DatabaseEntry, DatabaseEntry, Int32).

Bytes
Type: System::::UInt32
The number of bytes which, when added to GBytes, specifies the maximum number of bytes that will be sent in a single call to RepProcessMessage(DatabaseEntry, DatabaseEntry, Int32).
Remarks

Record transmission throttling is turned on by default with a limit of 10MB.
If both GBytes and Bytes are zero, then the transmission limit is turned off.
See Also

ReplicationConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
ReplicationConfig Class  

The ReplicationConfig type exposes the following members.
### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AckTimeout</td>
<td>The amount of time the replication manager's transport function waits to collect enough acknowledgments from replication group clients, before giving up and returning a failure indication. The default wait time is 1 second.</td>
</tr>
<tr>
<td>CheckpointDelay</td>
<td>The amount of time a master site will delay between completing a checkpoint and writing a checkpoint record into the log.</td>
</tr>
<tr>
<td>ClockskewFast</td>
<td>The value, relative to ClockskewSlow, of the fastest clock in the group of sites.</td>
</tr>
<tr>
<td>ClockskewSlow</td>
<td>The value of the slowest clock in the group of sites.</td>
</tr>
<tr>
<td>ConnectionRetry</td>
<td>The amount of time the replication manager will wait before trying to re-establish a connection to another site after a communication failure. The default wait time is 30 seconds.</td>
</tr>
<tr>
<td>ElectionRetry</td>
<td>Configure the amount of time the replication manager will wait before retrying a failed election. The default wait time is 10 seconds.</td>
</tr>
<tr>
<td>ElectionTimeout</td>
<td>The timeout period for an election. The default timeout is 2 seconds.</td>
</tr>
<tr>
<td>FullElectionTimeout</td>
<td>An optional configuration timeout period to wait for full election participation the first time the replication group finds a master. By default this option is turned off and normal election timeouts are used. (See the Elections section in the Berkeley DB Reference Guide for more information.)</td>
</tr>
</tbody>
</table>
HeartbeatMonitor

The amount of time the replication manager, running at a client site, waits for some message activity on the connection from the master (heartbeats or other messages) before concluding that the connection has been lost. When 0 (the default), no monitoring is performed.

HeartbeatSend

The frequency at which the replication manager, running at a master site, broadcasts a heartbeat message in an otherwise idle system. When 0 (the default), no heartbeat messages will be sent.

LeaseTimeout

The amount of time a client grants its master lease to a master. When using master leases all sites in a replication group must use the same lease timeout value. There is no default value. If leases are desired, this method must be called prior to calling RepStartClient() or RepStartMaster().

NSites

The total number of sites in the replication group.

Priority

The database environment's priority in replication group elections. A special value of 0 indicates that this environment cannot be a replication group master. If not configured, then a default value of 100 is used.

RetransmissionRequestMax

The maximum number of microseconds a client waits before requesting retransmission.

RetransmissionRequestMin

The minimum number of microseconds a client waits before requesting retransmission.

The bytes component of the byte-count limit on the amount of data that will be...
TransmitLimitBytes transmitted from a site in response to a single message processed by RepProcessMessage(DatabaseEntry, DatabaseEntry, Int32).

The gigabytes component of the byte-count limit on the amount of data that will be transmitted from a site in response to a single message processed by RepProcessMessage(DatabaseEntry, DatabaseEntry, Int32).
See Also

ReplicationConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
ReplicationConfig.AckTimeout Property

The amount of time the replication manager's transport function waits to collect enough acknowledgments from replication group clients, before giving up and returning a failure indication. The default wait time is 1 second.

Namespace: BerkeleyDB
**Syntax**

**C#**

```csharp
public uint AckTimeout { get; set; }
```

**Visual Basic (Declaration)**

```vbnet
Public Property AckTimeout As UInteger
```

**Visual C++**

```cpp
public:
property unsigned int AckTimeout {
    unsigned int get ();
    void set (unsigned int value);
}
```
See Also

ReplicationConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The amount of time a master site will delay between completing a checkpoint and writing a checkpoint record into the log.

Namespace: BerkeleyDB
Syntax

C#

public uint CheckpointDelay { get; set; }

Visual Basic (Declaration)

Public Property CheckpointDelay As UInteger

Visual C++

public:
property unsigned int CheckpointDelay {
    unsigned int get ();
    void set (unsigned int value);
}
Remarks

This delay allows clients to complete their own checkpoints before the master requires completion of them. The default is 30 seconds. If all databases in the environment, and the environment's transaction log, are configured to reside in memory (never preserved to disk), then, although checkpoints are still necessary, the delay is not useful and should be set to 0.
See Also

ReplicationConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The value, relative to \texttt{ClockskewSlow}, of the fastest clock in the group of sites.

\textbf{Namespace: }\texttt{BerkeleyDB}
\textbf{Assembly: }\texttt{libdb_dotnet48} (in \texttt{libdb_dotnet48.dll}) Version: 4.8.24.0
Syntax

C#

public uint ClockskewFast { get; }

Visual Basic (Declaration)

Public ReadOnly Property ClockskewFast AsUInteger

Visual C++

public:
property unsigned int ClockskewFast {
    unsigned int get ();
}
See Also

ReplicationConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The value of the slowest clock in the group of sites.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public uint ClockskewSlow { get; }

Visual Basic (Declaration)

Public ReadOnly Property ClockskewSlow AsUInteger

Visual C++

public:
  property unsigned int ClockskewSlow {
    unsigned int get ();
  }
See Also

ReplicationConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The amount of time the replication manager will wait before trying to re-establish a connection to another site after a communication failure. The default wait time is 30 seconds.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
**Syntax**

**C#**

public **uint** ConnectionRetry { get; set; }

**Visual Basic (Declaration)**

Public Property ConnectionRetry As **UInteger**

**Visual C++**

public:
property **unsigned int** ConnectionRetry {
    unsigned int get ();
    void set (unsigned int value);
}

See Also

ReplicationConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Configure the amount of time the replication manager will wait before retrying a failed election. The default wait time is 10 seconds.

**Namespace:** BerkeleyDB
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public uint ElectionRetry { get; set; }

Visual Basic (Declaration)

Public Property ElectionRetry As UInteger

Visual C++

public:
property unsigned int ElectionRetry {
    unsigned int get ();
    void set (unsigned int value);
}
See Also

ReplicationConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
ReplicationConfig::ElectionTimeout Property

The timeout period for an election. The default timeout is 2 seconds.

Namespace:  BerkeleyDB
Syntax

C#

public uint ElectionTimeout { get; set; }

Visual Basic (Declaration)

Public Property ElectionTimeout As UInteger

Visual C++

public:
property unsigned int ElectionTimeout {
    unsigned int get ();
    void set (unsigned int value);
}
See Also

ReplicationConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
An optional configuration timeout period to wait for full election participation the first time the replication group finds a master. By default this option is turned off and normal election timeouts are used. (See the Elections section in the Berkeley DB Reference Guide for more information.)

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll)  
**Version:** 4.8.24.0
Syntax

C#

public uint FullElectionTimeout { get; set; }

Visual Basic (Declaration)

Public Property FullElectionTimeout AsUInteger

Visual C++

public:
property unsigned int FullElectionTimeout {
    unsigned int get ();
    void set (unsigned int value);
}
See Also

ReplicationConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The amount of time the replication manager, running at a client site, waits for some message activity on the connection from the master (heartbeats or other messages) before concluding that the connection has been lost. When 0 (the default), no monitoring is performed.

Namespace: BerkeleyDB
Syntax

C#

```csharp
public uint HeartbeatMonitor { get; set; }
```

Visual Basic (Declaration)

```vbnet
Public Property HeartbeatMonitor AsUInteger
```

Visual C++

```cpp
public:
property unsigned int HeartbeatMonitor {
    unsigned int get ();
    void set (unsigned int value);
}
```
See Also

ReplicationConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
ReplicationConfig::HeartbeatSend Property

The frequency at which the replication manager, running at a master site, broadcasts a heartbeat message in an otherwise idle system. When 0 (the default), no heartbeat messages will be sent.

Namespace: BerkeleyDB
Syntax

C#

public uint HeartbeatSend { get; set; }

Visual Basic (Declaration)

Public Property HeartbeatSend As UInteger

Visual C++

public:
property unsigned int HeartbeatSend {
    unsigned int get ();
    void set (unsigned int value);
}
See Also

ReplicationConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
ReplicationConfig.LeaseTimeout Property

ReplicationConfig Class  See Also

The amount of time a client grants its master lease to a master. When using master leases all sites in a replication group must use the same lease timeout value. There is no default value. If leases are desired, this method must be called prior to calling RepStartClient() or RepStartMaster().

Namespace: BerkeleyDB
Syntax

C#

public uint LeaseTimeout { get; set; }

Visual Basic (Declaration)

Public Property LeaseTimeout As UInteger

Visual C++

public:
property unsigned int LeaseTimeout {
    unsigned int get ();
    void set (unsigned int value);
}
See Also

ReplicationConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
ReplicationConfig::NSites Property

The total number of sites in the replication group.

Namespace: BerkeleyDB
Syntax

C#

public uint NSites { get; set; }

Visual Basic (Declaration)

Public Property NSites As UInteger

Visual C++

public:
property unsigned int NSites {
    unsigned int get();
    void set (unsigned int value);
}
Remarks

This setting is typically used by applications which use the Berkeley DB library "replication manager" support. (However, see also RepHoldElection(), the description of the nsites parameter.)
See Also

- ReplicationConfig Class
- BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The database environment's priority in replication group elections. A special value of 0 indicates that this environment cannot be a replication group master. If not configured, then a default value of 100 is used.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public uint Priority { get; set; }

Visual Basic (Declaration)

Public Property Priority AsUInteger

Visual C++

public:
property unsigned int Priority {
    unsigned int get();
    void set (unsigned int value);
}

See Also

ReplicationConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
ReplicationConfig::RetransmissionRequestMax Property

The maximum number of microseconds a client waits before requesting retransmission.

Namespace: BerkeleyDB
Syntax

**C#**

```csharp
public uint RetransmissionRequestMax { get; }
```

**Visual Basic (Declaration)**

```vbnet
Public ReadOnly Property RetransmissionRequestMax As UInteger
```

**Visual C++**

```cpp
public:
property unsigned int RetransmissionRequestMax {
    unsigned int get ();
}
```
See Also

ReplicationConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
ReplicationConfig.ReTransmissionRequestMin Property

The minimum number of microseconds a client waits before requesting retransmission.

Namespace: BerkeleyDB
Syntax

C#

public uint RetransmissionRequestMin { get; }

Visual Basic (Declaration)

Public ReadOnly Property RetransmissionRequestMin AsUInteger

Visual C++

public:
property unsigned int RetransmissionRequestMin {
    unsigned int get ();
}

See Also

ReplicationConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
ReplicationConfig::TransmitLimitBytes Property

The bytes component of the byte-count limit on the amount of data that will be transmitted from a site in response to a single message processed by RepProcessMessage(DatabaseEntry, DatabaseEntry, Int32).

Namespace: BerkeleyDB
Syntax

C#

public uint TransmitLimitBytes { get; }

Visual Basic (Declaration)

Public ReadOnly Property TransmitLimitBytes As UInteger

Visual C++

public:
property unsigned int TransmitLimitBytes {
    unsigned int get ();
}
See Also

ReplicationConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
ReplicationConfig...:.TransmitLimitGBytes Property

ReplicationConfig Class  See Also

The gigabytes component of the byte-count limit on the amount of data that will be transmitted from a site in response to a single message processed by RepProcessMessage(DatabaseEntry, DatabaseEntry, Int32).

Namespace: BerkeleyDB
Syntax

C#

public uint TransmitLimitGBytes { get; }

Visual Basic (Declaration)

Public ReadOnly Property TransmitLimitGBytes AsUInteger

Visual C++

public:
property unsigned int TransmitLimitGBytes {
    unsigned int get ();
}

See Also

ReplicationConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
A class representing the address of a replication site used by Berkeley DB HA.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
## Syntax

**C#**

public class ReplicationHostAddress

**Visual Basic (Declaration)**

Public Class ReplicationHostAddress

**Visual C++**

public ref class ReplicationHostAddress
Inheritance Hierarchy

System..:::Object
BerkeleyDB..:::ReplicationHostAddress
See Also

ReplicationHostAddress Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `ReplicationHostAddress` type exposes the following members.
### Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ReplicationHostAddress</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="#">Object</a> is equal to the current <strong>Object</strong>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <strong>Type</strong> of the current instance. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a String that represents the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
</tbody>
</table>
## Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Host</strong></td>
<td>The site's host identification string, generally a TCP/IP host name.</td>
</tr>
<tr>
<td><strong>Port</strong></td>
<td>The port number on which the site is receiving.</td>
</tr>
</tbody>
</table>
See Also

ReplicationHostAddress Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation

ReplicationHostAddress Constructor

ReplicationHostAddress Class   See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ReplicationHostAddress()()</code></td>
<td>Instantiate a new, empty address</td>
</tr>
<tr>
<td><code>ReplicationHostAddress(String)</code></td>
<td>Instantiate a new address, parsing the host and port from the given string</td>
</tr>
<tr>
<td><code>ReplicationHostAddress(String, Uint32)</code></td>
<td>Instantiate a new address</td>
</tr>
</tbody>
</table>
See Also

ReplicationHostAddress Class
ReplicationHostAddress Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Instantiate a new, empty address

**Namespace:**  [BerkeleyDB](https://berkeleydb.com)
**Assembly:**  [libdb_dotnet48](https://github.com/ak建立了BerkeleyDB) (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public ReplicationHostAddress()

Visual Basic (Declaration)

Public Sub New

Visual C++

public:
ReplicationHostAddress()
See Also

ReplicationHostAddress Class
ReplicationHostAddress Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
ReplicationHostAddress Constructor (String)

ReplicationHostAddress Class  See Also

Instantiate a new address, parsing the host and port from the given string

Namespace: BerkeleyDB
Syntax

C#

```csharp
public ReplicationHostAddress(
    string HostAndPort
)
```

Visual Basic (Declaration)

```vbnet
Public Sub New ( _
    HostAndPort As String _
)
```

Visual C++

```cpp
public:
ReplicationHostAddress(
    String^ HostAndPort
)
```

Parameters

HostAndPort

Type: System::String

A string in host:port format
See Also

ReplicationHostAddress Class
ReplicationHostAddress Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Instantiate a new address

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public ReplicationHostAddress(
    string Host,
    uint Port
)

Visual Basic (Declaration)

Public Sub New (_
    Host As String, _
    Port As UInteger _
)

Visual C++

public:
ReplicationHostAddress(
    String^ Host,
    unsigned int Port
)

Parameters

Host
Type: System::::String
The site's host identification string

Port
Type: System::::UInt32
The port number on which the site is receiving.
See Also

ReplicationHostAddress Class
ReplicationHostAddress Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `ReplicationHostAddress` type exposes the following members.
## Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host</td>
<td>The site's host identification string, generally a TCP/IP host name.</td>
</tr>
<tr>
<td>Port</td>
<td>The port number on which the site is receiving.</td>
</tr>
</tbody>
</table>
See Also

ReplicationHostAddress Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
ReplicationHostAddress Class  See Also

The site's host identification string, generally a TCP/IP host name.

Namespace:  BerkeleyDB
**Syntax**

**C#**

public `string` Host

**Visual Basic (Declaration)**

Public Host As `String`

**Visual C++**

public:
`String` Host
See Also

ReplicationHostAddress Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
ReplicationHostAddress Port Field

The port number on which the site is receiving.

Namespace: BerkeleyDB
Syntax

C#

public uint Port

Visual Basic (Declaration)

Public Port AsUInteger

Visual C++

public:
unsigned int Port
See Also

ReplicationHostAddress Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation

ReplicationHostAddress Methods

ReplicationHostAddress Class  See Also

The **ReplicationHostAddress** type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified Object is equal to the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a String that represents the current Object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>
See Also

ReplicationHostAddress Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Statistical information about the replication subsystem

Namespace: BerkeleyDB
Syntax

C#
public class ReplicationStats

Visual Basic (Declaration)
Public Class ReplicationStats

Visual C++
public ref class ReplicationStats
Inheritance Hierarchy

System..:::Object
BerkeleyDB..:::ReplicationStats
See Also

ReplicationStats Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The ReplicationStats type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="#">Object</a>)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a>)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a <a href="#">String</a> that represents the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>)</td>
</tr>
</tbody>
</table>
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AppliedTransactions</td>
<td># of transactions applied.</td>
</tr>
<tr>
<td>AwaitedLSN</td>
<td>LSN we're awaiting, if any.</td>
</tr>
<tr>
<td>AwaitedPage</td>
<td>pg we're awaiting, if any.</td>
</tr>
<tr>
<td>BadGenerationMessages</td>
<td>Messages with a bad generation #.</td>
</tr>
<tr>
<td>BulkBufferFills</td>
<td>Bulk buffer fills.</td>
</tr>
<tr>
<td>BulkBufferOverflows</td>
<td>Bulk buffer overflows.</td>
</tr>
<tr>
<td>BulkBufferTransfers</td>
<td>Transfers of bulk buffers.</td>
</tr>
<tr>
<td>BulkRecordsStored</td>
<td>Bulk records stored.</td>
</tr>
<tr>
<td>ClientServiceRequests</td>
<td>Number of client service requests received by this client.</td>
</tr>
<tr>
<td>ClientServiceRequestsMissing</td>
<td>Number of client service requests missing on this client.</td>
</tr>
<tr>
<td>ClientStartupComplete</td>
<td>Site completed client sync-up.</td>
</tr>
<tr>
<td>CurrentElectionGenerationNumber</td>
<td>Current election gen number.</td>
</tr>
<tr>
<td>CurrentGenerationNumber</td>
<td>Current generation number.</td>
</tr>
<tr>
<td>CurrentQueuedLogRecords</td>
<td>Log records currently queued.</td>
</tr>
<tr>
<td>CurrentWinner</td>
<td>Current front-runner.</td>
</tr>
<tr>
<td>CurrentWinnerMaxLSN</td>
<td>Max. LSN of current winner.</td>
</tr>
<tr>
<td>DuplicateLogRecords</td>
<td>Log records received multiply.</td>
</tr>
<tr>
<td>DuplicatePages</td>
<td>Pages received multiply.</td>
</tr>
<tr>
<td>DupMasters</td>
<td># of times a duplicate master condition was detected.</td>
</tr>
<tr>
<td>ElectionGenerationNumber</td>
<td>Election generation number.</td>
</tr>
<tr>
<td>ElectionPriority</td>
<td>Current election priority.</td>
</tr>
<tr>
<td>Elections</td>
<td># of elections held.</td>
</tr>
<tr>
<td>ElectionStatus</td>
<td>Current election status.</td>
</tr>
<tr>
<td>ElectionsWon</td>
<td># of elections won by this site.</td>
</tr>
<tr>
<td>ElectionTiebreaker</td>
<td>Election tiebreaker value.</td>
</tr>
<tr>
<td>ElectionTimeSec</td>
<td>Last election time seconds.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ElectionTimeUSEc</td>
<td>Last election time useconds.</td>
</tr>
<tr>
<td>EnvID</td>
<td>Current environment ID.</td>
</tr>
<tr>
<td>EnvPriority</td>
<td>Current environment priority.</td>
</tr>
<tr>
<td>FailedMessageSends</td>
<td># of failed message sends.</td>
</tr>
<tr>
<td>ForcedRerequests</td>
<td>Number of forced rerequests.</td>
</tr>
<tr>
<td>IgnoredMessages</td>
<td>Messages ignored because this site was a client in recovery.</td>
</tr>
<tr>
<td>MasterChanges</td>
<td># of times we've switched masters.</td>
</tr>
<tr>
<td>MasterEnvID</td>
<td>Env. ID of the current master.</td>
</tr>
<tr>
<td>MaxLeaseSec</td>
<td>Maximum lease timestamp seconds.</td>
</tr>
<tr>
<td>MaxLeaseUSEc</td>
<td>Maximum lease timestamp useconds.</td>
</tr>
<tr>
<td>MaxPermanentLSN</td>
<td>Maximum permanent LSN.</td>
</tr>
<tr>
<td>MaxQueuedLogRecords</td>
<td>Max. log records queued at once.</td>
</tr>
<tr>
<td>MessagesSent</td>
<td># of successful message sends.</td>
</tr>
<tr>
<td>MissedLogRecords</td>
<td>Log recs. missed and requested.</td>
</tr>
<tr>
<td>MissedPages</td>
<td>Pages missed and requested.</td>
</tr>
<tr>
<td>NewSiteMessages</td>
<td># of NEWSITE msgs. received.</td>
</tr>
<tr>
<td>NextLSN</td>
<td>Next LSN to use or expect.</td>
</tr>
<tr>
<td>NextPage</td>
<td>Next pg we expect.</td>
</tr>
<tr>
<td>Outdated</td>
<td># of times we detected and returned an OUTDATED condition.</td>
</tr>
<tr>
<td>QueuedLogRecords</td>
<td>Total # of log recs. ever queued.</td>
</tr>
<tr>
<td>ReceivedLogRecords</td>
<td>Log records received and put.</td>
</tr>
<tr>
<td>ReceivedMessages</td>
<td>Messages received and processed.</td>
</tr>
<tr>
<td>ReceivedPages</td>
<td>Pages received and stored.</td>
</tr>
<tr>
<td>RegisteredSites</td>
<td># of &quot;registered voters&quot;.</td>
</tr>
<tr>
<td>RegisteredSitesNeeded</td>
<td># of &quot;registered voters&quot; needed.</td>
</tr>
<tr>
<td>Sites</td>
<td>Current number of sites we will assume during elections.</td>
</tr>
<tr>
<td>StartSyncMessagesDelayed</td>
<td># of STARTSYNC msgs delayed.</td>
</tr>
<tr>
<td>Status</td>
<td>Current replication status.</td>
</tr>
<tr>
<td>Throttled</td>
<td># of times we were throttled.</td>
</tr>
</tbody>
</table>
Votes

Votes received in this round.
See Also

ReplicationStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `ReplicationStats` type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="#">Object</a> is equal to the current <strong>Object</strong>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <strong>Type</strong> of the current instance. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a <a href="#">String</a> that represents the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
</tbody>
</table>
See Also

ReplicationStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `ReplicationStats` type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AppliedTransactions</td>
<td># of transactions applied.</td>
</tr>
<tr>
<td>AwaitedLSN</td>
<td>LSN we're awaiting, if any.</td>
</tr>
<tr>
<td>AwaitedPage</td>
<td>pg we're awaiting, if any.</td>
</tr>
<tr>
<td>BadGenerationMessages</td>
<td>Messages with a bad generation #.</td>
</tr>
<tr>
<td>BulkBufferFills</td>
<td>Bulk buffer fills.</td>
</tr>
<tr>
<td>BulkBufferOverflows</td>
<td>Bulk buffer overflows.</td>
</tr>
<tr>
<td>BulkBufferTransfers</td>
<td>Transfers of bulk buffers.</td>
</tr>
<tr>
<td>BulkRecordsStored</td>
<td>Bulk records stored.</td>
</tr>
<tr>
<td>ClientServiceRequests</td>
<td>Number of client service requests received by this client.</td>
</tr>
<tr>
<td>ClientServiceRequestsMissing</td>
<td>Number of client service requests missing on this client.</td>
</tr>
<tr>
<td>ClientStartupComplete</td>
<td>Site completed client sync-up.</td>
</tr>
<tr>
<td>CurrentElectionGenerationName</td>
<td>Current election gen number.</td>
</tr>
<tr>
<td>CurrentGenerationNumber</td>
<td>Current generation number.</td>
</tr>
<tr>
<td>CurrentQueuedLogRecords</td>
<td>Log records currently queued.</td>
</tr>
<tr>
<td>CurrentWinner</td>
<td>Current front-runner.</td>
</tr>
<tr>
<td>CurrentWinnerMaxLSN</td>
<td>Max. LSN of current winner.</td>
</tr>
<tr>
<td>DuplicateLogRecords</td>
<td>Log records received multiply.</td>
</tr>
<tr>
<td>DuplicatePages</td>
<td>Pages received multiply.</td>
</tr>
<tr>
<td>DupMasters</td>
<td># of times a duplicate master condition was detected.</td>
</tr>
<tr>
<td>ElectionGenerationNumber</td>
<td>Election generation number.</td>
</tr>
<tr>
<td>ElectionPriority</td>
<td>Current election priority.</td>
</tr>
<tr>
<td>Elections</td>
<td># of elections held.</td>
</tr>
<tr>
<td>ElectionStatus</td>
<td>Current election status.</td>
</tr>
<tr>
<td>ElectionsWon</td>
<td># of elections won by this site.</td>
</tr>
<tr>
<td>ElectionTiebreaker</td>
<td>Election tiebreaker value.</td>
</tr>
<tr>
<td>ElectionTimeSec</td>
<td>Last election time seconds.</td>
</tr>
<tr>
<td>Metric</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>ElectionTimeUSEc</strong></td>
<td>Last election time useconds.</td>
</tr>
<tr>
<td><strong>EnvID</strong></td>
<td>Current environment ID.</td>
</tr>
<tr>
<td><strong>EnvPriority</strong></td>
<td>Current environment priority.</td>
</tr>
<tr>
<td><strong>FailedMessageSends</strong></td>
<td># of failed message sends.</td>
</tr>
<tr>
<td><strong>ForcedRerequests</strong></td>
<td>Number of forced rerequests.</td>
</tr>
<tr>
<td><strong>IgnoredMessages</strong></td>
<td>Messages ignored because this site was a client in recovery.</td>
</tr>
<tr>
<td><strong>MasterChanges</strong></td>
<td># of times we've switched masters.</td>
</tr>
<tr>
<td><strong>MasterEnvID</strong></td>
<td>Env. ID of the current master.</td>
</tr>
<tr>
<td><strong>MaxLeaseSec</strong></td>
<td>Maximum lease timestamp seconds.</td>
</tr>
<tr>
<td><strong>MaxLeaseUSEc</strong></td>
<td>Maximum lease timestamp useconds.</td>
</tr>
<tr>
<td><strong>MaxPermanentLSN</strong></td>
<td>Maximum permanent LSN.</td>
</tr>
<tr>
<td><strong>MaxQueuedLogRecords</strong></td>
<td>Max. log records queued at once.</td>
</tr>
<tr>
<td><strong>MessagesSent</strong></td>
<td># of successful message sends.</td>
</tr>
<tr>
<td><strong>MissedLogRecords</strong></td>
<td>Log recs. missed and requested.</td>
</tr>
<tr>
<td><strong>MissedPages</strong></td>
<td>Pages missed and requested.</td>
</tr>
<tr>
<td><strong>NewSiteMessages</strong></td>
<td># of NEWSITE msgs. received.</td>
</tr>
<tr>
<td><strong>NextLSN</strong></td>
<td>Next LSN to use or expect.</td>
</tr>
<tr>
<td><strong>NextPage</strong></td>
<td>Next pg we expect.</td>
</tr>
<tr>
<td><strong>Outdated</strong></td>
<td># of times we detected and returned an OUTDATED condition.</td>
</tr>
<tr>
<td><strong>QueuedLogRecords</strong></td>
<td>Total # of log recs. ever queued.</td>
</tr>
<tr>
<td><strong>ReceivedLogRecords</strong></td>
<td>Log records received and put.</td>
</tr>
<tr>
<td><strong>ReceivedMessages</strong></td>
<td>Messages received and processed.</td>
</tr>
<tr>
<td><strong>ReceivedPages</strong></td>
<td>Pages received and stored.</td>
</tr>
<tr>
<td><strong>RegisteredSites</strong></td>
<td># of &quot;registered voters&quot;.</td>
</tr>
<tr>
<td><strong>RegisteredSitesNeeded</strong></td>
<td># of &quot;registered voters&quot; needed.</td>
</tr>
<tr>
<td><strong>Sites</strong></td>
<td>Current number of sites we will assume during elections.</td>
</tr>
<tr>
<td><strong>StartSyncMessagesDelayed</strong></td>
<td># of STARTSYNC msgs delayed.</td>
</tr>
<tr>
<td><strong>Status</strong></td>
<td>Current replication status.</td>
</tr>
<tr>
<td><strong>Throttled</strong></td>
<td># of times we were throttled.</td>
</tr>
<tr>
<td><strong>Votes</strong></td>
<td>Votes received in this round.</td>
</tr>
</tbody>
</table>
See Also

ReplicationStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
ReplicationStats::AppliedTransactions Property

# of transactions applied.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public ulong AppliedTransactions { get; }

Visual Basic (Declaration)

Public ReadOnly Property AppliedTransactions As ULong

Visual C++

public:
property unsigned long long AppliedTransactions {
    unsigned long long get ();
}
See Also

ReplicationStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
ReplicationStats...:..AwaitedLSN Property

Namespace: BerkeleyDB
Syntax

C#

public LSN AwaitedLSN { get; }

Visual Basic (Declaration)

Public ReadOnly Property AwaitedLSN As LSN

Visual C++

public:
property LSN^ AwaitedLSN {
    LSN^ get ();
}
See Also

ReplicationStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
ReplicationStats...:::AwaitedPage Property

ReplicationStats Class  See Also

pg we're awaiting, if any.

Namespace:  BerkeleyDB
Syntax

C#

public uint AwaitedPage { get; }

Visual Basic (Declaration)

Public ReadOnly Property AwaitedPage As UInteger

Visual C++

public:
property unsigned int AwaitedPage {
    unsigned int get ();
}


See Also

ReplicationStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
ReplicationStats::BadGenerationMessages Property

Messages with a bad generation #.

Namespace: BerkeleyDB
Syntax

C#

public ulong BadGenerationMessages { get; }

Visual Basic (Declaration)

Public ReadOnly Property BadGenerationMessages As ULong

Visual C++

public:
property unsigned long long BadGenerationMessages {
    unsigned long long get ();
}

See Also

ReplicationStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
ReplicationStats.BulkBufferFills Property

Bulk buffer fills.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public ulong BulkBufferFills { get; }

Visual Basic (Declaration)

Public ReadOnly Property BulkBufferFills As ULong

Visual C++

public:
property unsigned long long BulkBufferFills {
    unsigned long long get ();
}
See Also

ReplicationStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Bulk buffer overflows.

**Namespace:**  [BerkeleyDB](https://docs.oracle.com/cd/E17414_01/bddev.51/E14890/)

**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public ulong BulkBufferOverflows { get; }

Visual Basic (Declaration)

Public ReadOnly Property BulkBufferOverflows As ULong

Visual C++

public:
property unsigned long long BulkBufferOverflows {
    unsigned long long get ();
}

See Also

**ReplicationStats Class**
**BerkeleyDB Namespace**

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
ReplicationStats::BulkBufferTransfers Property

ReplicationStats Class  See Also

Transfers of bulk buffers.

Namespace:  BerkeleyDB
Syntax

C#

public ulong BulkBufferTransfers { get; }

Visual Basic (Declaration)

Public ReadOnly Property BulkBufferTransfers As ULong

Visual C++

public:
property unsigned long long BulkBufferTransfers {
    unsigned long long get ();
}
See Also

ReplicationStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
ReplicationStats::BulkRecordsStored Property

Bulk records stored.

**Namespace:** BerkeleyDB

**Assembly:** `libdb_dotnet48` (in `libdb_dotnet48.dll`) Version: 4.8.24.0
Syntax

C#

public ulong BulkRecordsstored { get; }

Visual Basic (Declaration)

Public ReadOnly Property BulkRecordsStored As ULong

Visual C++

public:
property unsigned long long BulkRecordsStored {
    unsigned long long get ();
}

See Also

ReplicationStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
ReplicationStats.ClientServiceRequests Property

Number of client service requests received by this client.

Namespace: BerkeleyDB
### Syntax

**C#**

```csharp
public ulong ClientServiceRequests { get; }
```

**Visual Basic (Declaration)**

```vbnet
Public ReadOnly Property ClientServiceRequests As ULong
```

**Visual C++**

```cpp
public:
property unsigned long long ClientServiceRequests {
    unsigned long long get ();
}
```
See Also

ReplicationStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
ReplicationStats..:::ClientServiceRequestsMissing Property

RepliationStats Class  See Also

Number of client service requests missing on this client.

Namespace:  BerkeleyDB
### Syntax

**C#**

```csharp
public ulong ClientServiceRequestsMissing { get; }
```

**Visual Basic (Declaration)**

```vbnet
Public ReadOnly Property ClientServiceRequestsMissing As ULong
```

**Visual C++**

```cpp
public:
    property unsigned long long ClientServiceRequestsMissing {
        unsigned long long get ();
    }
```
See Also

ReplicationStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Site completed client sync-up.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

```csharp
public bool ClientStartupComplete { get; }
```

Visual Basic (Declaration)

```vbnet
Public ReadOnly Property ClientStartupComplete As Boolean
```

Visual C++

```c++
public:
property bool ClientStartupComplete {
    bool get ();
}
```
See Also

*ReplicationStats Class*
*BerkeleyDB Namespace*

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
ReplicationStats...:::CurrentElectionGenerationNumber Property

ReplicationStats Class  See Also

Current election gen number.

Namespace:  BerkeleyDB
**Syntax**

**C#**

```csharp
public uint CurrentElectionGenerationNumber { get; }
```

**Visual Basic (Declaration)**

```vbnet
Public ReadOnly Property CurrentElectionGenerationNumber As UInteger
```

**Visual C++**

```cpp
public:
property unsigned int CurrentElectionGenerationNumber {
    unsigned int get ()
}
```
See Also

ReplicationStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Current generation number.

Namespace: BerkeleyDB
**Syntax**

**C#**

```csharp
public uint CurrentGenerationNumber { get; }
```

**Visual Basic (Declaration)**

```vbnet
Public ReadOnly Property CurrentGenerationNumber AsUInteger
```

**Visual C++**

```c++
public:
property unsigned int CurrentGenerationNumber {
    unsigned int get ();
}
```
See Also

ReplicationStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Log records currently queued.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public ulong CurrentQueuedLogRecords { get; }

Visual Basic (Declaration)

Public ReadOnly Property CurrentQueuedLogRecords As ULong

Visual C++

public:
property unsigned long long CurrentQueuedLogRecords {
    unsigned long long get ();
}

See Also

ReplicationStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Current front-runner.

**Namespace:** [BerkeleyDB](#)  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

```csharp
public int CurrentWinner { get; }
```

Visual Basic (Declaration)

```vbnet
Public ReadOnly Property CurrentWinner As Integer
```

Visual C++

```cpp
public:
    property int CurrentWinner {
        int get ();
    }
```
See Also

ReplicationStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
ReplicationStats...:::CurrentWinnerMaxLSN Property

Max. LSN of current winner.

Namespace: **BerkeleyDB**
Syntax

C#

public LSN CurrentWinnerMaxLSN { get; }

Visual Basic (Declaration)

Public ReadOnly Property CurrentWinnerMaxLSN As LSN

Visual C++

public:
property LSN^ CurrentWinnerMaxLSN {
    LSN^ get ();
}

See Also

ReplicationStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
ReplicationStats...:::DuplicateLogRecords Property

ReplicationStats Class  See Also

Log records received multiply.

Namespace:  BerkeleyDB
Syntax

**C#**

```csharp
public ulong DuplicateLogRecords { get; }
```

**Visual Basic (Declaration)**

```vbnet
Public Readonly Property DuplicateLogRecords As ULong
```

**Visual C++**

```cpp
public:
property unsigned long long DuplicateLogRecords {
    unsigned long long get ();
}
```
See Also

ReplicationStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Pages received multiply.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public ulong DuplicatePages { get; }

Visual Basic (Declaration)

Public ReadOnly Property DuplicatePages As ULong

Visual C++

public:
property unsigned long long DuplicatePages {
    unsigned long long get ();
}
See Also

ReplicationStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
ReplicationStats...:::DupMasters Property

ReplicationStats Class  See Also

# of times a duplicate master condition was detected.

Namespace:  BerkeleyDB
## Syntax

**C#**

```csharp
public uint DupMasters { get; }
```

**Visual Basic (Declaration)**

Public ReadOnly Property DupMasters AsUInteger

**Visual C++**

```c++
public:
property unsigned int DupMasters {
    unsigned int get ();
}
```
See Also

ReplicationStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Election generation number.

Namespace:  BerkeleyDB
Syntax

C#

public uint ElectionGenerationNumber { get; }

Visual Basic (Declaration)

Public ReadOnly Property ElectionGenerationNumber AsUInteger

Visual C++

public:
property unsigned int ElectionGenerationNumber {
    unsigned int get ();
}
See Also

ReplicationStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
ReplicationStats...

ElectionPriority Property

ReplicationStats Class  See Also

Current election priority.

Namespace:  BerkeleyDB
Syntax

C#

public uint ElectionPriority { get; }

Visual Basic (Declaration)

Public ReadOnly Property ElectionPriority AsUInteger

Visual C++

public:
property unsigned int ElectionPriority {
    unsigned int get ();
}
See Also

ReplicationStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
ReplicationStats::Elections Property

Namespace: BerkeleyDB

# of elections held.
Syntax

C#

public ulong Elections { get; }

Visual Basic (Declaration)

Public ReadOnly Property Elections As ULong

Visual C++

public:
    property unsigned long long Elections {
        unsigned long long get ();
    }

See Also

ReplicationStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
ReplicationStats.ElectionStatus Property

Current election status.

Namespace: BerkeleyDB
### Syntax

**C#**

```csharp
public int ElectionStatus { get; }
```

**Visual Basic (Declaration)**

```vbnet
Public ReadOnly Property ElectionStatus As Integer
```

**Visual C++**

```cpp
public:
    property int ElectionStatus {
        int get ();
    }
```
See Also

ReplicationStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation

ReplicationStats::ElectionsWon Property

ReplicationStats Class  See Also

# of elections won by this site.

**Namespace:**  BerkeleyDB

**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public ulong ElectionsWon { get; }

Visual Basic (Declaration)

Public ReadOnly Property ElectionsWon As ULong

Visual C++

public:
property unsigned long long ElectionsWon {
    unsigned long long get ();
}
See Also

ReplicationStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Election tiebreaker value.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public uint ElectionTiebreaker { get; }

Visual Basic (Declaration)

Public ReadOnly Property ElectionTiebreaker As UInteger

Visual C++

public:
property unsigned int ElectionTiebreaker {
    unsigned int get ();
}

See Also

ReplicationStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
ReplicationStats.ElectionTimeSec Property

Last election time seconds.

Namespace: BerkeleyDB
Syntax

C#

public uint ElectionTimeSec { get; }

Visual Basic (Declaration)

Public ReadOnly Property ElectionTimeSec As UInteger

Visual C++

public:
property unsigned int ElectionTimeSec {
    unsigned int get ();
}

See Also

ReplicationStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
ReplicationStats...ElectionTimeUSec Property

ReplicationStats Class  See Also

Last election time useconds.

Namespace:  BerkeleyDB
Syntax

**C#**

```csharp
public uint ElectionTimeUSec { get; }
```

**Visual Basic (Declaration)**

```vbnet
Public Readonly Property ElectionTimeUSec AsUInteger
```

**Visual C++**

```cpp
public:
property unsigned int ElectionTimeUSec {
    unsigned int get ();
}
```
See Also

ReplicationStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
ReplicationStats::EnvID Property

Current environment ID.

**Namespace:** BerkeleyDB
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public int EnvID { get; }

Visual Basic (Declaration)

Public ReadOnly Property EnvID As Integer

Visual C++

public:
property int EnvID {
    int get ();
}

}
See Also

ReplicationStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Current environment priority.

**Namespace:** BerkeleyDB  
**Assembly:** `libdb_dotnet48` (in `libdb_dotnet48.dll`) Version: 4.8.24.0
Syntax

C#

public uint EnvPriority { get; }

Visual Basic (Declaration)

Public ReadOnly Property EnvPriority As UInteger

Visual C++

public:
property unsigned int EnvPriority {
    unsigned int get ();
}

See Also

ReplicationStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
ReplicationStats...:::FailedMessageSends Property

Namespace: BerkeleyDB
Syntax

C#

```csharp
public ulong FailedMessageSends { get; }
```

Visual Basic (Declaration)

```vbnet
Public ReadOnly Property FailedMessageSends As ULong
```

Visual C++

```c++
public:
    property unsigned long long FailedMessageSends {
        unsigned long long get ();
    }
```
See Also

ReplicationStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
ReplicationStats ForcRerrequests Property

Number of forced rerquests.

Namespace: BerkeleyDB
Syntax

C#

public ulong ForcedRerequests { get; }

Visual Basic (Declaration)

Public ReadOnly Property ForcedRerequests As ULong

Visual C++

public:
property unsigned long long ForcedRerequests {
       unsigned long long get ();
}
See Also

ReplicationStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Messages ignored because this site was a client in recovery.

**Namespace:**  BerkeleyDB  
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll)  
**Version:**  4.8.24.0
Syntax

C#

public ulong IgnoredMessages { get; }

Visual Basic (Declaration)

Public ReadOnly Property IgnoredMessages As ULong

Visual C++

public:
property unsigned long long IgnoredMessages {
    unsigned long long get ();
}


See Also

ReplicationStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
ReplicationStats Class  See Also

# of times we've switched masters.

Namespace:  BerkeleyDB
Syntax

C#

public ulong MasterChanges { get; }

Visual Basic (Declaration)

Public ReadOnly Property MasterChanges As ULong

Visual C++

public:
property unsigned long long MasterChanges {
     unsigned long long get ();
}
See Also

ReplicationStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
ReplicationStats...:::MasterEnvID Property

ReplicationStats Class  See Also

Env. ID of the current master.

Namespace:  BerkeleyDB
Syntax

C#

public int MasterEnvID { get; }

Visual Basic (Declaration)

Public ReadOnly Property MasterEnvID As Integer

Visual C++

public:
property int MasterEnvID {
    int get ();
}


See Also

ReplicationStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
ReplicationStats::MaxLeaseSec Property

Maximum lease timestamp seconds.

Namespace:  BerkeleyDB
Syntax

C#

public uint MaxLeaseSec { get; }

Visual Basic (Declaration)

Public ReadOnly Property MaxLeaseSec As UInteger

Visual C++

public:
property unsigned int MaxLeaseSec {
    unsigned int get ();
}

See Also

ReplicationStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Maximum lease timestamp useconds.

Namespace: BerkeleyDB
C#

public uint MaxLeaseUSec { get; }

Visual Basic (Declaration)

Public ReadOnly Property MaxLeaseUSec AsUInteger

Visual C++

public:
property unsigned int MaxLeaseUSec {
unsigned int get ();
}
See Also

ReplicationStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Maximum permanent LSN.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public LSN MaxPermanentLSN { get; }

Visual Basic (Declaration)

Public ReadOnly Property MaxPermanentLSN As LSN

Visual C++

public:
property LSN^ MaxPermanentLSN {
    LSN^ get ();
}


See Also

ReplicationStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
ReplicationStats...:::MaxQueuedLogRecords Property

ReplicationStats Class  See Also

Max. log records queued at once.

Namespace:  BerkeleyDB
Syntax

C#

public ulong MaxQueuedLogRecords { get; }

Visual Basic (Declaration)

Public ReadOnly Property MaxQueuedLogRecords As ULong

Visual C++

public:
property unsigned long long MaxQueuedLogRecords {
    unsigned long long get ();
}

See Also

ReplicationStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
ReplicationStats::MessagesSent Property

ReplicationStats Class  See Also

# of successful message sends.

Namespace: BerkeleyDB
Syntax

C#

public ulong MessagesSent { get; }

Visual Basic (Declaration)

Public ReadOnly Property MessagesSent As ULong

Visual C++

public:
property unsigned long long MessagesSent {
    unsigned long long get ();
}

See Also

ReplicationStats Class
BerkeleyDB_NAMESPACE

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
C# Visual Basic Visual C++
Berkeley DB .NET API Documentation
ReplicationStats:::MissedLogRecords Property

ReplicationStats Class  See Also

Log recs. missed and requested.

Namespace:  BerkeleyDB
Syntax

**C#**

```csharp
public ulong MissedLogRecords { get; }
```

**Visual Basic (Declaration)**

```
Public ReadOnly Property MissedLogRecords As ULong
```

**Visual C++**

```cpp
public:
    property unsigned long long MissedLogRecords {
        unsigned long long get ();
    }
```
See Also

ReplicationStats Class  
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Pages missed and requested.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public ulong MissedPages { get; }

Visual Basic (Declaration)

Public ReadOnly Property MissedPages As ULong

Visual C++

public:
property unsigned long long MissedPages {
    unsigned long long get ();
}

See Also

ReplicationStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
ReplicationStats.NewSiteMessages Property

# of NEWSITE msgs. received.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public ulong NewSiteMessages { get; }

Visual Basic (Declaration)

Public ReadOnly Property NewSiteMessages As ULong

Visual C++

public:
property unsigned long long NewSiteMessages {
    unsigned long long get ();
}


See Also

ReplicationStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
ReplicationStats..::.NextLSN Property

Next LSN to use or expect.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public LSN NextLSN { get; }

Visual Basic (Declaration)

PublicReadOnly Property NextLSN As LSN

Visual C++

public:
property LSN^ NextLSN {
    LSN^ get ();
}


See Also

ReplicationStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Next pg we expect.

Namespace:  BerkeleyDB
Syntax

C#

public uint NextPage { get; }

Visual Basic (Declaration)

Public ReadOnly Property NextPage AsUInteger

Visual C++

public:
property unsigned int NextPage {
    unsigned int get ();
}
See Also

ReplicationStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
ReplicationStats...:::Outdated Property

ReplicationStats Class  See Also

# of times we detected and returned an OUTDATED condition.

Namespace:  BerkeleyDB
Syntax

C#

```csharp
public ulong Outdated { get; }
```

Visual Basic (Declaration)

Public ReadOnly Property Outdated As ULong

Visual C++

```c++
public:
property unsigned long long Outdated {
    unsigned long long get ();
}
```
See Also

ReplicationStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
ReplicationStats::QueuedLogRecords Property

ReplicationStats Class  See Also

Total # of log recs. ever queued.

Namespace:  BerkeleyDB
**Syntax**

**C#**

```csharp
public ulong QueuedLogRecords { get; }
```

**Visual Basic (Declaration)**

```vbnet
Public ReadOnly Property QueuedLogRecords As ULong
```

**Visual C++**

```cpp
public:
property unsigned long long QueuedLogRecords {
    unsigned long long get ();
}
```
See Also

ReplicationStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
ReplicationStats::ReceivedLogRecords Property

Log records received and put.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

**C#**

```
public ulong ReceivedLogRecords { get; }
```

**Visual Basic (Declaration)**

```
Public ReadOnly Property ReceivedLogRecords As ULong
```

**Visual C++**

```
public:
property unsigned long long ReceivedLogRecords {
    unsigned long long get ();
}
```
See Also

ReplicationStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Messages received and processed.

Namespace: BerkeleyDB
Syntax

C#

public ulong ReceivedMessages { get; }

Visual Basic (Declaration)

Public ReadOnly Property ReceivedMessages As ULong

Visual C++

public:
    property unsigned long long ReceivedMessages {
        unsigned long long get ();
    }
See Also

ReplicationStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
ReplicationStats::ReceivedPages Property

Pages received and stored.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public ulong ReceivedPages { get; }

Visual Basic (Declaration)

Public ReadOnly Property ReceivedPages As ULong

Visual C++

public:
property unsigned long long ReceivedPages {
    unsigned long long get ();
}

See Also

ReplicationStats Class  
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
ReplicationStats::RegisteredSites Property

Namespace: BerkeleyDB

# of "registered voters".
Syntax

C#

public uint RegisteredSites { get; }

Visual Basic (Declaration)

Public Readonly Property RegisteredSites AsUInteger

Visual C++

public:
property unsigned int RegisteredSites {
    unsigned int get ();
}

See Also

ReplicationStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
# of "registered voters" needed.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public uint RegisteredSitesNeeded { get; }

Visual Basic (Declaration)

Public Readonly Property RegisteredSitesNeeded AsUInteger

Visual C++

public:
property unsigned int RegisteredSitesNeeded {
    unsigned int get ();
}

See Also

ReplicationStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Current number of sites we will assume during elections.

Namespace:  [BerkeleyDB](#)
Assembly:  [libdb_dotnet48](#) (in libdb_dotnet48.dll) Version: 4.8.24.0
**Syntax**

**C#**

```csharp
public uint Sites { get; }
```

**Visual Basic (Declaration)**

```vbnet
Public ReadOnly Property Sites AsUInteger
```

**Visual C++**

```cpp
public:
property unsigned int Sites {
 unsigned int get ();
}
```
See Also

- ReplicationStats Class
- BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation

ReplicationStats...:::StartSyncMessagesDelayed Property

ReplicationStats Class  See Also

# of STARTSYNC msgs delayed.

Namespace:  BerkeleyDB
Syntax

C#

public ulong StartSyncMessagesDelayed { get; }

Visual Basic (Declaration)

Public ReadOnly Property StartSyncMessagesDelayed As ULong

Visual C++

public:
property unsigned long long StartSyncMessagesDelayed {
    unsigned long long get ();
}
See Also

ReplicationStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Current replication status.

**Namespace:**  BerkeleyDB  
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public uint Status { get; }

Visual Basic (Declaration)

Public ReadOnly Property Status AsUInteger

Visual C++

public:
property unsigned int Status {
    unsigned int get ()
}

See Also

ReplicationStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
ReplicationStats Class  See Also

# of times we were throttled.

Namespace: BerkeleyDB
Syntax

C#

public ulong Throttled { get; }

Visual Basic (Declaration)

Public ReadOnly Property Throttled As ULong

Visual C++

public:
property unsigned long long Throttled {
    unsigned long long get ();
}


See Also

ReplicationStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
ReplicationStats::Votes Property

Votes received in this round.

Namespace: BerkeleyDB
## Syntax

**C#**

```csharp
public uint Votes { get; }
```

**Visual Basic (Declaration)**

```vbnet
Public ReadOnly Property Votes AsUInteger
```

**Visual C++**

```cpp
public:
property unsigned int Votes {
    unsigned int get ();
}
```
See Also

ReplicationStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The function used to transmit data using the replication application's communication infrastructure.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public delegate int ReplicationTransportDelegate(
    DatabaseEntry control,
    DatabaseEntry rec,
    LSN lsn,
    int envid,
    uint flags
)

Visual Basic (Declaration)

Public Delegate Function ReplicationTransportDelegate ( _
    control As DatabaseEntry, _
    rec As DatabaseEntry, _
    lsn As LSN, _
    envid As Integer, _
    flags As UInteger _
) As Integer

Visual C++

public delegate int ReplicationTransportDelegate(
    DatabaseEntry^ control,
    DatabaseEntry^ rec,
    LSN^ lsn,
    int envid,
    unsigned int flags
)

Parameters

control
Type: BerkeleyDB::DatabaseEntry
The first of the two data elements to be transmitted by the send function.

rec
Type: BerkeleyDB::DatabaseEntry
The second of the two data elements to be transmitted by the send function.

**lsn**
Type: BerkeleyDB::LSN
If the type of message to be sent has an LSN associated with it, then this is the LSN of the record being sent. This LSN can be used to determine that certain records have been processed successfully by clients.

**envid**
Type: System::Int32
A positive integer identifier that specifies the replication environment to which the message should be sent.

The special identifier DB_EID_BROADCAST indicates that a message should be broadcast to every environment in the replication group. The application may use a true broadcast protocol or may send the message in sequence to each machine with which it is in communication. In both cases, the sending site should not be asked to process the message.

The special identifier DB_EID_INVALID indicates an invalid environment ID. This may be used to initialize values that are subsequently checked for validity.

**flags**
Type: System::UInt32
XXX: TBD

**Return Value**

0 on success and non-zero on failure
See Also

BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
A class representing a replication site used by Replication Manager

**Namespace:**  BerkeleyDB
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

**C#**

```csharp
public class RepMgrSite
```

**Visual Basic (Declaration)**

```vbnet
Public Class RepMgrSite
```

**Visual C++**

```cpp
public ref class RepMgrSite
```
Inheritance Hierarchy

System..:::Object
BerkeleyDB..:::RepMgrSite
See Also

RepMgrSite Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `RepMgrSite` type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a <a href="#">String</a> that represents the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
</tbody>
</table>
## Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address</td>
<td>The address of the site</td>
</tr>
<tr>
<td>EId</td>
<td>Environment ID assigned by the replication manager. This is the same value that is passed to <code>EventNotify</code> for the <code>REP_NEWMASTER</code> event.</td>
</tr>
<tr>
<td>isConnected</td>
<td>If true, the site is connected.</td>
</tr>
</tbody>
</table>
See Also

RepMgrSite Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The **RepMgrSite** type exposes the following members.
### Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address</td>
<td>The address of the site</td>
</tr>
<tr>
<td>EId</td>
<td>Environment ID assigned by the replication manager. This is the same value that is passed to <code>EventNotify</code> for the <code>REP_NEWMASTER</code> event.</td>
</tr>
<tr>
<td>isConnected</td>
<td>If true, the site is connected.</td>
</tr>
</tbody>
</table>
See Also

RepMgrSite Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
RepMgrSite..Address Field

RepMgrSite Class  See Also

The address of the site

Namespace:  BerkeleyDB
**Syntax**

**C#**

```csharp
public ReplicationHostAddress Address
```

**Visual Basic (Declaration)**

```vbnet
Public Address As ReplicationHostAddress
```

**Visual C++**

```cpp
public:
    ReplicationHostAddress^ Address
```
See Also

RepMgrSite Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
RepMgrSite.EId Field

Environment ID assigned by the replication manager. This is the same value that is passed to `EventNotify` for the `REP_NEWMASTER` event.

**Namespace:** BerkeleyDB  
**Assembly:** `libdb_dotnet48` (in `libdb_dotnet48.dll`) Version: 4.8.24.0
Syntax

C#

public int EId

Visual Basic (Declaration)

Public EId As Integer

Visual C++

public:
  int EId
See Also

RepMgrSite Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
RepMgrSite..:..isConnected Field

If true, the site is connected.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public  bool isConnected

Visual Basic (Declaration)

Public isConnected As Boolean

Visual C++

public:
bool isConnected
See Also

RepMgrSite Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The **RepMgrSite** type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a <a href="#">String</a> that represents the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
</tbody>
</table>
See Also

RepMgrSite Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Statistical information about the Replication Manager

Namespace: **BerkeleyDB**
Syntax

C#
public class RepMgrStats

Visual Basic (Declaration)
Public Class RepMgrStats

Visual C++
public ref class RepMgrStats
Inheritance Hierarchy

System..:::Object
BerkeleyDB..:::RepMgrStats
See Also

RepMgrStats Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The RepMgrStats type exposes the following members.
### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified Object is equal to the current Object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the Type of the current instance.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a String that represents the current Object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
</tbody>
</table>
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DroppedConnections</td>
<td>Existing connections dropped.</td>
</tr>
<tr>
<td>DroppedMessages</td>
<td># msgs discarded due to excessive queue length.</td>
</tr>
<tr>
<td>FailedConnections</td>
<td>Failed new connection attempts.</td>
</tr>
<tr>
<td>FailedMessages</td>
<td># of insufficiently ack'ed msgs.</td>
</tr>
<tr>
<td>QueuedMessages</td>
<td># msgs queued for network delay.</td>
</tr>
</tbody>
</table>
See Also

RepMgrStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `RepMgrStats` type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a <a href="#">String</a> that represents the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
</tbody>
</table>
See Also

RepMgrStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `RepMgrStats` type exposes the following members.
### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DroppedConnections</td>
<td>Existing connections dropped.</td>
</tr>
<tr>
<td>DroppedMessages</td>
<td># msgs discarded due to excessive queue length.</td>
</tr>
<tr>
<td>FailedConnections</td>
<td>Failed new connection attempts.</td>
</tr>
<tr>
<td>FailedMessages</td>
<td># of insufficiently ack'ed msgs.</td>
</tr>
<tr>
<td>QueuedMessages</td>
<td># msgs queued for network delay.</td>
</tr>
</tbody>
</table>
See Also

RepMgrStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Existing connections dropped.

Namespace: BerkeleyDB
Syntax

C#

public ulong DroppedConnections { get; }

Visual Basic (Declaration)

Public ReadOnly Property DroppedConnections As ULong

Visual C++

public:
property unsigned long long DroppedConnections {
    unsigned long long get ();
}

See Also

RepMgrStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation

RepMgrStats..:.DroppedMessages Property

RepMgrStats Class  See Also

# msgs discarded due to excessive queue length.

Namespace:  BerkeleyDB
Syntax

C#

public ulong DroppedMessages { get; }

Visual Basic (Declaration)

Public ReadOnly Property DroppedMessages As ULong

Visual C++

public:
property unsigned long long DroppedMessages {
    unsigned long long get ();
}


See Also

RepMgrStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Failed new connection attempts.

Namespace: BerkeleyDB
Syntax

C#

public ulong FailedConnections { get; }

Visual Basic (Declaration)

Public ReadOnly Property FailedConnections As ULong

Visual C++

public:
property unsigned long long FailedConnections {
    unsigned long long get ();
}

See Also

- RepMgrStats Class
- BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
FailedMessages Property

RepMgrStats Class  See Also

# of insufficiently ack'ed msgs.

Namespace:  BerkeleyDB
Syntax

C#

public ulong FailedMessages { get; }

Visual Basic (Declaration)

Public ReadOnly Property FailedMessages As ULong

Visual C++

public:
property unsigned long long FailedMessages {
    unsigned long long get ();
}

See Also

RepMgrStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
# RepMgrStats Class  See Also

# msgs queued for network delay.

**Namespace:**  [BerkeleyDB](https://example.com/BerkeleyDB)  
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public ulong QueuedMessages { get; }

Visual Basic (Declaration)

Public Readonly Property QueuedMessages As ULong

Visual C++

public:
property unsigned long long QueuedMessages {
    unsigned long long get ();
}
See Also

RepMgrStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
RepProcMsgResult Class

A class representing the return value of `RepProcessMessage(DatabaseEntry, DatabaseEntry, Int32)`.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public class RepProcMsgResult

Visual Basic (Declaration)

Public Class RepProcMsgResult

Visual C++

public ref class RepProcMsgResult
Inheritance Hierarchy

System::Object
BerkeleyDB::RepProcMsgResult
See Also

RepProcMsgResult Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `RepProcMsgResult` type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a <a href="#">String</a> that represents the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Result</td>
<td>The result of processing an incoming replication message.</td>
</tr>
<tr>
<td></td>
<td>The log sequence number of the permanent log message that could not be</td>
</tr>
<tr>
<td></td>
<td>written to disk if Result is NOT_PERMANENT.</td>
</tr>
<tr>
<td></td>
<td>The largest log sequence number of the permanent records that are now</td>
</tr>
<tr>
<td></td>
<td>written to disk as a result of processing the message, if Result is</td>
</tr>
<tr>
<td></td>
<td>IS_PERMANENT. In all other cases the value is undefined.</td>
</tr>
<tr>
<td>RetLsn</td>
<td></td>
</tr>
</tbody>
</table>
See Also

RepProcMsgResult Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `RepProcMsgResult` type exposes the following members.
## Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Result</td>
<td>The result of processing an incoming replication message. The log sequence number of the permanent log message that could not be written to disk if Result is NOT_PERMANENT.</td>
</tr>
<tr>
<td>RetLsn</td>
<td>The largest log sequence number of the permanent records that are now written to disk as a result of processing the message, if Result is IS_PERMANENT. In all other cases the value is undefined.</td>
</tr>
</tbody>
</table>
See Also

RepProcMsgResult Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
RepProcMsgResult Class  See Also

The result of processing an incoming replication message.

Namespace: BerkeleyDB
Syntax

C#

public RepProcMsgResult...::ProcMsgResult Result

Visual Basic (Declaration)

Public Result As RepProcMsgResult...::ProcMsgResult

Visual C++

public:
RepProcMsgResult...::ProcMsgResult Result
See Also

RepProcMsgResult Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
RepProcMsgResult Class  See Also

The log sequence number of the permanent log message that could not be written to disk if Result is NOT_PERMANENT. The largest log sequence number of the permanent records that are now written to disk as a result of processing the message, if Result is IS_PERMANENT. In all other cases the value is undefined.

Namespace: BerkeleyDB
Syntax

C#

public LSN RetLsn

Visual Basic (Declaration)

Public RetLsn As LSN

Visual C++

public:
    LSN^ RetLsn
See Also

RepProcMsgResult Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `RepProcMsgResult` type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="#">Object</a>)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a>)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a <a href="#">String</a> that represents the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>)</td>
</tr>
</tbody>
</table>
See Also

RepProcMsgResult Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
RepProcMsgResult Class  See Also

The result of processing an incoming replication message.

Namespace: BerkeleyDB
Syntax

C#

public enum ProcMsgResult

Visual Basic (Declaration)

Public Enumeration ProcMsgResult

Visual C++

public enum class ProcMsgResult
### Members

<table>
<thead>
<tr>
<th><strong>Member name</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>DUPLICATE_MASTER</td>
<td>The replication group has more than one master.</td>
</tr>
<tr>
<td>ERROR</td>
<td>An unspecified error occurred.</td>
</tr>
<tr>
<td>HOLD_ELECTION</td>
<td>An election is needed.</td>
</tr>
<tr>
<td>IGNORED</td>
<td>A message cannot be processed.</td>
</tr>
<tr>
<td>IS_PERMANENT</td>
<td>Processing a message resulted in the processing of records that are permanent.</td>
</tr>
<tr>
<td>JOIN_FAILURE</td>
<td>A new master has been chosen but the client is unable to synchronize with the new master.</td>
</tr>
<tr>
<td>NEW_SITE</td>
<td>The system received contact information from a new environment.</td>
</tr>
<tr>
<td>NOT_PERMANENT</td>
<td>A message carrying a DB_REP_PERMANENT flag was processed successfully, but was not written to disk.</td>
</tr>
<tr>
<td>SUCCESS</td>
<td>Processing a message succeeded.</td>
</tr>
</tbody>
</table>
See Also

BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB has encountered an error it considers fatal to an entire environment. Once a RunRecoveryException has been thrown by any interface, it will be returned from all subsequent Berkeley DB calls made by any threads of control participating in the environment.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
**Syntax**

**C#**

```csharp
public class RunRecoveryException : DatabaseException
```

**Visual Basic (Declaration)**

```vbnet
Public Class RunRecoveryException
    Inherits DatabaseException
```

**Visual C++**

```cpp
public ref class RunRecoveryException : public DatabaseException
```
Remarks

An example of this type of fatal error is a corrupted database page. The only way to recover from this type of error is to have all threads of control exit the Berkeley DB environment, run recovery of the environment, and re-enter Berkeley DB. (It is not strictly necessary that the processes exit, although that is the only way to recover system resources, such as file descriptors and memory, allocated by Berkeley DB.)
Inheritance Hierarchy

System...:::Object
  System...:::Exception
    BerkeleyDB...:::DatabaseException
      BerkeleyDB...:::RunRecoveryException
See Also

RunRecoveryException Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The RunRecoveryException type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RunRecoveryException</td>
<td>Initialize a new instance of the RunRecoveryException</td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified Object is equal to the current Object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td><strong>GetBaseException</strong></td>
<td>Returns the Exception that is the root cause of one or more subsequent exceptions.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Exception.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td><strong>GetObjectData</strong></td>
<td>Sets the SerializationInfo with information about the exception.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Exception.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the runtime type of the current instance.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Exception.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Creates and returns a string representation of the current exception.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Exception.)</td>
</tr>
</tbody>
</table>
Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ErrorCode</td>
<td>The underlying error code from the Berkeley DB C library. (Inherited from DatabaseException.)</td>
</tr>
</tbody>
</table>
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from <a href="#">Exception</a>.)</td>
</tr>
<tr>
<td>HelpLink</td>
<td>Gets or sets a link to the help file associated with this exception. (Inherited from <a href="#">Exception</a>.)</td>
</tr>
<tr>
<td>InnerException</td>
<td>Gets the <a href="#">Exception</a> instance that caused the current exception. (Inherited from <a href="#">Exception</a>.)</td>
</tr>
<tr>
<td>Message</td>
<td>Gets a message that describes the current exception. (Inherited from <a href="#">Exception</a>.)</td>
</tr>
<tr>
<td>Source</td>
<td>Gets or sets the name of the application or the object that causes the error. (Inherited from <a href="#">Exception</a>.)</td>
</tr>
<tr>
<td>StackTrace</td>
<td>Gets a string representation of the frames on the call stack at the time the current exception was thrown. (Inherited from <a href="#">Exception</a>.)</td>
</tr>
<tr>
<td>TargetSite</td>
<td>Gets the method that throws the current exception. (Inherited from <a href="#">Exception</a>.)</td>
</tr>
</tbody>
</table>
See Also

RunRecoveryException Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
RunRecoveryException Constructor

Initialize a new instance of the RunRecoveryException

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll)  
**Version:** 4.8.24.0
Syntax

C#

public RunRecoveryException()

Visual Basic (Declaration)

Public Sub New

Visual C++

public:
RunRecoveryException()
See Also

RunRecoveryException Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `RunRecoveryException` type exposes the following members.
## Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ErrorCode</td>
<td>The underlying error code from the Berkeley DB C library. (Inherited from DatabaseException.)</td>
</tr>
</tbody>
</table>
See Also

RunRecoveryException Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `RunRecoveryException` type exposes the following members.
### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| Equals        | Determines whether the specified [Object](https://docs.microsoft.com/en-us/dotnet/api/system.object) is equal to the current [Object](https://docs.microsoft.com/en-us/dotnet/api/system.object). When overridden in a derived class, returns the [Exception](https://docs.microsoft.com/en-us/dotnet/api/system.exception) that is the root cause of one or more subsequent exceptions.  
(Inherited from [Exception](https://docs.microsoft.com/en-us/dotnet/api/system.exception).) |
| GetBaseException | Serves as a hash function for a particular type. When overridden in a derived class, sets the [SerializationInfo](https://docs.microsoft.com/en-us/dotnet/api/system.serialization.serializationinfo) with information about the exception.  
(Inherited from [Exception](https://docs.microsoft.com/en-us/dotnet/api/system.exception).) |
| GetHashCode | Serves as a hash function for a particular type. When overridden in a derived class, sets the [SerializationInfo](https://docs.microsoft.com/en-us/dotnet/api/system.serialization.serializationinfo) with information about the exception.  
(Inherited from [Exception](https://docs.microsoft.com/en-us/dotnet/api/system.exception).) |
| GetObjectData | Gets the runtime type of the current instance.  
(Inherited from [Exception](https://docs.microsoft.com/en-us/dotnet/api/system.exception).) |
| GetType      | Gets the runtime type of the current instance.  
(Inherited from [Exception](https://docs.microsoft.com/en-us/dotnet/api/system.exception).) |
| ToString     | Creates and returns a string representation of the current exception.  
(Inherited from [Exception](https://docs.microsoft.com/en-us/dotnet/api/system.exception).) |
See Also

RunRecoveryException Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The **RunRecoveryException** type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td>HelpLink</td>
<td>Gets or sets a link to the help file associated with this exception.         (Inherited from Exception.)</td>
</tr>
<tr>
<td>InnerException</td>
<td>Gets the Exception instance that caused the current exception.            (Inherited from Exception.)</td>
</tr>
<tr>
<td>Message</td>
<td>Gets a message that describes the current exception.                      (Inherited from Exception.)</td>
</tr>
<tr>
<td>Source</td>
<td>Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)</td>
</tr>
<tr>
<td>StackTrace</td>
<td>Gets a string representation of the frames on the call stack at the time the current exception was thrown. (Inherited from Exception.)</td>
</tr>
<tr>
<td>TargetSite</td>
<td>Gets the method that throws the current exception.                         (Inherited from Exception.)</td>
</tr>
</tbody>
</table>
See Also

RunRecoveryException Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
A class representing a SecondaryBTreeDatabase. The Btree format is a representation of a sorted, balanced tree structure.

**Namespace**: BerkeleyDB

**Assembly**: libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public class SecondaryBTreeDatabase : SecondaryDatabase

Visual Basic (Declaration)

Public Class SecondaryBTreeDatabase
    Inherits SecondaryDatabase

Visual C++

public ref class SecondaryBTreeDatabase : public SecondaryDatabase
Inheritance Hierarchy

System..:::Object
  BerkeleyDB..:::BaseDatabase
    BerkeleyDB..:::SecondaryDatabase
      BerkeleyDB..:::SecondaryBTreeDatabase
See Also

SecondaryBTreeDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `SecondaryBTreeDatabase` type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overloaded.</td>
<td>Although closing a database will close any open cursors, it is recommended that applications explicitly close all their Cursor objects before closing the database. The reason why is that when the cursor is explicitly closed, the memory allocated for it is reclaimed; however, this will not happen if you close a database while cursors are still opened. The same rule, for the same reasons, hold true for <strong>Transaction</strong> objects. Simply make sure you resolve all your transaction objects before closing your database handle. Because key/data pairs are cached in memory, applications should make a point to always either close database handles or sync their data to disk (using <strong>Sync</strong> before exiting, to ensure that any data cached in main memory are reflected in the underlying file system. When called on a database that is the primary database for a secondary index, the primary database should be closed only after all secondary indices referencing it have been closed. When multiple threads are using the object concurrently, only a single thread may call the Close method. The object may not be accessed again after Close is called, regardless of its outcome.</td>
</tr>
</tbody>
</table>
Cursor
Overloaded.

Delete
Overloaded.

Dispose
Release the resources held by this object, and close the database if it's still open.
(Inherited from BaseDatabase.)

Equals
Determines whether the specified Object is equal to the current Object.
(Inherited from Object.)

Exists
Overloaded.

Get
Overloaded.

GetBoth
Overloaded.

GetHashCode
Serves as a hash function for a particular type.
(Inherited from Object.)

GetType
Gets the Type of the current instance.
(Inherited from Object.)

Open
Overloaded.

PrintFastStats
The statistical information is described by the BTreeStats, HashStats, QueueStats, and RecnoStats classes.

PrintStats
The statistical information is described by the BTreeStats, HashStats, QueueStats, and RecnoStats classes.

SecondaryCursor
Overloaded.

Sync
Flush any cached information to disk.
(Inherited from BaseDatabase.)

ToString
Returns a String that represents the current Object.
(Inherited from Object.)

Truncate
Overloaded.

When called on a database configured with secondary indices, Truncate will truncate the primary database and all secondary indices. A count of the records discarded from the primary database is returned.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AutoCommit</strong></td>
<td>If true, all database modification operations based on this object will be transactionally protected. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>CacheSize</strong></td>
<td>The size of the shared memory buffer pool -- that is, the cache. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>Compare</strong></td>
<td>The Btree key comparison function. The comparison function is called whenever it is necessary to compare a key specified by the application with a key currently stored in the tree. The CreatePolicy with which this database was opened. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>Creation</strong></td>
<td>The CreatePolicy with which this database was opened. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>DatabaseName</strong></td>
<td>The name of this database, if it has one. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>DoChecksum</strong></td>
<td>If true, do checksum verification of pages read into the cache from the backing filestore. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>DupCompare</strong></td>
<td>The duplicate data item comparison function.</td>
</tr>
<tr>
<td><strong>Duplicates</strong></td>
<td>Whether the insertion of duplicate data items in the database is permitted, and whether duplicates items are sorted.</td>
</tr>
<tr>
<td><strong>EncryptAlgorithm</strong></td>
<td>The algorithm used by the Berkeley DB library to perform encryption and decryption. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>Encrypted</strong></td>
<td>If true, encrypt all data stored in the database. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>Endianness</strong></td>
<td>The database byte order. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>ErrorFeedback</strong></td>
<td>The mechanism for reporting detailed error messages to the application.</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ErrorPrefix</td>
<td>The prefix string that appears before error messages issued by Berkeley DB.</td>
</tr>
<tr>
<td>Feedback</td>
<td>Monitor progress within long running operations.</td>
</tr>
<tr>
<td>FileName</td>
<td>The filename of this database, if it has one.</td>
</tr>
<tr>
<td>FreeThreaded</td>
<td>If true, the object is free-threaded; that is, concurrently usable by multiple threads in the address space.</td>
</tr>
<tr>
<td>HasMultiple</td>
<td>If true, the object references a physical file supporting multiple databases.</td>
</tr>
<tr>
<td>InHostOrder</td>
<td>If true, the underlying database files were created on an architecture of the same byte order as the current one. This information may be used to determine whether application data needs to be adjusted for this architecture or not.</td>
</tr>
<tr>
<td>KeyGen</td>
<td>The delegate that creates the set of secondary keys corresponding to a given primary key and data pair.</td>
</tr>
<tr>
<td>MinKeysPerPage</td>
<td>The minimum number of key/data pairs intended to be stored on any single Btree leaf page.</td>
</tr>
<tr>
<td>NoMMap</td>
<td>If true, this database is not mapped into process memory.</td>
</tr>
<tr>
<td>NonDurableTxns</td>
<td>If true, Berkeley DB will not write log records for this database.</td>
</tr>
<tr>
<td>Nullifier</td>
<td>(Inherited from SecondaryDatabase.)</td>
</tr>
</tbody>
</table>
| **Pagesize** | The database's current page size.  
(Inherited from [BaseDatabase](#)).  
The Btree prefix function. The prefix function is used to determine the amount by which keys stored on the Btree internal pages can be safely truncated without losing their uniqueness. |
| **PrefixCompare** | The cache priority for pages referenced by this object.  
(Inherited from [BaseDatabase](#)).  
If true, this database has been opened for reading only. Any attempt to modify items in the database will fail, regardless of the actual permissions of any underlying files.  
(Inherited from [BaseDatabase](#)).  
If true, this database supports transactional read operations with degree 1 isolation. Read operations on the database may request the return of modified but not yet committed data.  
(Inherited from [BaseDatabase](#)).  
If true, this database supports retrieval from the Btree using record numbers. |
| **Priority** | If false, empty pages will not be coalesced into higher-level pages. |
| **ReadOnly** | If true, this database has been opened in a transactional mode.  
(Inherited from [BaseDatabase](#)).  
If true, the underlying file was physically truncated upon open, discarding all previous databases it might have held.  
(Inherited from [BaseDatabase](#)). |
| **RecordNumbers** | The type of the underlying access method (and file format). This value may be used to determine the type of the database after an `Open(String, DatabaseConfig)`.  
(Inherited from [BaseDatabase](#)). |
| **ReadUncommitted** | If true, the database was opened with support for multiversion concurrency control. |
| **ReverseSplit** | |
(Inherited from BaseDatabase.)
See Also

SecondaryBTreeDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The **SecondaryBTreeDatabase** type exposes the following members.
Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overloaded.</td>
<td>Although closing a database will close any open cursors, it is recommended that applications explicitly close all their Cursor objects before closing the database. The reason why is that when the cursor is explicitly closed, the memory allocated for it is reclaimed; however, this will not happen if you close a database while cursors are still opened.</td>
</tr>
</tbody>
</table>

The same rule, for the same reasons, hold true for **Transaction** objects. Simply make sure you resolve all your transaction objects before closing your database handle.

Because key/data pairs are cached in memory, applications should make a point to always either close database handles or sync their data to disk (using **Sync()**) before exiting, to ensure that any data cached in main memory are reflected in the underlying file system.

When called on a database that is the primary database for a secondary index, the primary database should be closed only after all secondary indices referencing it have been closed.

When multiple threads are using the object concurrently, only a single thread may call the Close method.

The object may not be accessed again after Close is called, regardless of its outcome.
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Cursor</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>Delete</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>Dispose</strong></td>
<td>Release the resources held by this object, and close the database if it's still open.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified Object is equal to the current Object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td><strong>Exists</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>Get</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>GetBoth</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the Type of the current instance.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td><strong>Open</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>PrintFastStats</strong></td>
<td>The statistical information is described by the BTreeStats, HashStats, QueueStats, and RecnoStats classes.</td>
</tr>
<tr>
<td><strong>PrintStats</strong></td>
<td>The statistical information is described by the BTreeStats, HashStats, QueueStats, and RecnoStats classes.</td>
</tr>
<tr>
<td><strong>SecondaryCursor</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>Sync</strong></td>
<td>Flush any cached information to disk.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a String that represents the current Object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td><strong>Truncate</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td></td>
<td>When called on a database configured with secondary indices, Truncate will truncate the primary database and all secondary indices. A count of the records discarded from the primary database is returned.</td>
</tr>
</tbody>
</table>
See Also

SecondaryBTreeDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Although closing a database will close any open cursors, it is recommended that applications explicitly close all their Cursor objects before closing the database. The reason why is that when the cursor is explicitly closed, the memory allocated for it is reclaimed; however, this will not happen if you close a database while cursors are still opened.

The same rule, for the same reasons, hold true for Transaction objects. Simply make sure you resolve all your transaction objects before closing your database handle.

Because key/data pairs are cached in memory, applications should make a point to always either close database handles or sync their data to disk (using Sync()) before exiting, to ensure that any data cached in main memory are reflected in the underlying file system.

When called on a database that is the primary database for a secondary index, the primary database should be closed only after all secondary indices referencing it have been closed.

When multiple threads are using the object concurrently, only a single thread may call the Close method.

The object may not be accessed again after Close is called, regardless of its outcome.
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Close()()</td>
<td>Flush any cached database information to disk, close any open <code>Cursor()()</code> objects, free any allocated resources, and close any underlying files. (Inherited from <code>BaseDatabase</code>.)</td>
</tr>
<tr>
<td>Close(Boolean)</td>
<td>Optionally flush any cached database information to disk, close any open <code>Cursor()()</code> objects, free any allocated resources, and close any underlying files. (Inherited from <code>BaseDatabase</code>.)</td>
</tr>
</tbody>
</table>
See Also

SecondaryBTreeDatabase Class
SecondaryBTreeDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryBTreeDatabase.CURSOR METHOD

See Also

SecondaryBTreeDatabase Class
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| Cursor()() | Create a database cursor.  
(Inherited from [BaseDatabase](#).) |
| Cursor(CursorConfig) | Create a database cursor with the given configuration.  
(Inherited from [BaseDatabase](#).) |
| Cursor(Transaction) | Create a transactionally protected database cursor.  
(Inherited from [BaseDatabase](#).) |
| Cursor(CursorConfig, Transaction) | Create a transactionally protected database cursor with the given configuration.  
(Inherited from [BaseDatabase](#).) |
See Also

SecondaryBTreeDatabase Class
SecondaryBTreeDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryBTreeDatabase Delete Method

See Also

SecondaryBTreeDatabase Class
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delete(DatabaseEntry)</td>
<td>Remove key/data pairs from the database. The key/data pair associated with key is discarded from the database. In the presence of duplicate key values, all records associated with the designated key will be discarded. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>Delete(DatabaseEntry, Transaction)</td>
<td>Remove key/data pairs from the database. The key/data pair associated with key is discarded from the database. In the presence of duplicate key values, all records associated with the designated key will be discarded. (Inherited from BaseDatabase.)</td>
</tr>
</tbody>
</table>
See Also

SecondaryBTreeDatabase Class
SecondaryBTreeDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryBTreeDatabase::Exists Method

SecondaryBTreeDatabase Class  See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exists(DatabaseEntry)</td>
<td>Check whether key appears in the database. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>Exists(DatabaseEntry, Transaction)</td>
<td>Check whether key appears in the database. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>Exists(DatabaseEntry, Transaction, LockingInfo)</td>
<td>Check whether key appears in the database. (Inherited from BaseDatabase.)</td>
</tr>
</tbody>
</table>
See Also

SecondaryBTreeDatabase Class
SecondaryBTreeDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryBTreeDatabase.Get Method

SecondaryBTreeDatabase Class  See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Get(DatabaseEntry)</td>
<td>Retrieve a key/data pair from the database. In the presence of duplicate key values, Get will return the first data item for key. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>Get(DatabaseEntry, Transaction)</td>
<td>Retrieve a key/data pair from the database. In the presence of duplicate key values, Get will return the first data item for key. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>Get(DatabaseEntry, Transaction, LockingInfo)</td>
<td>Retrieve a key/data pair from the database. In the presence of duplicate key values, Get will return the first data item for key. (Inherited from BaseDatabase.)</td>
</tr>
</tbody>
</table>
See Also

SecondaryBTreeDatabase Class
SecondaryBTreeDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation
SecondaryBTreeDatabase.:::GetBoth Method
SecondaryBTreeDatabase Class  See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetBoth(DatabaseEntry, DatabaseEntry)</td>
<td>Retrieve a key/data pair from the database which matches key and data.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>GetBoth(DatabaseEntry, DatabaseEntry, Transaction)</td>
<td>Retrieve a key/data pair from the database which matches key and data.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>GetBoth(DatabaseEntry, DatabaseEntry, Transaction, LockingInfo)</td>
<td>Retrieve a key/data pair from the database which matches key and data.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from BaseDatabase.)</td>
</tr>
</tbody>
</table>
See Also

*SecondaryBTreeDatabase Class*
*SecondaryBTreeDatabase Members*
*BerkeleyDB Namespace*

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryBTreeDatabase Class

See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Open(String, SecondaryBTreeDatabaseConfig)</code></td>
<td>Instantiate a new SecondaryBTreeDatabase object, open the database represented by Filename and associate the database with the primary index.</td>
</tr>
<tr>
<td><code>Open(String, SecondaryBTreeDatabaseConfig, Transaction)</code></td>
<td>Instantiate a new SecondaryBTreeDatabase object, open the database represented by Filename and associate the database with the primary index.</td>
</tr>
<tr>
<td><code>Open(String, String, SecondaryBTreeDatabaseConfig)</code></td>
<td>Instantiate a new SecondaryBTreeDatabase object, open the database represented by Filename and associate the database with the primary index.</td>
</tr>
<tr>
<td><code>Open(String, String, SecondaryBTreeDatabaseConfig, Transaction)</code></td>
<td>Instantiate a new SecondaryBTreeDatabase object, open the database represented by Filename and associate the database with the primary index.</td>
</tr>
</tbody>
</table>
See Also

SecondaryBTreeDatabase Class
SecondaryBTreeDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Instantiate a new SecondaryBTreeDatabase object, open the database represented by Filename and associate the database with the primary index.

Namespace: BerkeleyDB
Syntax

C#

public static SecondaryBTreeDatabase Open(
    string Filename,
    SecondaryBTreeDatabaseConfig cfg
)

Visual Basic (Declaration)

Public Shared Function Open ( _
    Filename As String, _
    cfg As SecondaryBTreeDatabaseConfig _
) As SecondaryBTreeDatabase

Visual C++

public:
static SecondaryBTreeDatabase^ Open(
    String^ Filename,
    SecondaryBTreeDatabaseConfig^ cfg
)

Parameters

Filename
Type: System::String
The name of an underlying file that will be used to back the database. In-memory databases never intended to be preserved on disk may be created by setting this parameter to null.

cfg
Type: BerkeleyDB::SecondaryBTreeDatabaseConfig
The database's configuration

Return Value
A new, open database object
Remarks

If Filename is null, the database is strictly temporary and cannot be opened by any other thread of control, thus the database can only be accessed by sharing the single database object that created it, in circumstances where doing so is safe.

If AutoCommit is set, the operation will be implicitly transaction protected. Note that transactionally protected operations on a database object requires the object itself be transactionally protected during its open.
See Also

SecondaryBTreeDatabase Class
Open Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryBTreeDatabase::Open Method (String,
SecondaryBTreeDatabaseConfig, Transaction)

**SecondaryBTreeDatabase Class** [See Also](#)

Instantiate a new SecondaryBTreeDatabase object, open the database represented by Filename and associate the database with the primary index.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

```csharp
public static SecondaryBTreeDatabase Open(
    string Filename,
    SecondaryBTreeDatabaseConfig cfg,
    Transaction txn
)
```

Visual Basic (Declaration)

```vbnet
Public Shared Function Open ( _
    Filename As String, _
    cfg As SecondaryBTreeDatabaseConfig, _
    txn As Transaction _
) As SecondaryBTreeDatabase
```

Visual C++

```cpp
public:
static SecondaryBTreeDatabase^ Open(
    String^ Filename,
    SecondaryBTreeDatabaseConfig^ cfg,
    Transaction^ txn
)
```

Parameters

Filename
Type: `System::String`
The name of an underlying file that will be used to back the database. In-memory databases never intended to be preserved on disk may be created by setting this parameter to null.

cfg
Type: `BerkeleyDB::SecondaryBTreeDatabaseConfig`
The database’s configuration
txn

Type: BerkeleyDB::<Transaction>

If the operation is part of an application-specified transaction, txn is a Transaction object returned from BeginTransaction(); if the operation is part of a Berkeley DB Concurrent Data Store group, txn is a handle returned from BeginCDSGroup(); otherwise null.

**Return Value**

A new, open database object
Remarks

If Filename is null, the database is strictly temporary and cannot be opened by any other thread of control, thus the database can only be accessed by sharing the single database object that created it, in circumstances where doing so is safe.

If txn is null, but AutoCommit is set, the operation will be implicitly transaction protected. Note that transactionally protected operations on a database object requires the object itself be transactionally protected during its open. Also note that the transaction must be committed before the object is closed.
See Also

SecondaryBTreeDatabase Class
Open Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Instantiate a new SecondaryBTreeDatabase object, open the database represented by Filename and associate the database with the primary index.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public static SecondaryBTreeDatabase Open(
    string Filename,
    string DatabaseName,
    SecondaryBTreeDatabaseConfig cfg
)

Visual Basic (Declaration)

Public Shared Function Open (_
    Filename As String, _
    DatabaseName As String, _
    cfg As SecondaryBTreeDatabaseConfig _
) As SecondaryBTreeDatabase

Visual C++

public:
static SecondaryBTreeDatabase^ Open(
    String^ Filename,
    String^ DatabaseName,
    SecondaryBTreeDatabaseConfig^ cfg
)

Parameters

Filename
Type: System::String
The name of an underlying file that will be used to back the database. In-memory databases never intended to be preserved on disk may be created by setting this parameter to null.

DatabaseName
Type: System::String
This parameter allows applications to have multiple databases in a single file. Although no DatabaseName needs to be specified, it is an error to
attempt to open a second database in a file that was not initially created using a database name.

cfg

Type: BerkeleyDB::SecondaryBTreeDatabaseConfig

The database's configuration

**Return Value**

A new, open database object
Remarks

If both Filename and DatabaseName are null, the database is strictly temporary and cannot be opened by any other thread of control, thus the database can only be accessed by sharing the single database object that created it, in circumstances where doing so is safe. If Filename is null and DatabaseName is non-null, the database can be opened by other threads of control and will be replicated to client sites in any replication group.

If AutoCommit is set, the operation will be implicitly transaction protected. Note that transactionally protected operations on a database object requires the object itself be transactionally protected during its open.
See Also

SecondaryBTreeDatabase Class
Open Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryBTreeDatabase...:::Open Method (String, String, SecondaryBTreeDatabaseConfig, Transaction)

See Also

Instantiate a new SecondaryBTreeDatabase object, open the database represented by Filename and associate the database with the primary index.

Namespace: BerkeleyDB
**Syntax**

**C#**

```csharp
public static SecondaryBTreeDatabase Open(
    string Filename,
    string DatabaseName,
    SecondaryBTreeDatabaseConfig cfg,
    Transaction txn)
```

**Visual Basic (Declaration)**

```vbnet
Public Shared Function Open (
    _
    Filename As String, _
    DatabaseName As String, _
    cfg As SecondaryBTreeDatabaseConfig, _
    txn As Transaction _
) As SecondaryBTreeDatabase
```

**Visual C++**

```cpp
public:
static SecondaryBTreeDatabase^ Open(
    String^ Filename,
    String^ DatabaseName,
    SecondaryBTreeDatabaseConfig^ cfg,
    Transaction^ txn
)
```

**Parameters**

**Filename**

Type: `System::String`

The name of an underlying file that will be used to back the database. In-memory databases never intended to be preserved on disk may be created by setting this parameter to null.

**DatabaseName**

Type: `System::String`
This parameter allows applications to have multiple databases in a single file. Although no DatabaseName needs to be specified, it is an error to attempt to open a second database in a file that was not initially created using a database name.

cfg
Type: `BerkeleyDB::SecondaryBTreeDatabaseConfig`
The database's configuration

txn
Type: `BerkeleyDB::Transaction`
If the operation is part of an application-specified transaction, txn is a Transaction object returned from `BeginTransaction()`; if the operation is part of a Berkeley DB Concurrent Data Store group, txn is a handle returned from `BeginCDSGroup()`; otherwise null.

**Return Value**

A new, open database object
Remarks

If both Filename and DatabaseName are null, the database is strictly temporary and cannot be opened by any other thread of control, thus the database can only be accessed by sharing the single database object that created it, in circumstances where doing so is safe. If Filename is null and DatabaseName is non-null, the database can be opened by other threads of control and will be replicated to client sites in any replication group.

If txn is null, but AutoCommit is set, the operation will be implicitly transaction protected. Note that transactionally protected operations on a database object requires the object itself be transactionally protected during its open. Also note that the transaction must be committed before the object is closed.
See Also

SecondaryBTreeDatabase Class
Open Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The statistical information is described by the `BTreeStats`, `HashStats`, `QueueStats`, and `RecnoStats` classes.
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PrintFastStats()()()</td>
<td>Display the database statistical information which does not require traversal of the database. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>PrintFastStats(Boolean)</td>
<td>Display the database statistical information which does not require traversal of the database. (Inherited from BaseDatabase.)</td>
</tr>
</tbody>
</table>
See Also

SecondaryBTreeDatabase Class
SecondaryBTreeDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The statistical information is described by the `BTreeStats`, `HashStats`, `QueueStats`, and `RecnoStats` classes.
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PrintStats()</td>
<td>Display the database statistical information.</td>
</tr>
<tr>
<td>(Inherited from BaseDatabase.)</td>
<td></td>
</tr>
<tr>
<td>PrintStats(Boolean)</td>
<td>Display the database statistical information.</td>
</tr>
<tr>
<td>(Inherited from BaseDatabase.)</td>
<td></td>
</tr>
</tbody>
</table>
See Also

SecondaryBTreeDatabase Class
SecondaryBTreeDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryBTreeDatabase Class

See Also
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>SecondaryCursor()</code>()</td>
<td>Create a secondary database cursor. (Inherited from <a href="#">SecondaryDatabase</a>.)</td>
</tr>
<tr>
<td><code>SecondaryCursor(CursorConfig)</code></td>
<td>Create a secondary database cursor with the given configuration. (Inherited from <a href="#">SecondaryDatabase</a>.)</td>
</tr>
<tr>
<td><code>SecondaryCursor(Transaction)</code></td>
<td>Create a transactionally protected secondary database cursor. (Inherited from <a href="#">SecondaryDatabase</a>.)</td>
</tr>
<tr>
<td><code>SecondaryCursor(CursorConfig, Transaction)</code></td>
<td>Create a transactionally protected secondary database cursor with the given configuration. (Inherited from <a href="#">SecondaryDatabase</a>.)</td>
</tr>
</tbody>
</table>
See Also

SecondaryBTreeDatabase Class
SecondaryBTreeDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
When called on a database configured with secondary indices, Truncate will truncate the primary database and all secondary indices. A count of the records discarded from the primary database is returned.
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Truncate()</strong></td>
<td>Empty the database, discarding all records it contains. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>Truncate(Transaction)</strong></td>
<td>Empty the database, discarding all records it contains. (Inherited from BaseDatabase.)</td>
</tr>
</tbody>
</table>
See Also

SecondaryBTreeDatabase Class
SecondaryBTreeDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `SecondaryBTreeDatabase` type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AutoCommit</strong></td>
<td>If true, all database modification operations based on this object will be transactionally protected. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>CacheSize</strong></td>
<td>The size of the shared memory buffer pool -- that is, the cache. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>Compare</strong></td>
<td>The Btree key comparison function. The comparison function is called whenever it is necessary to compare a key specified by the application with a key currently stored in the tree. The CreatePolicy with which this database was opened. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>Creation</strong></td>
<td>The CreatePolicy with which this database was opened. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>DatabaseName</strong></td>
<td>The name of this database, if it has one. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>DoChecksum</strong></td>
<td>If true, do checksum verification of pages read into the cache from the backing filestore. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>DupCompare</strong></td>
<td>The duplicate data item comparison function. Whether the insertion of duplicate data items in the database is permitted, and whether duplicates items are sorted.</td>
</tr>
<tr>
<td><strong>Duplicates</strong></td>
<td>The algorithm used by the Berkeley DB library to perform encryption and decryption. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>EncryptAlgorithm</strong></td>
<td>If true, encrypt all data stored in the database. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>Encrypted</strong></td>
<td>The database byte order. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>Endianness</strong></td>
<td>The mechanism for reporting detailed error messages to the application.</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ErrorPrefix</td>
<td>The prefix string that appears before error messages issued by Berkeley DB.</td>
</tr>
<tr>
<td>Feedback</td>
<td>Monitor progress within long running operations.</td>
</tr>
<tr>
<td>FileName</td>
<td>The filename of this database, if it has one.</td>
</tr>
<tr>
<td>FreeThreaded</td>
<td>If true, the object is free-threaded; that is, concurrently usable by multiple threads in the address space.</td>
</tr>
<tr>
<td>HasMultiple</td>
<td>If true, the object references a physical file supporting multiple databases.</td>
</tr>
<tr>
<td>InHostOrder</td>
<td>If true, the underlying database files were created on an architecture of the same byte order as the current one. This information may be used to determine whether application data needs to be adjusted for this architecture or not.</td>
</tr>
<tr>
<td>KeyGen</td>
<td>The delegate that creates the set of secondary keys corresponding to a given primary key and data pair.</td>
</tr>
<tr>
<td>MinKeysPerPage</td>
<td>The minimum number of key/data pairs intended to be stored on any single Btree leaf page.</td>
</tr>
<tr>
<td>NoMMap</td>
<td>If true, this database is not mapped into process memory.</td>
</tr>
<tr>
<td>NonDurableTxns</td>
<td>If true, Berkeley DB will not write log records for this database.</td>
</tr>
<tr>
<td>Nullifier</td>
<td>(Inherited from SecondaryDatabase.)</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Pagesize</td>
<td>The database's current page size. (Inherited from <a href="#">BaseDatabase</a>). The Btree prefix function. The prefix function is used to determine the amount by which keys stored on the Btree internal pages can be safely truncated without losing their uniqueness.</td>
</tr>
<tr>
<td>PrefixCompare</td>
<td>The cache priority for pages referenced by this object. (Inherited from <a href="#">BaseDatabase</a>). If true, this database has been opened for reading only. Any attempt to modify items in the database will fail, regardless of the actual permissions of any underlying files.</td>
</tr>
<tr>
<td>Priority</td>
<td>If true, this database supports transactional read operations with degree 1 isolation. Read operations on the database may request the return of modified but not yet committed data. (Inherited from <a href="#">BaseDatabase</a>). If true, this database has been opened in a transactional mode.</td>
</tr>
<tr>
<td>ReadOnly</td>
<td>If true, the database has been opened for reading only. Any attempt to modify items in the database will fail, regardless of the actual permissions of any underlying files. (Inherited from <a href="#">BaseDatabase</a>). If true, this database supports transactional read operations with degree 1 isolation. Read operations on the database may request the return of modified but not yet committed data.</td>
</tr>
<tr>
<td>RecordNumbers</td>
<td>The type of the underlying access method (and file format). This value may be used to determine the type of the database after an <a href="##">Open(String, DatabaseConfig)</a>. (Inherited from <a href="#">BaseDatabase</a>). If true, the database was opened with support for multiversion concurrency control.</td>
</tr>
<tr>
<td>ReverseSplit</td>
<td>If true, this object supports retrieval from the Btree using record numbers. If false, empty pages will not be coalesced into higher-level pages.</td>
</tr>
<tr>
<td>Transactional</td>
<td>The type of the underlying access method (and file format). This value may be used to determine the type of the database after an <a href="##">Open(String, DatabaseConfig)</a>. (Inherited from <a href="#">BaseDatabase</a>). If true, the underlying file was physically truncated upon open, discarding all previous databases it might have held.</td>
</tr>
<tr>
<td>Truncated</td>
<td>The cache priority for pages referenced by this object. (Inherited from <a href="#">BaseDatabase</a>). If true, this database has been opened for reading only. Any attempt to modify items in the database will fail, regardless of the actual permissions of any underlying files.</td>
</tr>
<tr>
<td>UseMVCC</td>
<td>If true, this database was opened with support for multiversion concurrency control.</td>
</tr>
</tbody>
</table>
(Inherited from BaseDatabase.)
See Also

SecondaryBTreeDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The Btree key comparison function. The comparison function is called whenever it is necessary to compare a key specified by the application with a key currently stored in the tree.

**Namespace:** BerkeleyDB
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public EntryComparisonDelegate Compare { get; private set; }

Visual Basic (Declaration)

Public Property Compare As EntryComparisonDelegate

Visual C++

public:
    property EntryComparisonDelegate^ Compare {
        EntryComparisonDelegate^ get ();
        void set (EntryComparisonDelegate^ value);
    }
See Also

SecondaryBTreeDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryBTreeDatabase::DupCompare Property

The duplicate data item comparison function.

Namespace: BerkeleyDB
Syntax

C#

public EntryComparisonDelegate DupCompare { get; private set; }

Visual Basic (Declaration)

Public Property DupCompare As EntryComparisonDelegate

Visual C++

public:

property EntryComparisonDelegate^ DupCompare {
    EntryComparisonDelegate^ get ();
    void set (EntryComparisonDelegate^ value);
}
See Also

SecondaryBTreeDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Whether the insertion of duplicate data items in the database is permitted, and whether duplicates items are sorted.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
**Syntax**

**C#**

```csharp
public DuplicatesPolicy Duplicates { get; }
```

**Visual Basic (Declaration)**

```vbnet
Public ReadOnly Property Duplicates As DuplicatesPolicy
```

**Visual C++**

```cpp
public:
property DuplicatesPolicy Duplicates {
    DuplicatesPolicy get ();
}
```
See Also

SecondaryBTreeDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryBTreeDatabase Class  See Also

The minimum number of key/data pairs intended to be stored on any single Btree leaf page.

Namespace: BerkeleyDB
Syntax

C#

public uint MinKeysPerPage { get; }

Visual Basic (Declaration)

Public ReadOnly Property MinKeysPerPage As UInteger

Visual C++

public:
property unsigned int MinKeysPerPage {
    unsigned int get ();
}

See Also

SecondaryBTreeDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryBTreeDatabase::PrefixCompare Property

The Btree prefix function. The prefix function is used to determine the amount by which keys stored on the Btree internal pages can be safely truncated without losing their uniqueness.

Namespace: BerkeleyDB
Syntax

C#

public EntryComparisonDelegate PrefixCompare { get; private set; }

Visual Basic (Declaration)

Public Property PrefixCompare As EntryComparisonDelegate

Visual C++

public:
property EntryComparisonDelegate^ PrefixCompare {
    EntryComparisonDelegate^ get ();
    void set (EntryComparisonDelegate^ value);
}
See Also

SecondaryBTreeDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, this object supports retrieval from the Btree using record numbers.

Namespace:  BerkeleyDB  
Assembly:  libdb_dotnet48 (in libdb_dotnet48.dll)  
Version: 4.8.24.0
**Syntax**

**C#**

```csharp
public bool RecordNumbers { get; }
```

**Visual Basic (Declaration)**

```vbnet
Public ReadOnly Property RecordNumbers As Boolean
```

**Visual C++**

```c++
public:
property bool RecordNumbers {
    bool get ();
}
```
See Also

SecondaryBTreeDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If false, empty pages will not be coalesced into higher-level pages.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
# Syntax

**C#**

```csharp
public bool ReverseSplit { get; }
```

**Visual Basic (Declaration)**

```vbnet
Public Readonly Property ReverseSplit As Boolean
```

**Visual C++**

```cpp
public:
    property bool ReverseSplit {
        bool get ();
    }
```
See Also

SecondaryBTreeDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
A class representing configuration parameters for SecondaryBTreeDatabase

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public class SecondaryBTreeDatabaseConfig : SecondaryDatabaseConfig

Visual Basic (Declaration)

Public Class SecondaryBTreeDatabaseConfig
    Inherits SecondaryDatabaseConfig

Visual C++

public ref class SecondaryBTreeDatabaseConfig : public SecondaryDatabaseConfig
Inheritance Hierarchy

`System..:::Object`
`BerkeleyDB..:::DatabaseConfig`
`BerkeleyDB..:::SecondaryDatabaseConfig`
`BerkeleyDB..:::SecondaryBTreeDatabaseConfig`
See Also

SecondaryBTreeDatabaseConfig Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `SecondaryBTreeDatabaseConfig` type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>SecondaryBTreeDatabaseConfig</code></td>
<td>Create a new SecondaryBTreeDatabaseConfig object</td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Equals</code></td>
<td>Determines whether the specified <code>Object</code> is equal to the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>GetHashCode</code></td>
<td>Serves as a hash function for a particular type. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>GetType</code></td>
<td>Gets the <code>Type</code> of the current instance. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>SetEncryption</code></td>
<td>Set the password and algorithm used by the Berkeley DB library to perform encryption and decryption. (Inherited from <code>DatabaseConfig</code>.)</td>
</tr>
<tr>
<td><code>SetForeignKeyConstraint</code></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><code>ToString</code></td>
<td>Returns a <code>String</code> that represents the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
</tbody>
</table>
## Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AutoCommit</td>
<td>Enclose the open call within a transaction. If the call succeeds, the open operation will be recoverable and all subsequent database modification operations based on this handle will be transactionally protected. If the call fails, no database will have been created. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td>ByteOrder</td>
<td>The byte order for integers in the stored database metadata. The host byte order of the machine where the Berkeley DB library was compiled is the default value. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td>CacheSize</td>
<td>The size of the shared memory buffer pool -- that is, the cache. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td>Compare</td>
<td>The Btree key comparison function.</td>
</tr>
<tr>
<td>Creation</td>
<td>The policy for how to handle database creation. If true, do checksum verification of pages read into the cache from the backing filestore. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td>DoChecksum</td>
<td>The duplicate data item comparison function. Policy for duplicate data items in the database; that is, insertion when the key of the key/data pair being inserted already exists in the database will be successful.</td>
</tr>
<tr>
<td>Env</td>
<td>The Berkeley DB environment within which to create a database. If null, the database will be created stand-alone; that is, it is not part of any Berkeley DB environment. (Inherited from DatabaseConfig.)</td>
</tr>
</tbody>
</table>
| Duplicates   | The mechanism for reporting error messages to the 
<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Inherited From</th>
</tr>
</thead>
<tbody>
<tr>
<td>ErrorFeedback</td>
<td>application.</td>
<td>DatabaseConfig</td>
</tr>
<tr>
<td>ErrorPrefix</td>
<td>The prefix string that appears before error messages issued by Berkeley DB.</td>
<td>Inherited from DatabaseConfig</td>
</tr>
<tr>
<td>Feedback</td>
<td>Cause the database object to be free-threaded; that is, concurrently usable by multiple threads in the address space.</td>
<td>DatabaseConfig</td>
</tr>
<tr>
<td>FreeThreaded</td>
<td>If true, the secondary key is immutable.</td>
<td>SecondaryDatabaseConfig</td>
</tr>
<tr>
<td>ImmutableKey</td>
<td>Do not map this database into process memory.</td>
<td>DatabaseConfig</td>
</tr>
<tr>
<td>NoMMap</td>
<td>If true, Berkeley DB will not write log records for this database.</td>
<td>DatabaseConfig</td>
</tr>
<tr>
<td>NonDurableTxns</td>
<td>Turn reverse splitting in the Btree on or off.</td>
<td>DatabaseConfig</td>
</tr>
<tr>
<td>Populate</td>
<td>If true and the secondary database is empty, walk through Primary and create an index to it in the empty secondary. This operation is potentially very expensive.</td>
<td>SecondaryDatabaseConfig</td>
</tr>
<tr>
<td>PrefixCompare</td>
<td>The Btree prefix function.</td>
<td>SecondaryDatabaseConfig</td>
</tr>
<tr>
<td>Priority</td>
<td>The cache priority for pages referenced by the database.</td>
<td>DatabaseConfig</td>
</tr>
<tr>
<td>ReadOnly</td>
<td>Open the database for reading only. Any attempt to modify items in the database will fail, regardless of the actual permissions of any underlying files.</td>
<td>DatabaseConfig</td>
</tr>
<tr>
<td>ReadUncommitted</td>
<td>Support transactional read operations with degree 1 isolation.</td>
<td>DatabaseConfig</td>
</tr>
<tr>
<td>Truncate</td>
<td>Physically truncate the underlying file, discarding all previous databases it might have held.</td>
<td></td>
</tr>
</tbody>
</table>
(Inherited from `DatabaseConfig`.)

- **UseMVCC**
  
  Open the database with support for multiversion concurrency control.
  
  (Inherited from `DatabaseConfig`.)

- **UseRecordNumbers**
  
  If true, support retrieval from the Btree using record numbers.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EncryptAlgorithm</td>
<td>The algorithm used to perform encryption and decryption. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td>EncryptionPassword</td>
<td>The password used to perform encryption and decryption. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td>ForeignKeyDatabase</td>
<td>(Inherited from SecondaryDatabaseConfig.)</td>
</tr>
<tr>
<td>ForeignKeyNullfiier</td>
<td>(Inherited from SecondaryDatabaseConfig.)</td>
</tr>
<tr>
<td>KeyGen</td>
<td>The delegate that creates the set of secondary keys corresponding to a given primary key and data pair. (Inherited from SecondaryDatabaseConfig.)</td>
</tr>
<tr>
<td>MinKeysPerPage</td>
<td>The minimum number of key/data pairs intended to be stored on any single Btree leaf page. (Inherited from SecondaryDatabaseConfig.)</td>
</tr>
<tr>
<td>OnForeignKeyDelete</td>
<td>(Inherited from SecondaryDatabaseConfig.)</td>
</tr>
<tr>
<td>PageSize</td>
<td>The size of the pages used to hold items in the database, in bytes. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td>Primary</td>
<td>All updates to Primary will be automatically reflected in the secondary and all reads from the secondary will return corresponding data from Primary. (Inherited from SecondaryDatabaseConfig.)</td>
</tr>
</tbody>
</table>
See Also

SecondaryBTreeDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Create a new SecondaryBTreeDatabaseConfig object

Namespace: BerkeleyDB
Syntax

C#

public SecondaryBTreeDatabaseConfig(
    Database PrimaryDB,
    SecondaryKeyGenDelegate KeyGenFunc
)

Visual Basic (Declaration)

Public Sub New ( _
    PrimaryDB As Database, _
    KeyGenFunc As SecondaryKeyGenDelegate _
)

Visual C++

public:
SecondaryBTreeDatabaseConfig(
    Database^ PrimaryDB,
    SecondaryKeyGenDelegate^ KeyGenFunc
)

Parameters

PrimaryDB
Type: BerkeleyDB::Database

KeyGenFunc
Type: BerkeleyDB::SecondaryKeyGenDelegate
See Also

SecondaryBTreeDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `SecondaryBTreeDatabaseConfig` type exposes the following members.
### Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AutoCommit</strong></td>
<td>Enclose the open call within a transaction. If the call succeeds, the open operation will be recoverable and all subsequent database modification operations based on this handle will be transactionally protected. If the call fails, no database will have been created. (Inherited from <a href="#">DatabaseConfig</a>).</td>
</tr>
<tr>
<td><strong>ByteOrder</strong></td>
<td>The byte order for integers in the stored database metadata. The host byte order of the machine where the Berkeley DB library was compiled is the default value. (Inherited from <a href="#">DatabaseConfig</a>).</td>
</tr>
<tr>
<td><strong>CacheSize</strong></td>
<td>The size of the shared memory buffer pool -- that is, the cache. (Inherited from <a href="#">DatabaseConfig</a>).</td>
</tr>
<tr>
<td><strong>Compare</strong></td>
<td>The Btree key comparison function.</td>
</tr>
<tr>
<td><strong>Creation</strong></td>
<td>The policy for how to handle database creation. If false, the database must be created explicitly prior to use. If true, the database will be created upon first use. (Inherited from <a href="#">DatabaseConfig</a>).</td>
</tr>
<tr>
<td><strong>DoChecksum</strong></td>
<td>If true, do checksum verification of pages read into the cache from the backing filestore. (Inherited from <a href="#">DatabaseConfig</a>).</td>
</tr>
<tr>
<td><strong>DuplicateCompare</strong></td>
<td>The duplicate data item comparison function. Policy for duplicate data items in the database; that is, insertion when the key of the key/data pair being inserted already exists in the database will be successful.</td>
</tr>
<tr>
<td><strong>Duplicates</strong></td>
<td>The Berkeley DB environment within which to create a database. If null, the database will be created stand-alone; that is, it is not part of any Berkeley DB environment. (Inherited from <a href="#">DatabaseConfig</a>).</td>
</tr>
</tbody>
</table>
| **Env**            | The mechanism for reporting error messages to the }
ErrorFeedback application.
(Inherited from DatabaseConfig.)

ErrorPrefix The prefix string that appears before error messages issued by Berkeley DB.
(Inherited from DatabaseConfig.)

Feedback Cause the database object to be free-threaded; that is, concurrently usable by multiple threads in the address space.
(Inherited from DatabaseConfig.)

FreeThreaded (Inherited from DatabaseConfig.)

ImmutableKey If true, the secondary key is immutable.
(Inherited from SecondaryDatabaseConfig.)

NoMMap Do not map this database into process memory.
(Inherited from DatabaseConfig.)

NonDurableTxns If true, Berkeley DB will not write log records for this database.
(Inherited from DatabaseConfig.)

NoReverseSplitting Turn reverse splitting in the Btree on or off.
If true and the secondary database is empty, walk through Primary and create an index to it in the empty secondary. This operation is potentially very expensive.
(Inherited from SecondaryDatabaseConfig.)

Populate If true and the secondary database is empty, walk through Primary and create an index to it in the empty secondary. This operation is potentially very expensive.
(Inherited from SecondaryDatabaseConfig.)

PrefixCompare The Btree prefix function.
(Inherited from SecondaryDatabaseConfig.)

Priority The cache priority for pages referenced by the database.
(Inherited from DatabaseConfig.)

ReadOnly Open the database for reading only. Any attempt to modify items in the database will fail, regardless of the actual permissions of any underlying files.
(Inherited from DatabaseConfig.)

ReadUncommitted Support transactional read operations with degree 1 isolation.
(Inherited from DatabaseConfig.)

Truncate Physically truncate the underlying file, discarding all previous databases it might have held.
(Inherited from [DatabaseConfig](#).)

- **UseMVCC**
  
  Open the database with support for multiversion concurrency control.
  
  (Inherited from [DatabaseConfig](#).)

- **UseRecordNumbers**
  
  If true, support retrieval from the Btree using record numbers.
See Also

SecondaryBTreeDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryBTreeDatabaseConfig::Compare Field

The Btree key comparison function.

Namespace: BerkeleyDB
Syntax

C#

public EntryComparisonDelegate Compare

Visual Basic (Declaration)

Public Compare As EntryComparisonDelegate

Visual C++

public:
EntryComparisonDelegate^ Compare
Remarks

The comparison function is called whenever it is necessary to compare a key specified by the application with a key currently stored in the tree.

If no comparison function is specified, the keys are compared lexically, with shorter keys collating before longer keys.

If the database already exists, the comparison function must be the same as that historically used to create the database or corruption can occur.
See Also

SecondaryBTreeDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryBTreeDatabaseConfig::Creation Field

The policy for how to handle database creation.

Namespace: BerkeleyDB
Syntax

C#

public CreatePolicy Creation

Visual Basic (Declaration)

Public Creation As CreatePolicy

Visual C++

public:
CreatePolicy Creation
Remarks

If the database does not already exist and NEVER is set, Open(String, SecondaryBTreeDatabaseConfig) will fail.
See Also

SecondaryBTreeDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryBTreeDatabaseConfig.DuplicateCompare

The duplicate data item comparison function.

Namespace: BerkeleyDB
Syntax

C#

public EntryComparisonDelegate DuplicateCompare

Visual Basic (Declaration)

Public DuplicateCompare As EntryComparisonDelegate

Visual C++

public:
EntryComparisonDelegate^ DuplicateCompare
Remarks

The comparison function is called whenever it is necessary to compare a data item specified by the application with a data item currently stored in the database. Setting DuplicateCompare implies setting Duplicates to SORTED.

If no comparison function is specified, the data items are compared lexically, with shorter data items collating before longer data items.

If the database already exists when Open(String, SecondaryBTreeDatabaseConfig) is called, the delegate must be the same as that historically used to create the database or corruption can occur.
See Also

SecondaryBTreeDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryBTreeDatabaseConfig Class

See Also

Policy for duplicate data items in the database; that is, insertion when the key of the key/data pair being inserted already exists in the database will be successful.

Namespace: BerkeleyDB
Syntax

**C#**

public DuplicatesPolicy Duplicates

**Visual Basic (Declaration)**

Public Duplicates As DuplicatesPolicy

**Visual C++**

public: DuplicatesPolicy Duplicates
Remarks

The ordering of duplicates in the database for **UNSORTED** is determined by the order of insertion, unless the ordering is otherwise specified by use of a cursor operation or a duplicate sort function. The ordering of duplicates in the database for **SORTED** is determined by the duplicate comparison function. If the application does not specify a comparison function using **DuplicateCompare**, a default lexical comparison will be used.

**SORTED** is preferred to **UNSORTED** for performance reasons. **UNSORTED** should only be used by applications wanting to order duplicate data items manually.

If the database already exists, the value of Duplicates must be the same as the existing database or an error will be returned.

It is an error to specify **UseRecordNumbers** and anything other than **NONE**.
See Also

SecondaryBTreeDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Turn reverse splitting in the Btree on or off.

Namespace: BerkeleyDB
**Syntax**

**C#**

```csharp
public bool NoReverseSplitting
```

**Visual Basic (Declaration)**

```vbnet
Public NoReverseSplitting As Boolean
```

**Visual C++**

```cpp
public:

bool NoReverseSplitting
```
Remarks

As pages are emptied in a database, the Berkeley DB Btree implementation attempts to coalesce empty pages into higher-level pages in order to keep the database as small as possible and minimize search time. This can hurt performance in applications with cyclical data demands; that is, applications where the database grows and shrinks repeatedly. For example, because Berkeley DB does page-level locking, the maximum level of concurrency in a database of two pages is far smaller than that in a database of 100 pages, so a database that has shrunk to a minimal size can cause severe deadlocking when a new cycle of data insertion begins.
See Also

SecondaryBTreeDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
C# | Visual Basic
Visual C++

Berkeley DB .NET API Documentation
SecondaryBTreeDatabaseConfig::<>::PrefixCompare Field

SecondaryBTreeDatabaseConfig Class  See Also

The Btree prefix function.

Namespace: BerkeleyDB
Syntax

C#
public EntryComparisonDelegate PrefixCompare

Visual Basic (Declaration)
Public PrefixCompare As EntryComparisonDelegate

Visual C++
public: EntryComparisonDelegate^ PrefixCompare
Remarks

The prefix function is used to determine the amount by which keys stored on the Btree internal pages can be safely truncated without losing their uniqueness. See the Btree prefix comparison section of the Berkeley DB Reference Guide for more details about how this works. The usefulness of this is data-dependent, but can produce significantly reduced tree sizes and search times in some data sets.

If no prefix function or key comparison function is specified by the application, a default lexical comparison function is used as the prefix function. If no prefix function is specified and Compare is specified, no prefix function is used. It is an error to specify a prefix function without also specifying Compare.

If the database already exists, the prefix function must be the same as that historically used to create the database or corruption can occur.
See Also

SecondaryBTreeDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, support retrieval from the Btree using record numbers.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll)  
**Version:** 4.8.24.0
### Syntax

**C#**

```csharp
public bool UseRecordNumbers
```

**Visual Basic (Declaration)**

```vbnet
Public UseRecordNumbers As Boolean
```

**Visual C++**

```cpp
public:
bool UseRecordNumbers
```
Remarks

Logical record numbers in Btree databases are mutable in the face of record insertion or deletion. See Renumber for further discussion.

Maintaining record counts within a Btree introduces a serious point of contention, namely the page locations where the record counts are stored. In addition, the entire database must be locked during both insertions and deletions, effectively single-threading the database for those operations. Specifying UseRecordNumbers can result in serious performance degradation for some applications and data sets.

It is an error to specify UseRecordNumbers and anything other than NONE.

If the database already exists, the value of UseRecordNumbers must be the same as the existing database or an error will be returned.
See Also

SecondaryBTreeDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `SecondaryBTreeDatabaseConfig` type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="" /> <strong>Equals</strong></td>
<td>Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><img src="image" alt="" /> <strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><img src="image" alt="" /> <strong>GetType</strong></td>
<td>Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><img src="image" alt="" /> <strong>SetEncryption</strong></td>
<td>Set the password and algorithm used by the Berkeley DB library to perform encryption and decryption. (Inherited from <a href="#">DatabaseConfig</a>.)</td>
</tr>
<tr>
<td><img src="image" alt="" /> <strong>SetForeignKeyConstraint</strong></td>
<td>Overloaded. Returns a <a href="#">String</a> that represents the current <a href="#">Object</a>.</td>
</tr>
<tr>
<td><img src="image" alt="" /> <strong>ToString</strong></td>
<td>(Inherited from <a href="#">Object</a>.)</td>
</tr>
</tbody>
</table>
See Also

SecondaryBTreeDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryBTreeDatabaseConfig.

SetForeignKeyConstraint Method

SecondaryBTreeDatabaseConfig Class  See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>SetForeignKeyConstraint(Database, ForeignKeyDeleteAction)</code></td>
<td>(Inherited from <code>SecondaryDatabaseConfig</code>.)</td>
</tr>
<tr>
<td>`SetForeignKeyConstraint(Database, ForeignKeyDeleteAction,</td>
<td>(Inherited from <code>SecondaryDatabaseConfig</code>.)</td>
</tr>
<tr>
<td>ForeignKeyNullifyDelegate)`</td>
<td></td>
</tr>
</tbody>
</table>
See Also

SecondaryBTreeDatabaseConfig Class
SecondaryBTreeDatabaseConfig Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The SecondaryBTreeDatabaseConfig type exposes the following members.
### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EncryptAlgorithm</td>
<td>The algorithm used to perform encryption and decryption. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td>EncryptionPassword</td>
<td>The password used to perform encryption and decryption. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td>ForeignKeyDatabase</td>
<td>(Inherited from SecondaryDatabaseConfig.)</td>
</tr>
<tr>
<td>ForeignKeyNullfier</td>
<td>(Inherited from SecondaryDatabaseConfig.)</td>
</tr>
<tr>
<td>KeyGen</td>
<td>The delegate that creates the set of secondary keys corresponding to a given primary key and data pair. (Inherited from SecondaryDatabaseConfig.)</td>
</tr>
<tr>
<td>MinKeysPerPage</td>
<td>The minimum number of key/data pairs intended to be stored on any single Btree leaf page. (Inherited from SecondaryDatabaseConfig.)</td>
</tr>
<tr>
<td>OnForeignKeyDelete</td>
<td>(Inherited from SecondaryDatabaseConfig.)</td>
</tr>
<tr>
<td>PageSize</td>
<td>The size of the pages used to hold items in the database, in bytes. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td>Primary</td>
<td>All updates to Primary will be automatically reflected in the secondary and all reads from the secondary will return corresponding data from Primary. (Inherited from SecondaryDatabaseConfig.)</td>
</tr>
</tbody>
</table>
See Also

SecondaryBTreeDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryBTreeDatabaseConfig.MinKeysPerPage Property

The minimum number of key/data pairs intended to be stored on any single Btree leaf page.

Namespace: BerkeleyDB
Syntax

C#

public uint MinKeysPerPage { get; set; }

Visual Basic (Declaration)

Public Property MinKeysPerPage AsUInteger

Visual C++

public:
property unsigned int MinKeysPerPage {
    unsigned int get ();
    void set (unsigned int value);
}
Remarks

This value is used to determine if key or data items will be stored on overflow pages instead of Btree leaf pages. For more information on the specific algorithm used, see the Berkeley DB Reference Guide. The value specified must be at least 2; if not explicitly set, a value of 2 is used.

If the database already exists, MinKeysPerPage will be ignored.
See Also

SecondaryBTreeDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
A class representing database cursors over secondary indexes, which allow for traversal of database records.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public class SecondaryCursor : BaseCursor,
    IEnumerable<KeyValuePair<DatabaseEntry>, KeyValuePair<DatabaseEntry>

Visual Basic (Declaration)

Public Class SecondaryCursor _
    Inherits BaseCursor _
    Implements IEnumerable(Of KeyValuePair(Of DatabaseEntry, KeyValuePair<DatabaseEntry>

Visual C++

public ref class SecondaryCursor : public BaseCursor,
    IEnumerable<KeyValuePair<DatabaseEntry, KeyValuePair<DatabaseEntry>
    IEnumerable
Inheritance Hierarchy

System:::Object
BerkeleyDB:::BaseCursor
BerkeleyDB:::SecondaryCursor
See Also

SecondaryCursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The **SecondaryCursor** type exposes the following members.
# Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Close</td>
<td>Discard the cursor. It is possible for the Close() method to throw a <code>DeadlockException</code>, signaling that any enclosing transaction should be aborted. If the application is already intending to abort the transaction, this error should be ignored, and the application should proceed. After Close has been called, regardless of its result, the object may not be used again. (Inherited from <code>BaseCursor</code>.)</td>
</tr>
<tr>
<td>Compare</td>
<td>Compare this cursor's position to another's. (Inherited from <code>BaseCursor</code>.)</td>
</tr>
<tr>
<td>Count</td>
<td>Returns a count of the number of data items for the key to which the cursor refers. (Inherited from <code>BaseCursor</code>.)</td>
</tr>
<tr>
<td>Delete</td>
<td>Delete the key/data pair to which the cursor refers from the primary database and all secondary indices. Release the resources held by this object, and close the cursor if it's still open. (Inherited from <code>BaseCursor</code>.)</td>
</tr>
<tr>
<td>Dispose</td>
<td>Create a new cursor that uses the same transaction and locker ID as the original cursor. Determines whether the specified <code>Object</code> is equal to the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>Equals</td>
<td>Returns an enumerator that iterates through the <code>SecondaryCursor</code>.</td>
</tr>
</tbody>
</table>
- **GetHashCode**  
  Serves as a hash function for a particular type.  
  (Inherited from **Object**.)

- **GetType**  
  Gets the **Type** of the current instance.  
  (Inherited from **Object**.)

- **Move**  
  Overloaded.

- **MoveFirst**  
  Overloaded.

- **MoveLast**  
  Overloaded.

- **MoveNext**  
  Overloaded.

- **MoveNextDuplicate**  
  Overloaded.

- **MoveNextUnique**  
  Overloaded.

- **MovePrev**  
  Overloaded.

- **MovePrevDuplicate**  
  Overloaded.

- **MovePrevUnique**  
  Overloaded.

- **Refresh**  
  Overloaded.

- **ToString**  
  Returns a **String** that represents the current **Object**.  
  (Inherited from **Object**.)
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current</td>
<td>The secondary key and primary key/data pair at which the cursor currently points.</td>
</tr>
</tbody>
</table>
See Also

SecondaryCursor Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The **SecondaryCursor** type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| Close | Discard the cursor.  
It is possible for the Close() method to throw a [DeadlockException](#), signaling that any enclosing transaction should be aborted. If the application is already intending to abort the transaction, this error should be ignored, and the application should proceed.  
After Close has been called, regardless of its result, the object may not be used again.  
(Inherited from [BaseCursor](#).) |
| Compare | Compare this cursor's position to another's.  
(Inherited from [BaseCursor](#).) |
| Count | Returns a count of the number of data items for the key to which the cursor refers.  
(Inherited from [BaseCursor](#).) |
| Delete | Delete the key/data pair to which the cursor refers from the primary database and all secondary indices.  
Release the resources held by this object, and close the cursor if it's still open.  
(Inherited from [BaseCursor](#).) |
| Dispose | Create a new cursor that uses the same transaction and locker ID as the original cursor.  
Determines whether the specified [Object](#) is equal to the current [Object](#).  
(Inherited from [Object](#).) |
| Duplicate | Returns an enumerator that iterates through the [SecondaryCursor](#). |
- **GetHashCode** Serves as a hash function for a particular type. (Inherited from **Object**.)
- **GetType** Gets the **Type** of the current instance. (Inherited from **Object**.)
- **Move** Overloaded.
- **MoveFirst** Overloaded.
- **MoveLast** Overloaded.
- **MoveNext** Overloaded.
- **MoveNextDuplicate** Overloaded.
- **MoveNextUnique** Overloaded.
- **MovePrev** Overloaded.
- **MovePrevDuplicate** Overloaded.
- **MovePrevUnique** Overloaded.
- **Refresh** Overloaded.
- **ToString** Returns a **String** that represents the current **Object**. (Inherited from **Object**.)
See Also

SecondaryCursor Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Delete Method

SecondaryCursor::Delete Method

Delete the key/data pair to which the cursor refers from the primary database and all secondary indices.

Namespace: BerkeleyDB
Syntax

C#

public void Delete()

Visual Basic (Declaration)

Public Sub Delete

Visual C++

public:
void Delete()
Remarks

The cursor position is unchanged after a delete, and subsequent calls to cursor functions expecting the cursor to refer to an existing key will fail.
## Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>BerkeleyDB:::KeyEmptyException</td>
<td>The element has already been deleted.</td>
</tr>
</tbody>
</table>
See Also

SecondaryCursor Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryCursor...::Duplicate Method

Create a new cursor that uses the same transaction and locker ID as the original cursor.

Namespace: BerkeleyDB
Syntax

C#

public  SecondaryCursor  Duplicate(
                      bool  keepPosition
)

Visual Basic (Declaration)

Public Function Duplicate ( _  
                      keepPosition As  Boolean  _  
) As  SecondaryCursor

Visual C++

public:
SecondaryCursor^ Duplicate(
                      bool  keepPosition
)

Parameters

keepPosition

Type:  System::::Boolean
If true, the newly created cursor is initialized to refer to the same position in
the database as the original cursor (if any) and hold the same locks (if any).
If false, or the original cursor does not hold a database position and locks,
the created cursor is uninitialized and will behave like a cursor newly
created by  Cursor().()().

Return Value

A newly created cursor
Remarks

This is useful when an application is using locking and requires two or more cursors in the same thread of control.
See Also

SecondaryCursor Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryCursor..::.GetEnumerator Method

SecondaryCursor Class  See Also

Returns an enumerator that iterates through the SecondaryCursor.

Namespace:  BerkeleyDB
**Syntax**

**C#**

```csharp
public IEnumerator<
KeyValuePair<
DatabaseEntry,
KeyValuePair<
DatabaseEntry
>
>
```

**Visual Basic (Decl)**

```
Public Function GetEnumerator As IEnumerator(Of KeyValuePair(Of DatabaseEntry)
```

**Visual C++**

```c
public:
virtual IEnumerator<
KeyValuePair<
DatabaseEntry^,
KeyValuePair<
DatabaseEntry
>
>
```

**ReturnValue**

An enumerator for the SecondaryCursor.

**Implements**

```c
IEnumerable<Of (T)>, ... GetEnumerator()
```
Remarks

The enumerator will begin at the cursor's current position (or the first record if the cursor has not yet been positioned) and iterate forwards (i.e. in the direction of `MoveNext()`) over the remaining records.
See Also

SecondaryCursor Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryCursor Class

See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Move(DatabaseEntry, Boolean)</strong></td>
<td>Set the cursor to refer to key, and store the primary key/data pair associated with the given secondary key in <strong>Current</strong>. In the presence of duplicate key values, the first data item in the set of duplicates is stored in <strong>Current</strong>.</td>
</tr>
<tr>
<td><strong>Move(KeyValuePair(Of &lt;(DatabaseEntry, KeyValuePair(Of &lt;(DatabaseEntry, DatabaseEntry&gt;&gt;&gt;), Boolean))</strong></td>
<td>Move the cursor to the specified key/data pair of the database. The cursor is positioned to a key/data pair if both the key and data match the values provided on the key and data parameters. Set the cursor to refer to key, and store the primary key/data pair associated with the given secondary key in <strong>Current</strong>. In the presence of duplicate key values, the first data item in the set of duplicates is stored in <strong>Current</strong>.</td>
</tr>
<tr>
<td><strong>Move(DatabaseEntry, Boolean, LockingInfo)</strong></td>
<td>Move the cursor to the specified key/data pair of the database. The cursor is positioned to a key/data pair if both the key and data match the values provided on the key and data parameters.</td>
</tr>
</tbody>
</table>
See Also

SecondaryCursor Class
SecondaryCursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryCursor Class  See Also

Set the cursor to refer to key, and store the primary key/data pair associated with the given secondary key in Current. In the presence of duplicate key values, the first data item in the set of duplicates is stored in Current.

Namespace:  BerkeleyDB
Syntax

C#

public bool Move(
    DatabaseEntry key,
    bool exact
)

Visual Basic (Declaration)

Public Function Move ( _
    key As DatabaseEntry, _
    exact As Boolean _
) As Boolean

Visual C++

public:
    bool Move(
        DatabaseEntry^ key,
        bool exact
    )

Parameters

key
Type: BerkeleyDB::DatabaseEntry
The key at which to position the cursor

exact
Type: System::Boolean
If true, require the given key to match the key in the database exactly. If false, position the cursor at the smallest key greater than or equal to the specified key, permitting partial key matches and range searches.

Return Value
True if the cursor was positioned successfully, false otherwise.
Remarks

If positioning the cursor fails, Current will contain an empty KeyValuePair(Of TKey, TValue>).
See Also

SecondaryCursor Class
Move Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Move the cursor to the specified key/data pair of the database. The cursor is positioned to a key/data pair if both the key and data match the values provided on the key and data parameters.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool Move(
    KeyValuePair<DatabaseEntry, KeyValuePair<DatabaseEntry, DatabaseEntry, bool>> exact
)

Visual Basic (Declaration)

Public Function Move (_
    pair As KeyValuePair(Of DatabaseEntry, KeyValuePair(Of DatabaseEntry, DatabaseEntry, Boolean, _
    exact As Boolean _
) As Boolean

Visual C++

public:
    bool Move(
        KeyValuePair<DatabaseEntry^, KeyValuePair<DatabaseEntry^, DatabaseEntry, bool> exact
    )

Parameters

pair
    Type: System.Collections.Generic:::KeyValuePair<Of <<(DatabaseEntry, KeyValuePair<DatabaseEntry, DatabaseEntry>>)>)
The key/data pair at which to position the cursor.

exact
    Type: System:::Boolean
    If true, require the given key and data to match the key and data in the database exactly. If false, position the cursor at the smallest data value which is greater than or equal to the value provided by pair.Value (as determined by the comparison function).
Return Value

True if the cursor was positioned successfully, false otherwise.
Remarks

If positioning the cursor fails, Current will contain an empty KeyValuePair<(Of <(TKey, TValue)>)>.

If this flag is specified on a database configured without sorted duplicate support, the value of exact is ignored.
See Also

SecondaryCursor Class
Move Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryCursor::Move Method (DatabaseEntry, Boolean, LockingInfo)

SecondaryCursor Class  See Also

Set the cursor to refer to key, and store the primary key/data pair associated with the given secondary key in Current. In the presence of duplicate key values, the first data item in the set of duplicates is stored in Current.

Namespace:  BerkeleyDB
Syntax

C#

public bool Move(
    DatabaseEntry key,
    bool exact,
    LockingInfo info
)

Visual Basic (Declaration)

Public Function Move ( _
    key As DatabaseEntry, _
    exact As Boolean, _
    info As LockingInfo _
) As Boolean

Visual C++

public:
    bool Move(
        DatabaseEntry^ key,
        bool exact,
        LockingInfo^ info
    )

Parameters

key
Type: BerkeleyDB::DatabaseEntry
The key at which to position the cursor

exact
Type: System::Boolean
If true, require the given key to match the key in the database exactly. If false, position the cursor at the smallest key greater than or equal to the specified key, permitting partial key matches and range searches.
info
    Type: BerkeleyDB::::LockingInfo
    The locking behavior to use.

Return Value

True if the cursor was positioned successfully, false otherwise.
Remarks

If positioning the cursor fails, `Current` will contain an empty `KeyValuePair<(Of <(TKey, TValue)>)>`. 
See Also

SecondaryCursor Class
Move Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryCursor..::.Move Method (KeyValuePair(Of KeyValuePair(Of DatabaseEntry, DatabaseEntry), Boolean, LockingInfo))

Move the cursor to the specified key/data pair of the database. The cursor is positioned to a key/data pair if both the key and data match the values provided on the key and data parameters.

**Namespace:** [BerkeleyDB](#)

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

### C#

```csharp
public bool Move(
    KeyValuePair<DatabaseEntry, KeyValuePair<DatabaseEntry, DatabaseEntry>> exact,
    LockingInfo info
)
```

### Visual Basic (Declaration)

```vbnet
Public Function Move (_
    pair As KeyValuePair(Of DatabaseEntry, KeyValuePair(Of DatabaseEntry, DatabaseEntry)) exact As Boolean, _
    info As LockingInfo _
) As Boolean
```

### Visual C++

```cpp
public: _
    bool Move(_
        KeyValuePair<DatabaseEntry^, KeyValuePair<DatabaseEntry^, DatabaseEntry>> exact,
        LockingInfo^ info
    )
```

### Parameters

**pair**

Type: `System.Collections.Generic::KeyValuePair<DatabaseEntry, KeyValuePair<DatabaseEntry, DatabaseEntry>>`

The key/data pair at which to position the cursor.

**exact**

Type: `System ::= Boolean`

If true, require the given key and data to match the key and data in the database exactly. If false, position the cursor at the smallest data value which is greater than or equal to the value provided by pair.Value (as
determined by the comparison function).

info
Type: BerkeleyDB::LockingInfo
The locking behavior to use.

**Return Value**

True if the cursor was positioned successfully, false otherwise.
Remarks

If positioning the cursor fails, Current will contain an empty KeyValuePair<(Of<TKey, TValue>)>.

If this flag is specified on a database configured without sorted duplicate support, the value of exact is ignored.
See Also

SecondaryCursor Class
Move Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryCursor::MoveFirst Method

SecondaryCursor Class  See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MoveFirst()()()</td>
<td>Set the cursor to refer to the first key/data pair of the database, and store the secondary key along with the corresponding primary key/data pair in Current. If the first key has duplicate values, the first data item in the set of duplicates is stored in Current.</td>
</tr>
<tr>
<td>MoveFirst(LockingInfo)</td>
<td>Set the cursor to refer to the first key/data pair of the database, and store the secondary key along with the corresponding primary key/data pair in Current. If the first key has duplicate values, the first data item in the set of duplicates is stored in Current.</td>
</tr>
</tbody>
</table>
See Also

SecondaryCursor Class
SecondaryCursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryCursor..::.MoveFirst Method

SecondaryCursor Class  See Also

Set the cursor to refer to the first key/data pair of the database, and store the secondary key along with the corresponding primary key/data pair in Current. If the first key has duplicate values, the first data item in the set of duplicates is stored in Current.

Namespace: BerkeleyDB
Syntax

C#

public bool MoveFirst()

Visual Basic (Declaration)

Public Function MoveFirst As Boolean

Visual C++

public:
bool MoveFirst()

Return Value

True if the cursor was positioned successfully, false otherwise.
Remarks

If positioning the cursor fails, Current will contain an empty KeyValuePair(Of (TKey, TValue)).
See Also

SecondaryCursor Class
MoveFirst Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryCursor..:.MoveFirst Method (LockingInfo)

SecondaryCursor Class  See Also

Set the cursor to refer to the first key/data pair of the database, and store the secondary key along with the corresponding primary key/data pair in Current. If the first key has duplicate values, the first data item in the set of duplicates is stored in Current.

Namespace:  BerkeleyDB
**Syntax**

**C#**

```csharp
public bool MoveFirst(
    LockingInfo info
)
```

**Visual Basic (Declaration)**

```vbnet
Public Function MoveFirst ( _
    info As LockingInfo _
) As Boolean
```

**Visual C++**

```cpp
public:
bool MoveFirst(
    LockingInfo^ info
)
```

**Parameters**

info

Type: BerkeleyDB::::LockingInfo

The locking behavior to use.

**Return Value**

True if the cursor was positioned successfully, false otherwise.
Remarks

If positioning the cursor fails, `Current` will contain an empty `KeyValuePair<Of <(TKey, TValue)>)>`. 
See Also

SecondaryCursor Class
MoveFirst Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryCursor Class

See Also
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MoveLast()()</td>
<td>Set the cursor to refer to the last key/data pair of the database, and store the secondary key and primary key/data pair in <strong>Current</strong>. If the last key has duplicate values, the last data item in the set of duplicates is stored in <strong>Current</strong>.</td>
</tr>
<tr>
<td>MoveLast(LockingInfo)</td>
<td>Set the cursor to refer to the last key/data pair of the database, and store the secondary key and primary key/data pair in <strong>Current</strong>. If the last key has duplicate values, the last data item in the set of duplicates is stored in <strong>Current</strong>.</td>
</tr>
</tbody>
</table>
See Also

SecondaryCursor Class
SecondaryCursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryCursor..::.MoveLast Method

SecondaryCursor Class  See Also

Set the cursor to refer to the last key/data pair of the database, and store the secondary key and primary key/data pair in Current. If the last key has duplicate values, the last data item in the set of duplicates is stored in Current.

Namespace:  BerkeleyDB
Syntax

C#

```csharp
public bool MoveLast()
```

Visual Basic (Declaration)

```vbnet
Public Function MoveLast As Boolean
```

Visual C++

```c++
public:
bool MoveLast()
```

Return Value

True if the cursor was positioned successfully, false otherwise.
Remarks

If positioning the cursor fails, `Current` will contain an empty `KeyValuePair<(Of <(TKey, TValue)>)>`. 
See Also

SecondaryCursor Class
MoveLast Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryCursor Class  See Also

Set the cursor to refer to the last key/data pair of the database, and store the secondary key and primary key/data pair in `Current`. If the last key has duplicate values, the last data item in the set of duplicates is stored in `Current`.

**Namespace:**  BerkeleyDB  
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

```csharp
public bool MoveLast(
    LockingInfo info
)
```

Visual Basic (Declaration)

```vbnet
Public Function MoveLast ( _
    info As LockingInfo _
) As Boolean
```

Visual C++

```c++
public:
    bool MoveLast(
        LockingInfo^ info
    )
```

Parameters

info

Type: BerkeleyDB::LockingInfo

The locking behavior to use.

Return Value

True if the cursor was positioned successfully, false otherwise.
Remarks

If positioning the cursor fails, **Current** will contain an empty **KeyValuePair**<**(TKey, TValue)>**.
See Also

SecondaryCursor Class
MoveLast Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryCursor Class

See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MoveNext()()</td>
<td>If the cursor is not yet initialized, MoveNext is identical to MoveFirst(). Otherwise, move the cursor to the next key/data pair of the database, and store the secondary key and primary key/data pair in Current. In the presence of duplicate key values, the value of Current.Key may not change.</td>
</tr>
<tr>
<td>MoveNext(LockingInfo)</td>
<td>If the cursor is not yet initialized, MoveNext is identical to MoveFirst(). Otherwise, move the cursor to the next key/data pair of the database, and store the secondary key and primary key/data pair in Current. In the presence of duplicate key values, the value of Current.Key may not change.</td>
</tr>
</tbody>
</table>
See Also

SecondaryCursor Class
SecondaryCursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If the cursor is not yet initialized, MoveNext is identical to MoveFirst(). Otherwise, move the cursor to the next key/data pair of the database, and store the secondary key and primary key/data pair in Current. In the presence of duplicate key values, the value of Current.Key may not change.

Namespace: BerkeleyDB
Syntax

C#

public bool MoveNext()

Visual Basic (Declaration)

Public Function MoveNext As Boolean

Visual C++

public:
  bool MoveNext()

Return Value

True if the cursor was positioned successfully, false otherwise.
Remarks

If positioning the cursor fails, Current will contain an empty KeyValuePair<(Of <(TKey, TValue)>)>.
See Also

SecondaryCursor Class
MoveNext Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If the cursor is not yet initialized, MoveNext is identical to `MoveFirst()`. Otherwise, move the cursor to the next key/data pair of the database, and store the secondary key and primary key/data pair in `Current`. In the presence of duplicate key values, the value of `Current.Key` may not change.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool MoveNext(
    LockingInfo info
)

Visual Basic (Declaration)

Public Function MoveNext ( _
    info As LockingInfo _
) As Boolean

Visual C++

public:
bool MoveNext(
    LockingInfo^ info
)

Parameters

info

    Type: BerkeleyDB::::LockingInfo
    The locking behavior to use.

Return Value

True if the cursor was positioned successfully, false otherwise.
Remarks

If positioning the cursor fails, **Current** will contain an empty **KeyValuePair<Of *(TKey, TValue*)>**.
See Also

SecondaryCursor Class
MoveNext Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryCursor..::.MoveNextDuplicate Method

SecondaryCursor Class  See Also
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MoveNextDuplicate()()()</td>
<td>If the next key/data pair of the database is a duplicate data record for the current key/data pair, move the cursor to the next key/data pair in the database, and store the secondary key and primary key/data pair in Current. MoveNextDuplicate will return false if the next key/data pair of the database is not a duplicate data record for the current key/data pair.</td>
</tr>
<tr>
<td>MoveNextDuplicate(LockingInfo)</td>
<td>If the next key/data pair of the database is a duplicate data record for the current key/data pair, move the cursor to the next key/data pair in the database, and store the secondary key and primary key/data pair in Current. MoveNextDuplicate will return false if the next key/data pair of the database is not a duplicate data record for the current key/data pair.</td>
</tr>
</tbody>
</table>
See Also

SecondaryCursor Class
SecondaryCursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If the next key/data pair of the database is a duplicate data record for the current key/data pair, move the cursor to the next key/data pair in the database, and store the secondary key and primary key/data pair in `Current`. `MoveNextDuplicate` will return false if the next key/data pair of the database is not a duplicate data record for the current key/data pair.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool MoveNextDuplicate()

Visual Basic (Declaration)

Public Function MoveNextDuplicate As Boolean

Visual C++

public:
bool MoveNextDuplicate()

Return Value

True if the cursor was positioned successfully, false otherwise.
Remarks

If positioning the cursor fails, Current will contain an empty KeyValuePair(Of (TKey, TValue>).
See Also

SecondaryCursor Class
MoveNextDuplicate Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If the next key/data pair of the database is a duplicate data record for the current key/data pair, move the cursor to the next key/data pair in the database, and store the secondary key and primary key/data pair in `Current`. MoveNextDuplicate will return false if the next key/data pair of the database is not a duplicate data record for the current key/data pair.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

```csharp
public bool MoveNextDuplicate(
    LockingInfo info
)
```

Visual Basic (Declaration)

```vbnet
Public Function MoveNextDuplicate ( _
    info As LockingInfo _
) As Boolean
```

Visual C++

```cpp
public:
    bool MoveNextDuplicate(
        LockingInfo^ info
    )
```

Parameters

info
Type: BerkeleyDB::LockingInfo
The locking behavior to use.

Return Value

True if the cursor was positioned successfully, false otherwise.
Remarks

If positioning the cursor fails, `Current` will contain an empty `KeyValuePair<Of `<(TKey, TValue)>`>`. 
See Also

SecondaryCursor Class
MoveNextDuplicate Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryCursor::MoveNextUnique Method

SecondaryCursor Class   See Also
### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MoveNextUnique()</strong></td>
<td>If the cursor is not yet initialized, MoveNextUnique is identical to MoveFirst(). Otherwise, move the cursor to the next non-duplicate key in the database, and store the secondary key and primary key/data pair in Current. MoveNextUnique will return false if no non-duplicate key/data pairs exist after the cursor position in the database.</td>
</tr>
<tr>
<td><strong>MoveNextUnique(LockingInfo)</strong></td>
<td>If the cursor is not yet initialized, MoveNextUnique is identical to MoveFirst(). Otherwise, move the cursor to the next non-duplicate key in the database, and store the secondary key and primary key/data pair in Current. MoveNextUnique will return false if no non-duplicate key/data pairs exist after the cursor position in the database.</td>
</tr>
</tbody>
</table>
See Also

SecondaryCursor Class
SecondaryCursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If the cursor is not yet initialized, MoveNextUnique is identical to `MoveFirst()`(). Otherwise, move the cursor to the next non-duplicate key in the database, and store the secondary key and primary key/data pair in `Current`. MoveNextUnique will return false if no non-duplicate key/data pairs exist after the cursor position in the database.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool MoveNextUnique()

Visual Basic (Declaration)

Public Function MoveNextUnique As Boolean

Visual C++

public:
bool MoveNextUnique()

Return Value

True if the cursor was positioned successfully, false otherwise.
Remarks

If positioning the cursor fails, Current will contain an empty KeyValuePair<(Of (TKey, TValue)>).
See Also

SecondaryCursor Class
MoveNextUnique Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If the cursor is not yet initialized, MoveNextUnique is identical to MoveFirst(). Otherwise, move the cursor to the next non-duplicate key in the database, and store the secondary key and primary key/data pair in Current. MoveNextUnique will return false if no non-duplicate key/data pairs exist after the cursor position in the database.

Namespace: BerkeleyDB
Syntax

C#

public bool MoveNextUnique(
    LockingInfo info
)

Visual Basic (Declaration)

Public Function MoveNextUnique ( 
    info As LockingInfo 
) As Boolean

Visual C++

public:
    bool MoveNextUnique( 
        LockingInfo^ info
    )

Parameters

info

Type: BerkeleyDB::::LockingInfo
The locking behavior to use.

Return Value

True if the cursor was positioned successfully, false otherwise.
Remarks

If the database is a Queue or Recno database, MoveNextUnique will ignore any keys that exist but were never explicitly created by the application, or those that were created and later deleted.

If positioning the cursor fails, Current will contain an empty KeyValuePair<(Of <(TKey, TValue)>)).
See Also

SecondaryCursor Class
MoveNextUnique Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryCursor::MovePrev Method

SecondaryCursor Class  See Also
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MovePrev()()</td>
<td>If the cursor is not yet initialized, MovePrev is identical to MoveLast(). Otherwise, move the cursor to the previous key/data pair of the database, and store the secondary key and primary key/data pair in Current. In the presence of duplicate key values, the value of Current.Key may not change. If the cursor is not yet initialized, MovePrev is identical to MoveLast(LockingInfo). Otherwise, move the cursor to the previous key/data pair of the database, and store the secondary key and primary key/data pair in Current. In the presence of duplicate key values, the value of Current.Key may not change.</td>
</tr>
<tr>
<td>MovePrev(LockingInfo)</td>
<td></td>
</tr>
</tbody>
</table>
See Also

SecondaryCursor Class
SecondaryCursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If the cursor is not yet initialized, MovePrev is identical to `MoveLast()`. Otherwise, move the cursor to the previous key/data pair of the database, and store the secondary key and primary key/data pair in `Current`. In the presence of duplicate key values, the value of `Current.Key` may not change.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool MovePrev()

Visual Basic (Declaration)

Public Function MovePrev As Boolean

Visual C++

public:
    bool MovePrev()

Return Value

True if the cursor was positioned successfully, false otherwise.
Remarks

If positioning the cursor fails, Current will contain an empty KeyValuePair<(Of <(TKey, TValue)>)>.
See Also

SecondaryCursor Class
MovePrev Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If the cursor is not yet initialized, MovePrev is identical to MoveLast(LockingInfo). Otherwise, move the cursor to the previous key/data pair of the database, and store the secondary key and primary key/data pair in Current. In the presence of duplicate key values, the value of Current.Key may not change.

Namespace: BerkeleyDB
Syntax

C#

```csharp
public bool MovePrev(
    LockingInfo info
)
```

**Visual Basic (Declaration)**

```vbnet
Public Function MovePrev (_
    info As LockingInfo _
) As Boolean
```

**Visual C++**

```cpp
public:
bool MovePrev(
    LockingInfo^ info
)
```

**Parameters**

info

Type: BerkeleyDB::LockingInfo

The locking behavior to use.

**Return Value**

True if the cursor was positioned successfully, false otherwise.
Remarks

If positioning the cursor fails, Current will contain an empty KeyValuePair<((TKey, TValue)>).
See Also

SecondaryCursor Class
MovePrev Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryCursor::MovePrevDuplicate Method

SecondaryCursor Class  See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MovePrevDuplicate()()</td>
<td>If the previous key/data pair of the database is a duplicate data record for the current key/data pair, the cursor is moved to the previous key/data pair of the database, and the secondary key and primary key/data pair in Current. MovePrevDuplicate will return false if the previous key/data pair of the database is not a duplicate data record for the current key/data pair.</td>
</tr>
<tr>
<td>MovePrevDuplicate(LockingInfo)</td>
<td>If the previous key/data pair of the database is a duplicate data record for the current key/data pair, the cursor is moved to the previous key/data pair of the database, and the secondary key and primary key/data pair in Current. MovePrevDuplicate will return false if the previous key/data pair of the database is not a duplicate data record for the current key/data pair.</td>
</tr>
</tbody>
</table>
See Also

SecondaryCursor Class
SecondaryCursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If the previous key/data pair of the database is a duplicate data record for the current key/data pair, the cursor is moved to the previous key/data pair of the database, and the secondary key and primary key/data pair in Current. MovePrevDuplicate will return false if the previous key/data pair of the database is not a duplicate data record for the current key/data pair.

Namespace: BerkeleyDB
Syntax

C#

public bool MovePrevDuplicate()

Visual Basic (Declaration)

Public Function MovePrevDuplicate As Boolean

Visual C++

public:
bool MovePrevDuplicate()

Return Value

True if the cursor was positioned successfully, false otherwise.
Remarks

If positioning the cursor fails, `Current` will contain an empty `KeyValuePair<Of <(TKey, TValue)>)>`. 
See Also

SecondaryCursor Class
MovePrevDuplicate Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If the previous key/data pair of the database is a duplicate data record for the current key/data pair, the cursor is moved to the previous key/data pair of the database, and the secondary key and primary key/data pair in `Current`. MovePrevDuplicate will return false if the previous key/data pair of the database is not a duplicate data record for the current key/data pair.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

```csharp
public bool MovePrevDuplicate(
    LockingInfo info
)
```

Visual Basic (Declaration)

```vbnet
Public Function MovePrevDuplicate ( _
    info As LockingInfo _
) As Boolean
```

Visual C++

```cpp
public:
    bool MovePrevDuplicate(
        LockingInfo^ info
    )
```

Parameters

`info`

Type: BerkeleyDB::::LockingInfo

The locking behavior to use.

Return Value

True if the cursor was positioned successfully, false otherwise.
Remarks

If positioning the cursor fails, `Current` will contain an empty `KeyValuePair<Of<TKey, TValue>>`. 
See Also

SecondaryCursor Class
MovePrevDuplicate Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryCursor::MovePrevUnique Method

SecondaryCursor Class  See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MovePrevUnique()()</td>
<td>If the cursor is not yet initialized, MovePrevUnique is identical to MoveLast(). Otherwise, move the cursor to the previous non-duplicate key in the database, and store the secondary key and primary key/data pair in Current. MovePrevUnique will return false if no non-duplicate key/data pairs exist after the cursor position in the database.</td>
</tr>
<tr>
<td>MovePrevUnique(LockingInfo)</td>
<td>If the cursor is not yet initialized, MovePrevUnique is identical to MoveLast(LockingInfo). Otherwise, move the cursor to the previous non-duplicate key in the database, and store the secondary key and primary key/data pair in Current. MovePrevUnique will return false if no non-duplicate key/data pairs exist after the cursor position in the database.</td>
</tr>
</tbody>
</table>
See Also

SecondaryCursor Class
SecondaryCursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If the cursor is not yet initialized, MovePrevUnique is identical to MoveLast(). Otherwise, move the cursor to the previous non-duplicate key in the database, and store the secondary key and primary key/data pair in Current. MovePrevUnique will return false if no non-duplicate key/data pairs exist after the cursor position in the database.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool MovePrevUnique()

Visual Basic (Declaration)

Public Function MovePrevUnique As Boolean

Visual C++

public:
bool MovePrevUnique()

Return Value

True if the cursor was positioned successfully, false otherwise.
Remarks

If positioning the cursor fails, Current will contain an empty KeyValuePair<(Of <(TKey, TValue)>)>.
See Also

SecondaryCursor Class
MovePrevUnique Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryCursor..::.MovePrevUnique Method (LockingInfo)

SecondaryCursor Class  See Also

If the cursor is not yet initialized, MovePrevUnique is identical to MoveLast(LockingInfo). Otherwise, move the cursor to the previous non-duplicate key in the database, and store the secondary key and primary key/data pair in Current. MovePrevUnique will return false if no non-duplicate key/data pairs exist after the cursor position in the database.

Namespace:  BerkeleyDB
Syntax

**C#**

```csharp
public bool MovePrevUnique(
    LockingInfo info
)
```

**Visual Basic (Declaration)**

```vbnet
Public Function MovePrevUnique (_
    info As LockingInfo _
) As Boolean
```

**Visual C++**

```cpp
public:
bool MovePrevUnique(
    LockingInfo^ info
)
```

**Parameters**

- `info`
  - Type: `BerkeleyDB::LockingInfo`
  - The locking behavior to use.

**Return Value**

- True if the cursor was positioned successfully, false otherwise.
Remarks

If positioning the cursor fails, `Current` will contain an empty `KeyValuePair<Of <(TKey, TValue)>`).
See Also

SecondaryCursor Class
MovePrevUnique Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryCursor::Refresh Method

See Also

SecondaryCursor Class
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Refresh()()</td>
<td>Store the secondary key and primary key/data pair to which the cursor refers in Current.</td>
</tr>
<tr>
<td>Refresh(LockingInfo)</td>
<td>Store the secondary key and primary key/data pair to which the cursor refers in Current.</td>
</tr>
</tbody>
</table>
See Also

SecondaryCursor Class
SecondaryCursor Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Store the secondary key and primary key/data pair to which the cursor refers in `Current`.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
### Syntax

**C#**

```csharp
public bool Refresh()
```

**Visual Basic (Declaration)**

Public Function Refresh As Boolean

**Visual C++**

```cpp
public:
bool Refresh()
```

### Return Value

True if the cursor was positioned successfully, false otherwise.
Remarks

If positioning the cursor fails, **Current** will contain an empty `KeyValuePair(Of (TKey, TValue)>).`
See Also

SecondaryCursor Class
Refresh Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryCursor Class

See Also

Store the secondary key and primary key/data pair to which the cursor refers in Current.

Namespace: BerkeleyDB
Syntax

C#

public bool Refresh(
    LockingInfo info
)

Visual Basic (Declaration)

Public Function Refresh ( _
    info As LockingInfo _
) As Boolean

Visual C++

public:
    bool Refresh(
        LockingInfo^ info
    )

Parameters

info
    Type: BerkeleyDB::::LockingInfo
    The locking behavior to use.

Return Value

True if the cursor was positioned successfully, false otherwise.
Remarks

If positioning the cursor fails, Current will contain an empty KeyValuePair(Of (TKey, TValue)>).
See Also

SecondaryCursor Class
Refresh Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `SecondaryCursor` type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current</td>
<td>The secondary key and primary key/data pair at which the cursor currently points.</td>
</tr>
</tbody>
</table>
See Also

SecondaryCursor Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryCursor..::.Current Property

SecondaryCursor Class  See Also

The secondary key and primary key/data pair at which the cursor currently points.

Namespace:  BerkeleyDB  
## Syntax

**C#**

```csharp
public KeyValuePair<DatabaseEntry, KeyValuePair<DatabaseEntry, KeyValuePair<DatabaseEntry, DatabaseEntry>>>
```

**Visual Basic (Declaration)**

Public Property Current As KeyValuePair(Of DatabaseEntry, KeyValuePair<DatabaseEntry, DatabaseEntry>)

**Visual C++**

```cpp
public:
property KeyValuePair<DatabaseEntry^, KeyValuePair<DatabaseEntry^, KeyValuePair<DatabaseEntry^, DatabaseEntry>>
void set (KeyValuePair<DatabaseEntry^, KeyValuePair<DatabaseEntry^, DatabaseEntry>>)
```
See Also

SecondaryCursor Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
A class representing a secondary Berkeley DB database, a base class for access method specific classes.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public class SecondaryDatabase : BaseDatabase

Visual Basic (Declaration)

Public Class SecondaryDatabase _
    Inherits BaseDatabase

Visual C++

public ref class SecondaryDatabase : public BaseDatabase
Inheritance Hierarchy

System::<Object
BerkeleyDB::<BaseDatabase
BerkeleyDB::<SecondaryDatabase
   BerkeleyDB::<SecondaryBTreeDatabase
   BerkeleyDB::<SecondaryHashDatabase
   BerkeleyDB::<SecondaryQueueDatabase
   BerkeleyDB::<SecondaryRecnoDatabase
See Also

SecondaryDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The *SecondaryDatabase* type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overloaded.</td>
<td>Although closing a database will close any open cursors, it is recommended that applications explicitly close all their Cursor objects before closing the database. The reason why is that when the cursor is explicitly closed, the memory allocated for it is reclaimed; however, this will not happen if you close a database while cursors are still opened. The same rule, for the same reasons, hold true for Transaction objects. Simply make sure you resolve all your transaction objects before closing your database handle. Because key/data pairs are cached in memory, applications should make a point to always either close database handles or sync their data to disk (using SyncQQ) before exiting, to ensure that any data cached in main memory are reflected in the underlying file system. When called on a database that is the primary database for a secondary index, the primary database should be closed only after all secondary indices referencing it have been closed. When multiple threads are using the object concurrently, only a single thread may call the Close method. The object may not be accessed again after Close is called, regardless of its outcome.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Cursor</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>Delete</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>Dispose</td>
<td>Release the resources held by this object, and close the database if it's still open. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>Exists</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>Get</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>GetBoth</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as a hash function for a particular type. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>Open</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>PrintFastStats</td>
<td>The statistical information is described by the BTreeStats, HashStats, QueueStats, and RecnoStats classes.</td>
</tr>
<tr>
<td>PrintStats</td>
<td>The statistical information is described by the BTreeStats, HashStats, QueueStats, and RecnoStats classes.</td>
</tr>
<tr>
<td>SecondaryCursor</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>Sync</td>
<td>Flush any cached information to disk. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a String that represents the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Truncate</td>
<td>When called on a database configured with secondary indices, Truncate will truncate the primary database and all secondary indices. A count of the records discarded from the primary database is returned.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>-------------</td>
</tr>
<tr>
<td><strong>AutoCommit</strong></td>
<td>If true, all database modification operations based on this object will be transactionally protected. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>CacheSize</strong></td>
<td>The size of the shared memory buffer pool -- that is, the cache. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>Creation</strong></td>
<td>The CreatePolicy with which this database was opened. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>DatabaseName</strong></td>
<td>The name of this database, if it has one. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>DoChecksum</strong></td>
<td>If true, do checksum verification of pages read into the cache from the backing filestore. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>EncryptAlgorithm</strong></td>
<td>The algorithm used by the Berkeley DB library to perform encryption and decryption. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>Encrypted</strong></td>
<td>If true, encrypt all data stored in the database. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>Endianness</strong></td>
<td>The database byte order. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>ErrorFeedback</strong></td>
<td>The mechanism for reporting detailed error messages to the application. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>ErrorPrefix</strong></td>
<td>The prefix string that appears before error messages issued by Berkeley DB. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>Feedback</strong></td>
<td>Monitor progress within long running operations. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>FileName</strong></td>
<td>The filename of this database, if it has one. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>FreeThreaded</strong></td>
<td>If true, the object is free-threaded; that is, concurrently usable by multiple threads in the address space.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>HasMultiple</strong></td>
<td>If true, the object references a physical file supporting multiple databases.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>InHostOrder</strong></td>
<td>If true, the underlying database files were created on an architecture of the same byte order as the current one. This information may be used to determine whether application data needs to be adjusted for this architecture or not.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>KeyGen</strong></td>
<td>The delegate that creates the set of secondary keys corresponding to a given primary key and data pair.</td>
</tr>
<tr>
<td><strong>NoMMap</strong></td>
<td>If true, this database is not mapped into process memory.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>NonDurableTxns</strong></td>
<td>If true, Berkeley DB will not write log records for this database.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>Nullifier</strong></td>
<td>The database's current page size.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>Priority</strong></td>
<td>The cache priority for pages referenced by this object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>ReadOnly</strong></td>
<td>If true, this database has been opened for reading only. Any attempt to modify items in the database will fail, regardless of the actual permissions of any underlying files.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td></td>
<td>If true, this database supports transactional read</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>ReadUncommitted</strong></td>
<td>Read operations on the database may request the return of modified but not yet committed data. (Inherited from <strong>BaseDatabase</strong>.)</td>
</tr>
<tr>
<td><strong>Transactional</strong></td>
<td>If true, this database has been opened in a transactional mode. (Inherited from <strong>BaseDatabase</strong>.)</td>
</tr>
<tr>
<td><strong>Truncated</strong></td>
<td>If true, the underlying file was physically truncated upon open, discarding all previous databases it might have held. (Inherited from <strong>BaseDatabase</strong>.)</td>
</tr>
<tr>
<td><strong>Type</strong></td>
<td>The type of the underlying access method (and file format). This value may be used to determine the type of the database after an <strong>Open(String, DatabaseConfig)</strong>. (Inherited from <strong>BaseDatabase</strong>.)</td>
</tr>
<tr>
<td><strong>UseMVCC</strong></td>
<td>If true, the database was opened with support for multiversion concurrency control. (Inherited from <strong>BaseDatabase</strong>.)</td>
</tr>
</tbody>
</table>
See Also

SecondaryDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The SecondaryDatabase type exposes the following members.
Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overloaded.</td>
<td>Although closing a database will close any open cursors, it is recommended that applications explicitly close all their Cursor objects before closing the database. The reason why is that when the cursor is explicitly closed, the memory allocated for it is reclaimed; however, this will not happen if you close a database while cursors are still opened.</td>
</tr>
</tbody>
</table>

The same rule, for the same reasons, hold true for **Transaction** objects. Simply make sure you resolve all your transaction objects before closing your database handle.

Because key/data pairs are cached in memory, applications should make a point to always either close database handles or sync their data to disk (using **SyncOOO** before exiting, to ensure that any data cached in main memory are reflected in the underlying file system.

When called on a database that is the primary database for a secondary index, the primary database should be closed only after all secondary indices referencing it have been closed.

When multiple threads are using the object concurrently, only a single thread may call the Close method.

The object may not be accessed again after Close is called, regardless of its outcome.
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Dispose</strong></td>
<td>Release the resources held by this object, and close the database if it's still open. (Inherited from <code>BaseDatabase</code>.)</td>
</tr>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <code>Object</code> is equal to the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>Exists</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>Get</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>GetBoth</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Gets the <code>Type</code> of the current instance. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>Open</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>PrintFastStats</strong></td>
<td>The statistical information is described by the <code>BTreeStats</code>, <code>HashStats</code>, <code>QueueStats</code>, and <code>RecnoStats</code> classes. Overloaded.</td>
</tr>
<tr>
<td><strong>PrintStats</strong></td>
<td>The statistical information is described by the <code>BTreeStats</code>, <code>HashStats</code>, <code>QueueStats</code>, and <code>RecnoStats</code> classes.</td>
</tr>
<tr>
<td><strong>SecondaryCursor</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>Sync</strong></td>
<td>Flush any cached information to disk. (Inherited from <code>BaseDatabase</code>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a <code>String</code> that represents the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>Truncate</strong></td>
<td>When called on a database configured with secondary indices, Truncate will truncate the primary database and all secondary indices. A count of the records discarded from the primary database is returned.</td>
</tr>
</tbody>
</table>
See Also

SecondaryDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Although closing a database will close any open cursors, it is recommended that applications explicitly close all their Cursor objects before closing the database. The reason why is that when the cursor is explicitly closed, the memory allocated for it is reclaimed; however, this will not happen if you close a database while cursors are still opened.

The same rule, for the same reasons, hold true for Transaction objects. Simply make sure you resolve all your transaction objects before closing your database handle.

Because key/data pairs are cached in memory, applications should make a point to always either close database handles or sync their data to disk (using Sync() before exiting, to ensure that any data cached in main memory are reflected in the underlying file system.

When called on a database that is the primary database for a secondary index, the primary database should be closed only after all secondary indices referencing it have been closed.

When multiple threads are using the object concurrently, only a single thread may call the Close method.

The object may not be accessed again after Close is called, regardless of its outcome.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Close()()</td>
<td>Flush any cached database information to disk, close any open <code>Cursor()()</code> objects, free any allocated resources, and close any underlying files. (Inherited from <code>BaseDatabase</code>.)</td>
</tr>
<tr>
<td>Close(Boolean)</td>
<td>Optionally flush any cached database information to disk, close any open <code>Cursor()()</code> objects, free any allocated resources, and close any underlying files. (Inherited from <code>BaseDatabase</code>.)</td>
</tr>
</tbody>
</table>
See Also

SecondaryDatabase Class
SecondaryDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryDatabase::Cursor Method

SecondaryDatabase Class  See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Cursor()()</code></td>
<td>Create a database cursor. (Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
<tr>
<td><code>Cursor(CursorConfig)</code></td>
<td>Create a database cursor with the given configuration. (Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
<tr>
<td><code>Cursor(Transaction)</code></td>
<td>Create a transactionally protected database cursor. (Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
<tr>
<td><code>Cursor(CursorConfig, Transaction)</code></td>
<td>Create a transactionally protected database cursor with the given configuration. (Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
</tbody>
</table>
See Also

SecondaryDatabase Class
SecondaryDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryDatabase.

SecondaryDatabase..::.Delete Method

SecondaryDatabase Class  See Also
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delete(DatabaseEntry)</td>
<td>Remove key/data pairs from the database. The key/data pair associated with key is discarded from the database. In the presence of duplicate key values, all records associated with the designated key will be discarded. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>Delete(DatabaseEntry, Transaction)</td>
<td>Remove key/data pairs from the database. The key/data pair associated with key is discarded from the database. In the presence of duplicate key values, all records associated with the designated key will be discarded. (Inherited from BaseDatabase.)</td>
</tr>
</tbody>
</table>
See Also

SecondaryDatabase Class
SecondaryDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryDatabase.exists Method

SecondaryDatabase Class  See Also
# Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exists(DatabaseEntry)</td>
<td>Check whether key appears in the database. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>Exists(DatabaseEntry, Transaction)</td>
<td>Check whether key appears in the database. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>Exists(DatabaseEntry, Transaction, LockingInfo)</td>
<td>Check whether key appears in the database. (Inherited from BaseDatabase.)</td>
</tr>
</tbody>
</table>
See Also

SecondaryDatabase Class
SecondaryDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryDatabase Method

SecondaryDatabase Class See Also

C# Visual Basic Visual C++ Include Protected Members Include Inherited Members Berkeley DB .NET API Documentation

SecondaryDatabase..::.Get Method

SecondaryDatabase Class See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Inherited from</th>
</tr>
</thead>
<tbody>
<tr>
<td>Get(DatabaseEntry)</td>
<td>Retrieve a key/data pair from the database. In the presence of duplicate key values, Get will return the first data item for key. (Inherited from <a href="#">BaseDatabase</a>.)</td>
<td><a href="#">BaseDatabase</a></td>
</tr>
<tr>
<td>Get(DatabaseEntry, Transaction)</td>
<td>Retrieve a key/data pair from the database. In the presence of duplicate key values, Get will return the first data item for key.</td>
<td><a href="#">BaseDatabase</a></td>
</tr>
<tr>
<td>Get(DatabaseEntry, Transaction, LockingInfo)</td>
<td>Retrieve a key/data pair from the database. In the presence of duplicate key values, Get will return the first data item for key.</td>
<td><a href="#">BaseDatabase</a></td>
</tr>
</tbody>
</table>
See Also

SecondaryDatabase Class
SecondaryDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryDatabase Class

See Also
### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetBoth(DatabaseEntry, DatabaseEntry)</td>
<td>Retrieve a key/data pair from the database which matches key and data. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>GetBoth(DatabaseEntry, DatabaseEntry, Transaction)</td>
<td>Retrieve a key/data pair from the database which matches key and data. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>GetBoth(DatabaseEntry, DatabaseEntry, Transaction, LockingInfo)</td>
<td>Retrieve a key/data pair from the database which matches key and data. (Inherited from BaseDatabase.)</td>
</tr>
</tbody>
</table>
See Also

SecondaryDatabase Class
SecondaryDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryDatabase.

See Also

SecondaryDatabase Class  See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Open(String, SecondaryDatabaseConfig)</code></td>
<td>Instantiate a new <code>SecondaryDatabase</code> object, open the database represented by <code>Filename</code> and associate the database with the primary index. The file specified by <code>Filename</code> must exist.</td>
</tr>
<tr>
<td><code>Open(String, SecondaryDatabaseConfig, Transaction)</code></td>
<td>Instantiate a new <code>SecondaryDatabase</code> object, open the database represented by <code>Filename</code> and associate the database with the primary index. The file specified by <code>Filename</code> must exist.</td>
</tr>
<tr>
<td><code>Open(String, String, SecondaryDatabaseConfig)</code></td>
<td>Instantiate a new <code>SecondaryDatabase</code> object, open the database represented by <code>Filename</code> and associate the database with the primary index. The file specified by <code>Filename</code> must exist.</td>
</tr>
<tr>
<td><code>Open(String, String, SecondaryDatabaseConfig, Transaction)</code></td>
<td>Instantiate a new <code>SecondaryDatabase</code> object, open the database represented by <code>Filename</code> and associate the database with the primary index. The file specified by <code>Filename</code> must exist.</td>
</tr>
</tbody>
</table>
See Also

SecondaryDatabase Class
SecondaryDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation
SecondaryDatabase::Open Method (String, SecondaryDatabaseConfig)

SecondaryDatabase Class  See Also

Instantiate a new SecondaryDatabase object, open the database represented by Filename and associate the database with the primary index. The file specified by Filename must exist.

Namespace:  BerkeleyDB
### Syntax

**C#**

```csharp
public static SecondaryDatabase Open(
    string Filename,
    SecondaryDatabaseConfig cfg
)
```

**Visual Basic (Declaration)**

```vbnet
Public Shared Function Open ( 
    Filename As String,
    cfg As SecondaryDatabaseConfig
) As SecondaryDatabase
```

**Visual C++**

```cpp
public:
static SecondaryDatabase^ Open(
    String^ Filename,
    SecondaryDatabaseConfig^ cfg
)
```

### Parameters

**Filename**
- **Type:** `System::::String`
- The name of an underlying file that will be used to back the database.

**cfg**
- **Type:** `BerkeleyDB::::SecondaryDatabaseConfig`
- The database's configuration

### Return Value

A new, open database object
Remarks

If `AutoCommit` is set, the operation will be implicitly transaction protected. Note that transactionally protected operations on a database object requires the object itself be transactionally protected during its open.
See Also

SecondaryDatabase Class
Open Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryDatabase..::.Open Method (String, SecondaryDatabaseConfig, Transaction)

SecondaryDatabase Class  See Also

Instantiate a new SecondaryDatabase object, open the database represented by Filename and associate the database with the primary index. The file specified by Filename must exist.

Namespace: BerkeleyDB
Syntax

C#

```csharp
public static SecondaryDatabase Open(
    string Filename,
    SecondaryDatabaseConfig cfg,
    Transaction txn
)
```

Visual Basic (Declaration)

```vbnet
Public Shared Function Open ( _
    Filename As String, _
    cfg As SecondaryDatabaseConfig, _
    txn As Transaction _
) As SecondaryDatabase
```

Visual C++

```cpp
public:
static SecondaryDatabase^ Open(
    String^ Filename,
    SecondaryDatabaseConfig^ cfg,
    Transaction^ txn
)
```

Parameters

Filename
   Type: System::String
   The name of an underlying file that will be used to back the database.

cfg
   Type: BerkeleyDB::SecondaryDatabaseConfig
   The database's configuration

txn
   Type: BerkeleyDB::Transaction
If the operation is part of an application-specified transaction, `txn` is a Transaction object returned from `beginTransaction()`; if the operation is part of a Berkeley DB Concurrent Data Store group, `txn` is a handle returned from `beginCDSGroup()`; otherwise null.

**Return Value**

A new, open database object
Remarks

If AutoCommit is set, the operation will be implicitly transaction protected. Note that transactionally protected operations on a database object requires the object itself be transactionally protected during its open.
See Also

SecondaryDatabase Class
Open Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryDatabase..::.Open Method (String, String, SecondaryDatabaseConfig)

**SecondaryDatabase Class**  [See Also](#)

Instantiate a new `SecondaryDatabase` object, open the database represented by `Filename` and associate the database with the `primary index`. The file specified by `Filename` must exist.

**Namespace:**  [BerkeleyDB](#)

**Assembly:**  `libdb_dotnet48` (in `libdb_dotnet48.dll`) Version: 4.8.24.0
**Syntax**

**C#**

```csharp
public static SecondaryDatabase Open(
    string Filename,
    string DatabaseName,
    SecondaryDatabaseConfig cfg
)
```

**Visual Basic (Declaration)**

```vbnet
Public Shared Function Open (_
    Filename As String, _
    DatabaseName As String, _
    cfg As SecondaryDatabaseConfig _
) As SecondaryDatabase
```

**Visual C++**

```cpp
public:
static SecondaryDatabase^ Open(
    String^ Filename,
    String^ DatabaseName,
    SecondaryDatabaseConfig^ cfg
)
```

**Parameters**

**Filename**

Type: `System::String`

The name of an underlying file that will be used to back the database.

**DatabaseName**

Type: `System::String`

This parameter allows applications to have multiple databases in a single file. Although no DatabaseName needs to be specified, it is an error to attempt to open a second database in a file that was not initially created using a database name.
cfg
   Type: BerkeleyDB::SecondaryDatabaseConfig
   The database's configuration

Return Value

A new, open database object
Remarks

If Filename is null and DatabaseName is non-null, the database can be opened by other threads of control and will be replicated to client sites in any replication group.

If AutoCommit is set, the operation will be implicitly transaction protected. Note that transactionally protected operations on a database object requires the object itself be transactionally protected during its open.
See Also

SecondaryDatabase Class
Open Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Instantiate a new SecondaryDatabase object, open the database represented by Filename and associate the database with the primary index. The file specified by Filename must exist.

Namespace: BerkeleyDB
Syntax

C#

```csharp
public static SecondaryDatabase Open(
    string Filename,
    string DatabaseName,
    SecondaryDatabaseConfig cfg,
    Transaction txn
)
```

Visual Basic (Declaration)

```vbnet
Public Shared Function Open ( _
    Filename As String, _
    DatabaseName As String, _
    cfg As SecondaryDatabaseConfig, _
    txn As Transaction _
) As SecondaryDatabase
```

Visual C++

```cpp
public:
static SecondaryDatabase^ Open(
    String^ Filename,
    String^ DatabaseName,
    SecondaryDatabaseConfig^ cfg,
    Transaction^ txn
)
```

Parameters

Filename
Type: `System::String`
The name of an underlying file that will be used to back the database.

DatabaseName
Type: `System::String`
This parameter allows applications to have multiple databases in a single file. Although no DatabaseName needs to be specified, it is an error to
attempt to open a second database in a file that was not initially created using a database name.

cfg
Type: BerkeleyDB:::SecondaryDatabaseConfig
The database's configuration

txn
Type: BerkeleyDB:::Transaction
If the operation is part of an application-specified transaction, txn is a Transaction object returned from BeginTransaction(); if the operation is part of a Berkeley DB Concurrent Data Store group, txn is a handle returned from BeginCDSGroup(); otherwise null.

Return Value

A new, open database object
Remarks

If Filename is null and DatabaseName is non-null, the database can be opened by other threads of control and will be replicated to client sites in any replication group.

If AutoCommit is set, the operation will be implicitly transaction protected. Note that transactionally protected operations on a database object requires the object itself be transactionally protected during its open.
See Also

SecondaryDatabase Class  
Open Overload  
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryDatabase..::.PrintFastStats Method

SecondaryDatabase Class  See Also

The statistical information is described by the BTreeStats, HashStats, QueueStats, and RecnoStats classes.
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PrintFastStats()()()</td>
<td>Display the database statistical information which does not require traversal of the database. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>PrintFastStats(Boolean)</td>
<td>Display the database statistical information which does not require traversal of the database. (Inherited from BaseDatabase.)</td>
</tr>
</tbody>
</table>
See Also

SecondaryDatabase Class
SecondaryDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryDatabase.

PrintStats Method

SecondaryDatabase Class  See Also

The statistical information is described by the BTreeStats, HashStats, QueueStats, and RecnoStats classes.
# Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>PrintStats()</code></td>
<td>Display the database statistical information. (Inherited from <code>BaseDatabase</code>.)</td>
</tr>
<tr>
<td><code>PrintStats(Boolean)</code></td>
<td>Display the database statistical information. (Inherited from <code>BaseDatabase</code>.)</td>
</tr>
</tbody>
</table>
See Also

SecondaryDatabase Class
SecondaryDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>SecondaryCursor()</code></td>
<td>Create a secondary database cursor.</td>
</tr>
<tr>
<td><code>SecondaryCursor(CursorConfig)</code></td>
<td>Create a secondary database cursor with the given configuration.</td>
</tr>
<tr>
<td><code>SecondaryCursor(Transaction)</code></td>
<td>Create a transactionally protected secondary database cursor.</td>
</tr>
<tr>
<td><code>SecondaryCursor(CursorConfig, Transaction)</code></td>
<td>Create a transactionally protected secondary database cursor with the given configuration.</td>
</tr>
</tbody>
</table>
See Also

SecondaryDatabase Class
SecondaryDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Create a secondary database cursor.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

```csharp
public SecondaryCursor SecondaryCursor()
```

Visual Basic (Declaration)

```vbnet
Public Function SecondaryCursor As SecondaryCursor
```

Visual C++

```c++
public: 
SecondaryCursor^ SecondaryCursor()
```

Return Value

A newly created cursor
See Also

SecondaryDatabase Class
SecondaryCursor Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryDatabase..::.SecondaryCursor Method (CursorConfig)

SecondaryDatabase Class  See Also

Create a secondary database cursor with the given configuration.

Namespace:  BerkeleyDB
Syntax

C#

```csharp
public SecondaryCursor SecondaryCursor(CursorConfig cfg)
```

Visual Basic (Declaration)

```vbnet
Public Function SecondaryCursor(_
    cfg As CursorConfig _
) As SecondaryCursor
```

Visual C++

```cpp
public:
SecondaryCursor^ SecondaryCursor(_
    CursorConfig^ cfg
)
```

Parameters

cfg

Type: BerkeleyDB::::CursorConfig
The configuration properties for the cursor.

ReturnValue

A newly created cursor
See Also

SecondaryDatabase Class
SecondaryCursor Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Create a transactionally protected secondary database cursor.

**Namespace:** [BerkeleyDB](https://www.berdex.com)

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

```csharp
public SecondaryCursor SecondaryCursor(
    Transaction txn
)
```

Visual Basic (Declaration)

```vbnet
Public Function SecondaryCursor (_
    As Transaction _
) As SecondaryCursor
```

Visual C++

```cpp
public:
SecondaryCursor‡ SecondaryCursor(
    Transaction‡ txn
)
```

Parameters

`txn`

Type: `BerkeleyDB::Transaction`

The transaction context in which the cursor may be used.

Return Value

A newly created cursor
See Also

SecondaryDatabase Class
SecondaryCursor Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Create a transactionally protected secondary database cursor with the given configuration.

**Namespace:**  BerkeleyDB

**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

```csharp
public SecondaryCursor SecondaryCursor(
    CursorConfig cfg,
    Transaction txn
)
```

Visual Basic (Declaration)

```vbnet
Public Function SecondaryCursor ( _
    cfg As CursorConfig, _
    txn As Transaction _
) As SecondaryCursor
```

Visual C++

```cpp
public:
SecondaryCursor^ SecondaryCursor(
    CursorConfig^ cfg,
    Transaction^ txn
)
```

Parameters

cfg
Type: BerkeleyDB::::CursorConfig
The configuration properties for the cursor.

txn
Type: BerkeleyDB::::Transaction
The transaction context in which the cursor may be used.

Return Value

A newly created cursor
See Also

SecondaryDatabase Class
SecondaryCursor Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
When called on a database configured with secondary indices, Truncate will truncate the primary database and all secondary indices. A count of the records discarded from the primary database is returned.
# Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>TruncateQQ()</code></td>
<td>Empty the database, discarding all records it contains. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><code>Truncate(Transaction)</code></td>
<td>Empty the database, discarding all records it contains. (Inherited from BaseDatabase.)</td>
</tr>
</tbody>
</table>
See Also

SecondaryDatabase Class
SecondaryDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `SecondaryDatabase` type exposes the following members.
### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AutoCommit</strong></td>
<td>If true, all database modification operations based on this object will be transactionally protected. <em>(Inherited from <a href="#">BaseDatabase</a>).</em></td>
</tr>
<tr>
<td><strong>CacheSize</strong></td>
<td>The size of the shared memory buffer pool -- that is, the cache. <em>(Inherited from <a href="#">BaseDatabase</a>).</em></td>
</tr>
<tr>
<td><strong>Creation</strong></td>
<td>The CreatePolicy with which this database was opened. <em>(Inherited from <a href="#">BaseDatabase</a>).</em></td>
</tr>
<tr>
<td><strong>DatabaseName</strong></td>
<td>The name of this database, if it has one. <em>(Inherited from <a href="#">BaseDatabase</a>).</em></td>
</tr>
<tr>
<td><strong>DoChecksum</strong></td>
<td>If true, do checksum verification of pages read into the cache from the backing filestore. <em>(Inherited from <a href="#">BaseDatabase</a>).</em></td>
</tr>
<tr>
<td><strong>EncryptAlgorithm</strong></td>
<td>The algorithm used by the Berkeley DB library to perform encryption and decryption. <em>(Inherited from <a href="#">BaseDatabase</a>).</em></td>
</tr>
<tr>
<td><strong>Encrypted</strong></td>
<td>If true, encrypt all data stored in the database. <em>(Inherited from <a href="#">BaseDatabase</a>).</em></td>
</tr>
<tr>
<td><strong>Endianness</strong></td>
<td>The database byte order. <em>(Inherited from <a href="#">BaseDatabase</a>).</em></td>
</tr>
<tr>
<td><strong>ErrorFeedback</strong></td>
<td>The mechanism for reporting detailed error messages to the application. <em>(Inherited from <a href="#">BaseDatabase</a>).</em></td>
</tr>
<tr>
<td><strong>ErrorPrefix</strong></td>
<td>The prefix string that appears before error messages issued by Berkeley DB. <em>(Inherited from <a href="#">BaseDatabase</a>).</em></td>
</tr>
<tr>
<td><strong>Feedback</strong></td>
<td>Monitor progress within long running operations. <em>(Inherited from <a href="#">BaseDatabase</a>).</em></td>
</tr>
<tr>
<td><strong>FileName</strong></td>
<td>The filename of this database, if it has one. <em>(Inherited from <a href="#">BaseDatabase</a>).</em></td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>FreeThreaded</td>
<td>If true, the object is free-threaded; that is, concurrently usable by multiple threads in the address space. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>HasMultiple</td>
<td>If true, the object references a physical file supporting multiple databases. (Inherited from BaseDatabase.) If true, the underlying database files were created on an architecture of the same byte order as the current one. This information may be used to determine whether application data needs to be adjusted for this architecture or not. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>InHostOrder</td>
<td>The delegate that creates the set of secondary keys corresponding to a given primary key and data pair.</td>
</tr>
<tr>
<td>KeyGen</td>
<td>If true, this database is not mapped into process memory.</td>
</tr>
<tr>
<td>NoMMap</td>
<td>See MMapSize for further information. (Inherited from BaseDatabase.) If true, Berkeley DB will not write log records for this database. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>NonDurableTxns</td>
<td>The database's current page size. (Inherited from BaseDatabase.) The cache priority for pages referenced by this object. (Inherited from BaseDatabase.) If true, this database has been opened for reading only. Any attempt to modify items in the database will fail, regardless of the actual permissions of any underlying files. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>Nullifier</td>
<td></td>
</tr>
<tr>
<td>Pagesize</td>
<td></td>
</tr>
<tr>
<td>Priority</td>
<td></td>
</tr>
<tr>
<td>ReadOnly</td>
<td></td>
</tr>
</tbody>
</table>
**ReadUncommitted** operations with degree 1 isolation. Read operations on the database may request the return of modified but not yet committed data. (Inherited from BaseDatabase.)

If true, this database has been opened in a transactional mode. (Inherited from BaseDatabase.)

If true, the underlying file was physically truncated upon open, discarding all previous databases it might have held. (Inherited from BaseDatabase.)

The type of the underlying access method (and file format). This value may be used to determine the type of the database after an Open(String, DatabaseConfig). (Inherited from BaseDatabase.)

If true, the database was opened with support for multiversion concurrency control. (Inherited from BaseDatabase.)
See Also

SecondaryDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The delegate that creates the set of secondary keys corresponding to a given primary key and data pair.

**Namespace:**  BerkeleyDB  
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
C#

```csharp
public SecondaryKeyGenDelegate KeyGen { get; private set; }
```

Visual Basic (Declaration)

```vbnet
Public Property KeyGen As SecondaryKeyGenDelegate
```

Visual C++

```cpp
public:
    property SecondaryKeyGenDelegate^ KeyGen {
        SecondaryKeyGenDelegate^ get ()
        void set (SecondaryKeyGenDelegate^ value);
    }
```
See Also

SecondaryDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryDatabase Class

Namespace: BerkeleyDB
**Syntax**

**C#**

```csharp
public ForeignKeyNullifyDelegate Nullifier { get; private set; }
```

**Visual Basic (Declaration)**

Public Property Nullifier As ForeignKeyNullifyDelegate

**Visual C++**

```cpp
public:
property ForeignKeyNullifyDelegate^ Nullifier {
ForeignKeyNullifyDelegate^ get ();
void set (ForeignKeyNullifyDelegate^ value);
}
```
See Also

SecondaryDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
A class representing configuration parameters for SecondaryDatabase

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public class SecondaryDatabaseConfig : DatabaseConfig

Visual Basic (Declaration)

Public Class SecondaryDatabaseConfig_
    Inherits DatabaseConfig

Visual C++

public ref class SecondaryDatabaseConfig : public DatabaseConfig
Inheritance Hierarchy

System::Object
BerkeleyDB::DatabaseConfig
BerkeleyDB::SecondaryDatabaseConfig
   BerkeleyDB::SecondaryBTreeDatabaseConfig
   BerkeleyDB::SecondaryHashDatabaseConfig
   BerkeleyDB::SecondaryQueueDatabaseConfig
   BerkeleyDB::SecondaryRecnoDatabaseConfig
See Also

SecondaryDatabaseConfig Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryDatabaseConfig Members

See Also

The SecondaryDatabaseConfig type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SecondaryDatabaseConfig</strong></td>
<td>Instantiate a new SecondaryDatabaseConfig object, with the default configuration settings.</td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html">Object</a> is equal to the current <a href="https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html">Object</a>. (Inherited from <a href="https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <a href="https://docs.oracle.com/javase/7/docs/api/java/lang/Class.html">Type</a> of the current instance. (Inherited from <a href="https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html">Object</a>.)</td>
</tr>
<tr>
<td><strong>SetEncryption</strong></td>
<td>Set the password and algorithm used by the Berkeley DB library to perform encryption and decryption. (Inherited from <a href="https://docs.oracle.com/javase/7/docs/api/com/sun/ds/db/jdbc/config/DatabaseConfig.html">DatabaseConfig</a>.)</td>
</tr>
<tr>
<td><strong>SetForeignKeyConstraint</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a <a href="https://docs.oracle.com/javase/7/docs/api/java/lang/String.html">String</a> that represents the current <a href="https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html">Object</a>. (Inherited from <a href="https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html">Object</a>.)</td>
</tr>
</tbody>
</table>
# Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AutoCommit</strong></td>
<td>Enclose the open call within a transaction. If the call succeeds, the open operation will be recoverable and all subsequent database modification operations based on this handle will be transactionally protected. If the call fails, no database will have been created. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td><strong>ByteOrder</strong></td>
<td>The byte order for integers in the stored database metadata. The host byte order of the machine where the Berkeley DB library was compiled is the default value. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td><strong>CacheSize</strong></td>
<td>The size of the shared memory buffer pool -- that is, the cache. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td><strong>DoChecksum</strong></td>
<td>If true, do checksum verification of pages read into the cache from the backing filestore. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td><strong>Env</strong></td>
<td>The Berkeley DB environment within which to create a database. If null, the database will be created stand-alone; that is, it is not part of any Berkeley DB environment. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td><strong>ErrorFeedback</strong></td>
<td>The mechanism for reporting error messages to the application. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td><strong>ErrorPrefix</strong></td>
<td>The prefix string that appears before error messages issued by Berkeley DB. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td><strong>Feedback</strong></td>
<td>Cause the database object to be free-threaded; that is, concurrently usable by multiple threads in the</td>
</tr>
<tr>
<td>Feature</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>FreeThreaded</td>
<td>address space. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td>ImmutableKey</td>
<td>If true, the secondary key is immutable.</td>
</tr>
<tr>
<td>NoMMap</td>
<td>Do not map this database into process memory. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td>NonDurableTxns</td>
<td>If true, Berkeley DB will not write log records for this database.</td>
</tr>
<tr>
<td>Populate</td>
<td>If true and the secondary database is empty, walk through Primary and create an index to it in the empty secondary. This operation is potentially very expensive.</td>
</tr>
<tr>
<td>Priority</td>
<td>The cache priority for pages referenced by the database. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td>ReadOnly</td>
<td>Open the database for reading only. Any attempt to modify items in the database will fail, regardless of the actual permissions of any underlying files. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td>ReadUncommitted</td>
<td>Support transactional read operations with degree 1 isolation.</td>
</tr>
<tr>
<td>Truncate</td>
<td>Physically truncate the underlying file, discarding all previous databases it might have held. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td>UseMVCC</td>
<td>Open the database with support for multiversion concurrency control.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from DatabaseConfig.)</td>
</tr>
</tbody>
</table>
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EncryptAlgorithm</td>
<td>The algorithm used to perform encryption and decryption. (Inherited from <a href="#">DatabaseConfig</a>)</td>
</tr>
<tr>
<td>EncryptionPassword</td>
<td>The password used to perform encryption and decryption. (Inherited from <a href="#">DatabaseConfig</a>)</td>
</tr>
<tr>
<td>ForeignKeyDatabase</td>
<td></td>
</tr>
<tr>
<td>ForeignKeyNullifier</td>
<td></td>
</tr>
<tr>
<td>KeyGen</td>
<td>The delegate that creates the set of secondary keys corresponding to a given primary key and data pair.</td>
</tr>
<tr>
<td>OnForeignKeyDelete</td>
<td></td>
</tr>
<tr>
<td>PageSize</td>
<td>The size of the pages used to hold items in the database, in bytes. (Inherited from <a href="#">DatabaseConfig</a>)</td>
</tr>
</tbody>
</table>
| Primary                   | All updates to Primary will be automatically reflected in the secondary and all reads from the secondary will return corresponding data from Primary.
See Also

SecondaryDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Instantiate a new SecondaryDatabaseConfig object, with the default configuration settings.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
## Syntax

### C#

```csharp
public SecondaryDatabaseConfig(
    Database PrimaryDB,
    SecondaryKeyGenDelegate KeyGenFunc
)
```

### Visual Basic (Declaration)

```vbnet
Public Sub New ( _
    PrimaryDB As Database, _
    KeyGenFunc As SecondaryKeyGenDelegate _
)
```

### Visual C++

```cpp
public:
SecondaryDatabaseConfig( 
    Database^ PrimaryDB,
    SecondaryKeyGenDelegate^ KeyGenFunc
)
```

## Parameters

### PrimaryDB
- **Type:** [BerkeleyDB::Database](#)

### KeyGenFunc
- **Type:** [BerkeleyDB::SecondaryKeyGenDelegate](#)
See Also

SecondaryDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `SecondaryDatabaseConfig` type exposes the following members.
## Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AutoCommit</strong></td>
<td>Enclose the open call within a transaction. If the call succeeds, the open operation will be recoverable and all subsequent database modification operations based on this handle will be transactionally protected. If the call fails, no database will have been created. (Inherited from <a href="#">DatabaseConfig</a>).</td>
</tr>
<tr>
<td><strong>ByteOrder</strong></td>
<td>The byte order for integers in the stored database metadata. The host byte order of the machine where the Berkeley DB library was compiled is the default value. (Inherited from <a href="#">DatabaseConfig</a>).</td>
</tr>
<tr>
<td><strong>CacheSize</strong></td>
<td>The size of the shared memory buffer pool -- that is, the cache. (Inherited from <a href="#">DatabaseConfig</a>).</td>
</tr>
<tr>
<td><strong>DoChecksum</strong></td>
<td>If true, do checksum verification of pages read into the cache from the backing filestore. (Inherited from <a href="#">DatabaseConfig</a>).</td>
</tr>
<tr>
<td><strong>Env</strong></td>
<td>The Berkeley DB environment within which to create a database. If null, the database will be created stand-alone; that is, it is not part of any Berkeley DB environment. (Inherited from <a href="#">DatabaseConfig</a>).</td>
</tr>
<tr>
<td><strong>ErrorFeedback</strong></td>
<td>The mechanism for reporting error messages to the application. (Inherited from <a href="#">DatabaseConfig</a>).</td>
</tr>
<tr>
<td><strong>ErrorPrefix</strong></td>
<td>The prefix string that appears before error messages issued by Berkeley DB. (Inherited from <a href="#">DatabaseConfig</a>).</td>
</tr>
<tr>
<td><strong>Feedback</strong></td>
<td>Cause the database object to be free-threaded; that is, concurrently usable by multiple threads in the</td>
</tr>
<tr>
<td><strong>FreeThreaded</strong></td>
<td>address space. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td><strong>ImmutableKey</strong></td>
<td>If true, the secondary key is immutable.</td>
</tr>
<tr>
<td><strong>NoMMap</strong></td>
<td>Do not map this database into process memory. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td><strong>NonDurableTxns</strong></td>
<td>If true, Berkeley DB will not write log records for this database. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td><strong>Populate</strong></td>
<td>If true and the secondary database is empty, walk through Primary and create an index to it in the empty secondary. This operation is potentially very expensive.</td>
</tr>
<tr>
<td><strong>Priority</strong></td>
<td>The cache priority for pages referenced by the database. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td><strong>ReadOnly</strong></td>
<td>Open the database for reading only. Any attempt to modify items in the database will fail, regardless of the actual permissions of any underlying files. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td><strong>ReadUncommitted</strong></td>
<td>Support transactional read operations with degree 1 isolation. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td><strong>Truncate</strong></td>
<td>Physically truncate the underlying file, discarding all previous databases it might have held. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td><strong>UseMVCC</strong></td>
<td>Open the database with support for multiversion concurrency control. (Inherited from DatabaseConfig.)</td>
</tr>
</tbody>
</table>
See Also

SecondaryDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, the secondary key is immutable.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#
public bool ImmutableKey

Visual Basic (Declaration)
Public ImmutableKey As Boolean

Visual C++
public:
bool ImmutableKey
Remarks

This setting can be used to optimize updates when the secondary key in a primary record will never be changed after the primary record is inserted. For immutable secondary keys, a best effort is made to avoid calling the secondary callback function when primary records are updated. This optimization may reduce the overhead of update operations significantly if the callback function is expensive.

Be sure to specify this setting only if the secondary key in the primary record is never changed. If this rule is violated, the secondary index will become corrupted, that is, it will become out of sync with the primary.
See Also

SecondaryDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true and the secondary database is empty, walk through Primary and create an index to it in the empty secondary. This operation is potentially very expensive.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool Populate

Visual Basic (Declaration)

Public Populate As Boolean

Visual C++

public:
bool Populate
Remarks

If the secondary database has been opened in an environment configured with transactions, the entire secondary index creation is performed in the context of a single transaction.

Care should be taken not to use a newly-populated secondary database in another thread of control until `Open(String, SecondaryDatabaseConfig)` has returned successfully in the first thread.

If transactions are not being used, care should be taken not to modify a primary database being used to populate a secondary database, in another thread of control, until `Open(String, SecondaryDatabaseConfig)` has returned successfully in the first thread. If transactions are being used, Berkeley DB will perform appropriate locking and the application need not do any special operation ordering.
See Also

SecondaryDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation
SecondaryDatabaseConfig Methods

See Also

The SecondaryDatabaseConfig type exposes the following members.
### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <code>Object</code> is equal to the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <code>Type</code> of the current instance. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>SetEncryption</strong></td>
<td>Set the password and algorithm used by the Berkeley DB library to perform encryption and decryption. (Inherited from <code>DatabaseConfig</code>.)</td>
</tr>
<tr>
<td><strong>SetForeignKeyConstraint</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a <code>String</code> that represents the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
</tbody>
</table>
See Also

SecondaryDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryDatabaseConfig::SetForeignKeyConstraint Method

SecondaryDatabaseConfig Class  See Also
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>SetForeignKeyConstraint(Database, ForeignKeyDeleteAction)</code></td>
<td></td>
</tr>
<tr>
<td><code>SetForeignKeyConstraint(Database, ForeignKeyDeleteAction, ForeignKeyNullifyDelegate)</code></td>
<td></td>
</tr>
</tbody>
</table>
See Also

SecondaryDatabaseConfig Class
SecondaryDatabaseConfig Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryDatabaseConfig..::.SetForeignKeyConstraint Method (Database, ForeignKeyDeleteAction)

SecondaryDatabaseConfig Class  See Also

Namespace:  BerkeleyDB
Syntax

C#

public void SetForeignKeyConstraint(
    Database ForeignDB,
    ForeignKeyDeleteAction OnDelete
)

Visual Basic (Declaration)

Public Sub SetForeignKeyConstraint ( _
    ForeignDB As Database, _
    OnDelete As ForeignKeyDeleteAction _
)

Visual C++

public:
void SetForeignKeyConstraint(
    Database^ ForeignDB,
    ForeignKeyDeleteAction OnDelete
)

Parameters

ForeignDB
Type: BerkeleyDB::Database

OnDelete
Type: BerkeleyDB::ForeignKeyDeleteAction
See Also

SecondaryDatabaseConfig Class
SetForeignKeyConstraint Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryDatabaseConfig::SetForeignKeyConstraint Method (Database, ForeignKeyDeleteAction, ForeignKeyNullifyDelegate)

SecondaryDatabaseConfig Class  See Also

Namespace: BerkeleyDB
Syntax

C#

```csharp
public void SetForeignKeyConstraint(
    Database ForeignDB,
    ForeignKeyDeleteAction OnDelete,
    ForeignKeyNullifyDelegate NullifyFunc
);
```

Visual Basic (Declaration)

```vbnet
Public Sub SetForeignKeyConstraint ( _
    ForeignDB As Database, _
    OnDelete As ForeignKeyDeleteAction, _
    NullifyFunc As ForeignKeyNullifyDelegate _
)
```

Visual C++

```cpp
public:
void SetForeignKeyConstraint(
    Database^ ForeignDB,
    ForeignKeyDeleteAction OnDelete,
    ForeignKeyNullifyDelegate^ NullifyFunc
)
```

Parameters

ForeignDB
  Type: `BerkeleyDB::Database`

OnDelete
  Type: `BerkeleyDB::ForeignKeyDeleteAction`

NullifyFunc
  Type: `BerkeleyDB::ForeignKeyNullifyDelegate`
See Also

SecondaryDatabaseConfig Class
SetForeignKeyConstraint Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `SecondaryDatabaseConfig` type exposes the following members.
# Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EncryptAlgorithm</td>
<td>The algorithm used to perform encryption and decryption. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td>EncryptionPassword</td>
<td>The password used to perform encryption and decryption. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td>ForeignKeyDatabase</td>
<td>FOREIGN KEY DATABASE FUNCTION</td>
</tr>
<tr>
<td>ForeignKeyNullifier</td>
<td>FOREIGN KEY NULLIFIER FUNCTION</td>
</tr>
<tr>
<td>KeyGen</td>
<td>The delegate that creates the set of secondary keys corresponding to a given primary key and data pair.</td>
</tr>
<tr>
<td>OnForeignKeyDelete</td>
<td>The size of the pages used to hold items in the database, in bytes. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td>PageSize</td>
<td>All updates to Primary will be automatically reflected in the secondary and all reads from the secondary will return corresponding data from Primary.</td>
</tr>
<tr>
<td>Primary</td>
<td></td>
</tr>
</tbody>
</table>
See Also

SecondaryDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryDatabaseConfig..::.ForeignKeyDatabase Property

SecondaryDatabaseConfig Class  See Also

Namespace:  BerkeleyDB
Syntax

**C#**

```csharp
public Database ForeignKeyDatabase { get; }
```

**Visual Basic (Declaration)**

```vbnet
Public ReadOnly Property ForeignKeyDatabase As Database
```

**Visual C++**

```cpp
public:
    property Database^ ForeignKeyDatabase {
        Database^ get ();
    }
```
See Also

SecondaryDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryDatabaseConfig::ForeignKeyNullfier Property

Namespace: BerkeleyDB
Syntax

C#

public ForeignKeyNullifyDelegate ForeignKeyNullfier { get; }

Visual Basic (Declaration)

Public ReadOnly Property ForeignKeyNullfier As ForeignKeyNullifyDelegate

Visual C++

public:
    property ForeignKeyNullifyDelegate^ ForeignKeyNullfier {
        ForeignKeyNullifyDelegate^ get ();
    }

See Also

SecondaryDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The delegate that creates the set of secondary keys corresponding to a given primary key and data pair.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

```csharp
public SecondaryKeyGenDelegate KeyGen { get; set; }
```

Visual Basic (Declaration)

```vbnet
Public Property KeyGen As SecondaryKeyGenDelegate
```

Visual C++

```cpp
public:
property SecondaryKeyGenDelegate^ KeyGen {
SecondaryKeyGenDelegate^ get ();
void set (SecondaryKeyGenDelegate^ value);
}
```
Remarks

KeyGen may be null if both Primary.ReadOnly and ReadOnly are true.
See Also

SecondaryDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryDatabaseConfig.:::OnForeignKeyDelete Property

SecondaryDatabaseConfig Class  See Also

Namespace: BerkeleyDB
### Syntax

**C#**

```csharp
public ForeignKeyDeleteAction OnForeignKeyDelete { get; }
```

**Visual Basic (Declaration)**

```vbnet
Public ReadOnly Property OnForeignKeyDelete As ForeignKeyDeleteAction
```

**Visual C++**

```cpp
public:
    property ForeignKeyDeleteAction OnForeignKeyDelete {
        ForeignKeyDeleteAction get ();
    }
```
See Also

SecondaryDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
All updates to Primary will be automatically reflected in the secondary and all reads from the secondary will return corresponding data from Primary.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public Database Primary { get; set; }

Visual Basic (Declaration)

Public Property Primary As Database

Visual C++

public:
property Database^ Primary {
    Database^ get ();
    void set (Database^ value);
}
Remarks

Note that as primary keys must be unique for secondary indices to work, Primary must have been configured with NONE.
See Also

SecondaryDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
A class representing a SecondaryHashDatabase. The Hash format is an extensible, dynamic hashing scheme.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public class SecondaryHashDatabase : SecondaryDatabase

Visual Basic (Declaration)

Public Class SecondaryHashDatabase
    Inherits SecondaryDatabase

Visual C++

public ref class SecondaryHashDatabase : public SecondaryDatabase
Inheritance Hierarchy

`System..:::Object`
`BerkeleyDB..:::BaseDatabase`
`BerkeleyDB..:::SecondaryDatabase`
`BerkeleyDB..:::SecondaryHashDatabase`
See Also

SecondaryHashDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The **SecondaryHashDatabase** type exposes the following members.
# Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Close</td>
<td>Overloaded. Although closing a database will close any open cursors, it is recommended that applications explicitly close all their Cursor objects before closing the database. The reason why is that when the cursor is explicitly closed, the memory allocated for it is reclaimed; however, this will not happen if you close a database while cursors are still opened. The same rule, for the same reasons, hold true for Transaction objects. Simply make sure you resolve all your transaction objects before closing your database handle. Because key/data pairs are cached in memory, applications should make a point to always either close database handles or sync their data to disk (using <code>Sync()</code>) before exiting, to ensure that any data cached in main memory are reflected in the underlying file system. When called on a database that is the primary database for a secondary index, the primary database should be closed only after all secondary indices referencing it have been closed. When multiple threads are using the object concurrently, only a single thread may call the Close method. The object may not be accessed again after Close is called, regardless of its outcome.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>--------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Dispose</strong></td>
<td>Release the resources held by this object, and close the database if it's still open. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified Object is equal to the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>Exists</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>Get</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>GetBoth</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>Open</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>PrintFastStats</strong></td>
<td>The statistical information is described by the BTreeStats, HashStats, QueueStats, and RecnoStats classes.</td>
</tr>
<tr>
<td><strong>PrintStats</strong></td>
<td>The statistical information is described by the BTreeStats, HashStats, QueueStats, and RecnoStats classes.</td>
</tr>
<tr>
<td><strong>SecondaryCursor</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>Sync</strong></td>
<td>Flush any cached information to disk. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a String that represents the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>Truncate</strong></td>
<td>When called on a database configured with secondary indices, Truncate will truncate the primary database and all secondary indices. A count of the records discarded from the primary database is returned.</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>AutoCommit</strong></td>
<td>If true, all database modification operations based on this object will be transactionally protected. (Inherited from <a href="#">BaseDatabase</a>)</td>
</tr>
<tr>
<td><strong>CacheSize</strong></td>
<td>The size of the shared memory buffer pool -- that is, the cache. (Inherited from <a href="#">BaseDatabase</a>)</td>
</tr>
<tr>
<td><strong>Compare</strong></td>
<td>The secondary Hash key comparison function. The comparison function is called whenever it is necessary to compare a key specified by the application with a key currently stored in the tree. The CreatePolicy with which this database was opened. (Inherited from <a href="#">BaseDatabase</a>)</td>
</tr>
<tr>
<td><strong>Creation</strong></td>
<td></td>
</tr>
<tr>
<td><strong>DatabaseName</strong></td>
<td>The name of this database, if it has one. (Inherited from <a href="#">BaseDatabase</a>)</td>
</tr>
<tr>
<td><strong>DoChecksum</strong></td>
<td>If true, do checksum verification of pages read into the cache from the backing filestore. (Inherited from <a href="#">BaseDatabase</a>)</td>
</tr>
<tr>
<td><strong>DupCompare</strong></td>
<td>The duplicate data item comparison function.</td>
</tr>
<tr>
<td><strong>Duplicates</strong></td>
<td>Whether the insertion of duplicate data items in the database is permitted, and whether duplicates items are sorted.</td>
</tr>
<tr>
<td><strong>EncryptAlgorithm</strong></td>
<td>The algorithm used by the Berkeley DB library to perform encryption and decryption. (Inherited from <a href="#">BaseDatabase</a>)</td>
</tr>
<tr>
<td><strong>Encrypted</strong></td>
<td>If true, encrypt all data stored in the database. (Inherited from <a href="#">BaseDatabase</a>)</td>
</tr>
<tr>
<td><strong>Endianness</strong></td>
<td>The database byte order. (Inherited from <a href="#">BaseDatabase</a>)</td>
</tr>
<tr>
<td><strong>ErrorFeedback</strong></td>
<td>The mechanism for reporting detailed error messages to the application.</td>
</tr>
</tbody>
</table>
ErrorPrefix

The prefix string that appears before error messages issued by Berkeley DB.
(Inherited from BaseDatabase.)

Feedback

Monitor progress within long running operations.
(Inherited from BaseDatabase.)

FileName

The filename of this database, if it has one.
(Inherited from BaseDatabase.)

FillFactor

The desired density within the hash table.
If true, the object is free-threaded; that is, concurrently usable by multiple threads in the address space.
(Inherited from BaseDatabase.)

FreeThreaded

A user-defined hash function; if no hash function is specified, a default hash function is used.
If true, the object references a physical file supporting multiple databases.
(Inherited from BaseDatabase.)

HashFunction

If true, the underlying database files were created on an architecture of the same byte order as the current one. This information may be used to determine whether application data needs to be adjusted for this architecture or not.
(Inherited from BaseDatabase.)

HasMultiple

InHostOrder

The delegate that creates the set of secondary keys corresponding to a given primary key and data pair.
(Inherited from SecondaryDatabase.)

KeyGen

If true, this database is not mapped into process memory.

NoMMap

See MMapSize for further information.

(Inherited from BaseDatabase.)

NonDurableTxns

If true, Berkeley DB will not write log records for this database.
(Inherited from BaseDatabase.)
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Nullifier</strong></td>
<td>(Inherited from <a href="#">SecondaryDatabase</a>). The database's current page size.</td>
</tr>
<tr>
<td><strong>Pagesize</strong></td>
<td>(Inherited from <a href="#">BaseDatabase</a>). The cache priority for pages referenced by this object.</td>
</tr>
<tr>
<td><strong>Priority</strong></td>
<td>(Inherited from <a href="#">BaseDatabase</a>). If true, this database has been opened for reading only. Any attempt to modify items in the database will fail, regardless of the actual permissions of any underlying files.</td>
</tr>
<tr>
<td><strong>ReadOnly</strong></td>
<td>(Inherited from <a href="#">BaseDatabase</a>). If true, this database supports transactional read operations with degree 1 isolation. Read operations on the database may request the return of modified but not yet committed data.</td>
</tr>
<tr>
<td><strong>ReadUncommitted</strong></td>
<td>An estimate of the final size of the hash table. If true, this database has been opened in a transactional mode.</td>
</tr>
<tr>
<td><strong>Truncate</strong></td>
<td>The type of the underlying access method (and file format). This value may be used to determine the type of the database after an <a href="#">Open(String, DatabaseConfig)</a>.</td>
</tr>
<tr>
<td><strong>Type</strong></td>
<td>(Inherited from <a href="#">BaseDatabase</a>). If true, the database was opened with support for multiversion concurrency control.</td>
</tr>
</tbody>
</table>
See Also

SecondaryHashDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `SecondaryHashDatabase` type exposes the following members.
Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| Close | Overloaded.  

Although closing a database will close any open cursors, it is recommended that applications explicitly close all their Cursor objects before closing the database. The reason why is that when the cursor is explicitly closed, the memory allocated for it is reclaimed; however, this will not happen if you close a database while cursors are still opened.

The same rule, for the same reasons, hold true for Transaction objects. Simply make sure you resolve all your transaction objects before closing your database handle.

Because key/data pairs are cached in memory, applications should make a point to always either close database handles or sync their data to disk (using Sync() before exiting, to ensure that any data cached in main memory are reflected in the underlying file system.

When called on a database that is the primary database for a secondary index, the primary database should be closed only after all secondary indices referencing it have been closed.

When multiple threads are using the object concurrently, only a single thread may call the Close method.

The object may not be accessed again after Close is called, regardless of its outcome.
™ **Cursor** Overloaded.

™ **Delete** Overloaded.

™ **Dispose**
Release the resources held by this object, and close the database if it's still open.
(Inherited from **BaseDatabase**.)

™ **Equals**
Determines whether the specified **Object** is equal to the current **Object**.
(Inherited from **Object**.)

™ **Exists** Overloaded.

™ **Get** Overloaded.

™ **GetBoth** Overloaded.

™ **GetHashCode**
Serves as a hash function for a particular type.
(Inherited from **Object**.)

™ **GetType**
Gets the **Type** of the current instance.
(Inherited from **Object**.)

™ **Open** Overloaded.

™ **Open** Overloaded.

™ **PrintFastStats**
The statistical information is described by the **BTreeStats**, **HashStats**, **QueueStats**, and **RecnoStats** classes.

™ **PrintFastStats** Overloaded.

™ **PrintStats**
The statistical information is described by the **BTreeStats**, **HashStats**, **QueueStats**, and **RecnoStats** classes.

™ **SecondaryCursor** Overloaded.

™ **Sync**
Flush any cached information to disk.
(Inherited from **BaseDatabase**.)

™ **ToString**
Returns a **String** that represents the current **Object**.
(Inherited from **Object**.)

™ **ToString** Overloaded.

™ **Truncate**
When called on a database configured with secondary indices, Truncate will truncate the primary database and all secondary indices. A count of the records discarded from the primary database is returned.
See Also

SecondaryHashDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Although closing a database will close any open cursors, it is recommended that applications explicitly close all their Cursor objects before closing the database. The reason why is that when the cursor is explicitly closed, the memory allocated for it is reclaimed; however, this will not happen if you close a database while cursors are still opened.

The same rule, for the same reasons, hold true for Transaction objects. Simply make sure you resolve all your transaction objects before closing your database handle.

Because key/data pairs are cached in memory, applications should make a point to always either close database handles or sync their data to disk (using Sync() before exiting, to ensure that any data cached in main memory are reflected in the underlying file system.

When called on a database that is the primary database for a secondary index, the primary database should be closed only after all secondary indices referencing it have been closed.

When multiple threads are using the object concurrently, only a single thread may call the Close method.

The object may not be accessed again after Close is called, regardless of its outcome.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Close()()</td>
<td>Flush any cached database information to disk, close any open <code>Cursor()()</code> objects, free any allocated resources, and close any underlying files. (Inherited from <code>BaseDatabase</code>.)</td>
</tr>
<tr>
<td>Close(Boolean)</td>
<td>Optionally flush any cached database information to disk, close any open <code>Cursor()()</code> objects, free any allocated resources, and close any underlying files. (Inherited from <code>BaseDatabase</code>.)</td>
</tr>
</tbody>
</table>
See Also

SecondaryHashDatabase Class
SecondaryHashDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryHashDatabase Cursor Method

SecondaryHashDatabase Class   See Also
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| Cursor()() | Create a database cursor.  
(Inherited from `BaseDatabase`.) |
| Cursor(CursorConfig) | Create a database cursor with the given configuration.  
(Inherited from `BaseDatabase`.) |
| Cursor(Transaction) | Create a transactionally protected database cursor.  
(Inherited from `BaseDatabase`.) |
| Cursor(CursorConfig, Transaction) | Create a transactionally protected database cursor with the given configuration.  
(Inherited from `BaseDatabase`.) |
See Also

SecondaryHashDatabase Class
SecondaryHashDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryHashDatabase Class

See Also
# Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delete(DatabaseEntry)</td>
<td>Remove key/data pairs from the database. The key/data pair associated with key is discarded from the database. In the presence of duplicate key values, all records associated with the designated key will be discarded. <em>(Inherited from BaseDatabase.)</em></td>
</tr>
<tr>
<td>Delete(DatabaseEntry, Transaction)</td>
<td>Remove key/data pairs from the database. The key/data pair associated with key is discarded from the database. In the presence of duplicate key values, all records associated with the designated key will be discarded. <em>(Inherited from BaseDatabase.)</em></td>
</tr>
</tbody>
</table>
See Also

SecondaryHashDatabase Class
SecondaryHashDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryHashDatabase::Exists Method

SecondaryHashDatabase Class  See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exists(DatabaseEntry)</td>
<td>Check whether key appears in the database. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>Exists(DatabaseEntry, Transaction)</td>
<td>Check whether key appears in the database. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>Exists(DatabaseEntry, Transaction, LockingInfo)</td>
<td>Check whether key appears in the database. (Inherited from BaseDatabase.)</td>
</tr>
</tbody>
</table>
See Also

SecondaryHashDatabase Class
SecondaryHashDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryHashDatabase Class

See Also
### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Get(DatabaseEntry)</strong></td>
<td>Retrieve a key/data pair from the database. In the presence of duplicate key values, Get will return the first data item for key. (Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
<tr>
<td><strong>Get(DatabaseEntry, Transaction)</strong></td>
<td>Retrieve a key/data pair from the database. In the presence of duplicate key values, Get will return the first data item for key. (Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
<tr>
<td><strong>Get(DatabaseEntry, Transaction, LockingInfo)</strong></td>
<td>Retrieve a key/data pair from the database. In the presence of duplicate key values, Get will return the first data item for key. (Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
</tbody>
</table>
See Also

SecondaryHashDatabase Class
SecondaryHashDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryHashDatabase Class

See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetBoth(DatabaseEntry, DatabaseEntry)</td>
<td>Retrieve a key/data pair from the database which matches key and data. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>GetBoth(DatabaseEntry, DatabaseEntry, Transaction)</td>
<td>Retrieve a key/data pair from the database which matches key and data. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>GetBoth(DatabaseEntry, DatabaseEntry, Transaction, LockingInfo)</td>
<td>Retrieve a key/data pair from the database which matches key and data. (Inherited from BaseDatabase.)</td>
</tr>
</tbody>
</table>
See Also

SecondaryHashDatabase Class
SecondaryHashDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryHashDatabase Class  See Also
# Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Open(String, SecondaryHashDatabaseConfig)</code></td>
<td>Instantiate a new SecondaryHashDatabase object, open the database represented by Filename and associate the database with the primary index.</td>
</tr>
<tr>
<td><code>Open(String, SecondaryHashDatabaseConfig, Transaction)</code></td>
<td>Instantiate a new SecondaryHashDatabase object, open the database represented by Filename and associate the database with the primary index.</td>
</tr>
<tr>
<td><code>Open(String, String, SecondaryHashDatabaseConfig)</code></td>
<td>Instantiate a new SecondaryHashDatabase object, open the database represented by Filename and associate the database with the primary index.</td>
</tr>
<tr>
<td><code>Open(String, String, SecondaryHashDatabaseConfig, Transaction)</code></td>
<td>Instantiate a new SecondaryHashDatabase object, open the database represented by Filename and associate the database with the primary index.</td>
</tr>
</tbody>
</table>
See Also

SecondaryHashDatabase Class
SecondaryHashDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryHashDatabase Class  See Also

Instantiate a new SecondaryHashDatabase object, open the database represented by Filename and associate the database with the primary index.

Namespace:  BerkeleyDB  
Syntax

C#

public static SecondaryHashDatabase Open(
    string Filename,
    SecondaryHashDatabaseConfig cfg
)

Visual Basic (Declaration)

Public Shared Function Open ( _
    Filename As String, _
    cfg As SecondaryHashDatabaseConfig _
) As SecondaryHashDatabase

Visual C++

public:
static SecondaryHashDatabase^ Open(
    String^ Filename,
    SecondaryHashDatabaseConfig^ cfg
)

Parameters

Filename
Type: System::String
The name of an underlying file that will be used to back the database. In-memory databases never intended to be preserved on disk may be created by setting this parameter to null.

cfg
Type: BerkeleyDB::SecondaryHashDatabaseConfig
The database's configuration

Return Value
A new, open database object
Remarks

If Filename is null, the database is strictly temporary and cannot be opened by any other thread of control, thus the database can only be accessed by sharing the single database object that created it, in circumstances where doing so is safe.

If AutoCommit is set, the operation will be implicitly transaction protected. Note that transactionally protected operations on a database object requires the object itself be transactionally protected during its open.
See Also

SecondaryHashDatabase Class
Open Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Instantiate a new SecondaryHashDatabase object, open the database represented by Filename and associate the database with the primary index.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public static SecondaryHashDatabase Open(
    string Filename,
    SecondaryHashDatabaseConfig cfg,
    Transaction txn
)

Visual Basic (Declaration)

Public Shared Function Open ( _
    Filename As String, _
    cfg As SecondaryHashDatabaseConfig, _
    txn As Transaction _
) As SecondaryHashDatabase

Visual C++

public:
static SecondaryHashDatabase Open(
    String^ Filename,
    SecondaryHashDatabaseConfig^ cfg,
    Transaction^ txn
)

Parameters

Filename
Type: System::String
The name of an underlying file that will be used to back the database. In-memory databases never intended to be preserved on disk may be created by setting this parameter to null.

cfg
Type: BerkeleyDB::SecondaryHashDatabaseConfig
The database’s configuration
txn

Type: BerkeleyDB::::Transaction
If the operation is part of an application-specified transaction, txn is a Transaction object returned from BeginTransaction(); if the operation is part of a Berkeley DB Concurrent Data Store group, txn is a handle returned from BeginCDSGroup(); otherwise null.

Return Value

A new, open database object
Remarks

If Filename is null, the database is strictly temporary and cannot be opened by any other thread of control, thus the database can only be accessed by sharing the single database object that created it, in circumstances where doing so is safe.

If txn is null, but AutoCommit is set, the operation will be implicitly transaction protected. Note that transactionally protected operations on a database object requires the object itself be transactionally protected during its open. Also note that the transaction must be committed before the object is closed.
See Also

SecondaryHashDatabase Class
Open Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryHashDatabase Class  See Also

Instantiate a new SecondaryHashDatabase object, open the database represented by Filename and associate the database with the primary index.

Namespace:  BerkeleyDB
Syntax

C#

public static SecondaryHashDatabase Open(
    string Filename,
    string DatabaseName,
    SecondaryHashDatabaseConfig cfg
)

Visual Basic (Declaration)

Public Shared Function Open ( _
    Filename As String, _
    DatabaseName As String, _
    cfg As SecondaryHashDatabaseConfig _
) As SecondaryHashDatabase

Visual C++

public:
static SecondaryHashDatabase^ Open(
    String^ Filename,
    String^ DatabaseName,
    SecondaryHashDatabaseConfig^ cfg
)

Parameters

Filename
Type: System::String
The name of an underlying file that will be used to back the database. In-memory databases never intended to be preserved on disk may be created by setting this parameter to null.

DatabaseName
Type: System::String
This parameter allows applications to have multiple databases in a single file. Although no DatabaseName needs to be specified, it is an error to
attempt to open a second database in a file that was not initially created using a database name.

cfg
    Type: BerkeleyDB::SecondaryHashDatabaseConfig
The database's configuration

**Return Value**

A new, open database object
Remarks

If both Filename and DatabaseName are null, the database is strictly temporary and cannot be opened by any other thread of control, thus the database can only be accessed by sharing the single database object that created it, in circumstances where doing so is safe. If Filename is null and DatabaseName is non-null, the database can be opened by other threads of control and will be replicated to client sites in any replication group.

If AutoCommit is set, the operation will be implicitly transaction protected. Note that transactionally protected operations on a database object requires the object itself be transactionally protected during its open.
See Also

SecondaryHashDatabase Class
Open Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryHashDatabase Class  See Also

Instantiate a new SecondaryHashDatabase object, open the database represented by Filename and associate the database with the primary index.

Namespace: BerkeleyDB
Syntax

C#

```csharp
public static SecondaryHashDatabase Open(
    string Filename,
    string DatabaseName,
    SecondaryHashDatabaseConfig cfg,
    Transaction txn
)
```

Visual Basic (Declaration)

```vbnet
Public Shared Function Open ( _
    Filename As String, _
    DatabaseName As String, _
    cfg As SecondaryHashDatabaseConfig, _
    txn As Transaction _
) As SecondaryHashDatabase
```

Visual C++

```cpp
public:
static SecondaryHashDatabase^ Open(
    String^ Filename, 
    String^ DatabaseName, 
    SecondaryHashDatabaseConfig^ cfg, 
    Transaction^ txn
)
```

Parameters

Filename
Type: `System::::String`
The name of an underlying file that will be used to back the database. In-memory databases never intended to be preserved on disk may be created by setting this parameter to null.

DatabaseName
Type: `System::::String`
This parameter allows applications to have multiple databases in a single file. Although no DatabaseName needs to be specified, it is an error to attempt to open a second database in a file that was not initially created using a database name.

**cfg**
Type: BerkeleyDB::SecondaryHashDatabaseConfig
The database's configuration

**txn**
Type: BerkeleyDB::Transaction
If the operation is part of an application-specified transaction, txn is a Transaction object returned from `BeginTransaction()`; if the operation is part of a Berkeley DB Concurrent Data Store group, txn is a handle returned from `BeginCDSGroup()`; otherwise null.

**Return Value**
A new, open database object
Remarks

If both Filename and DatabaseName are null, the database is strictly temporary and cannot be opened by any other thread of control, thus the database can only be accessed by sharing the single database object that created it, in circumstances where doing so is safe. If Filename is null and DatabaseName is non-null, the database can be opened by other threads of control and will be replicated to client sites in any replication group.

If txn is null, but AutoCommit is set, the operation will be implicitly transaction protected. Note that transactionally protected operations on a database object requires the object itself be transactionally protected during its open. Also note that the transaction must be committed before the object is closed.
See Also

SecondaryHashDatabase Class
Open Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryHashDatabase Class  See Also

The statistical information is described by the BTreeStats, HashStats, QueueStats, and RecnoStats classes.
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PrintFastStats()()()</td>
<td>Display the database statistical information which does not require traversal of the database. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>PrintFastStats(Boolean)</td>
<td>Display the database statistical information which does not require traversal of the database. (Inherited from BaseDatabase.)</td>
</tr>
</tbody>
</table>
See Also

SecondaryHashDatabase Class  
SecondaryHashDatabase Members  
BerkeleyDB Namespace  

Report Feedback on this item in the Oracle Technology Network Forum  

Copyright (c) 1996-2009 Oracle. All rights reserved.
The statistical information is described by the `BTreeStats`, `HashStats`, `QueueStats`, and `RecnoStats` classes.
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>PrintStats()</code></td>
<td>Display the database statistical information.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
<tr>
<td><code>PrintStats(Boolean)</code></td>
<td>Display the database statistical information.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
</tbody>
</table>
See Also

SecondaryHashDatabase Class
SecondaryHashDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryHashDatabase Class
# Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>SecondaryCursor()</code>()</td>
<td>Create a secondary database cursor. (Inherited from <a href="#">SecondaryDatabase</a>.)</td>
</tr>
<tr>
<td><code>SecondaryCursor(CursorConfig)</code></td>
<td>Create a secondary database cursor with the given configuration. (Inherited from <a href="#">SecondaryDatabase</a>.)</td>
</tr>
<tr>
<td><code>SecondaryCursor(Transaction)</code></td>
<td>Create a transactionally protected secondary database cursor. (Inherited from <a href="#">SecondaryDatabase</a>.)</td>
</tr>
<tr>
<td><code>SecondaryCursor(CursorConfig, Transaction)</code></td>
<td>Create a transactionally protected secondary database cursor with the given configuration. (Inherited from <a href="#">SecondaryDatabase</a>.)</td>
</tr>
</tbody>
</table>
See Also

SecondaryHashDatabase Class
SecondaryHashDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
When called on a database configured with secondary indices, Truncate will truncate the primary database and all secondary indices. A count of the records discarded from the primary database is returned.
### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Truncate();</strong></td>
<td>Empty the database, discarding all records it contains. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>Truncate(Transaction)</strong></td>
<td>Empty the database, discarding all records it contains. (Inherited from BaseDatabase.)</td>
</tr>
</tbody>
</table>
See Also

SecondaryHashDatabase Class
SecondaryHashDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The **SecondaryHashDatabase** type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AutoCommit</strong></td>
<td>If true, all database modification operations based on this object will be transactionally protected. (Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
<tr>
<td><strong>CacheSize</strong></td>
<td>The size of the shared memory buffer pool -- that is, the cache. (Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
<tr>
<td><strong>Compare</strong></td>
<td>The secondary Hash key comparison function. The comparison function is called whenever it is necessary to compare a key specified by the application with a key currently stored in the tree. The CreatePolicy with which this database was opened. (Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
<tr>
<td><strong>Creation</strong></td>
<td>The CreatePolicy with which this database was opened. (Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
<tr>
<td><strong>DatabaseName</strong></td>
<td>The name of this database, if it has one. (Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
<tr>
<td><strong>DoChecksum</strong></td>
<td>If true, do checksum verification of pages read into the cache from the backing filestore. (Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
<tr>
<td><strong>DupCompare</strong></td>
<td>The duplicate data item comparison function.</td>
</tr>
<tr>
<td><strong>Duplicates</strong></td>
<td>Whether the insertion of duplicate data items in the database is permitted, and whether duplicates items are sorted.</td>
</tr>
<tr>
<td><strong>EncryptAlgorithm</strong></td>
<td>The algorithm used by the Berkeley DB library to perform encryption and decryption. (Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
<tr>
<td><strong>Encrypted</strong></td>
<td>If true, encrypt all data stored in the database. (Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
<tr>
<td><strong>Endianness</strong></td>
<td>The database byte order. (Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
<tr>
<td><strong>ErrorFeedback</strong></td>
<td>The mechanism for reporting detailed error messages to the application.</td>
</tr>
</tbody>
</table>
ErrorPrefix
The prefix string that appears before error messages issued by Berkeley DB.
(Inherited from BaseDatabase.)

Feedback
Monitor progress within long running operations.
(Inherited from BaseDatabase.)

FileName
The filename of this database, if it has one.
(Inherited from BaseDatabase.)

FillFactor
The desired density within the hash table.
If true, the object is free-threaded; that is, concurrently usable by multiple threads in the address space.
(Inherited from BaseDatabase.)

FreeThreaded
If true, the object is free-threaded; that is, concurrently usable by multiple threads in the address space.
(Inherited from BaseDatabase.)

HashFunction
A user-defined hash function; if no hash function is specified, a default hash function is used.
If true, the object references a physical file supporting multiple databases.
(Inherited from BaseDatabase.)

HasMultiple
If true, the underlying database files were created on an architecture of the same byte order as the current one. This information may be used to determine whether application data needs to be adjusted for this architecture or not.
(Inherited from BaseDatabase.)

InHostOrder
The delegate that creates the set of secondary keys corresponding to a given primary key and data pair.
(Inherited from SecondaryDatabase.)

KeyGen
If true, this database is not mapped into process memory.

NoMMap
See MMapSize for further information.
(Inherited from BaseDatabase.)

NonDurableTxns
If true, Berkeley DB will not write log records for this database.
(Inherited from BaseDatabase.)
Nullifier  (Inherited from SecondaryDatabase.)
The database's current page size.

Pagesize  (Inherited from BaseDatabase.)
The cache priority for pages referenced by this object.

Priority  (Inherited from BaseDatabase.)
If true, this database has been opened for reading only. Any attempt to modify items in the database will fail, regardless of the actual permissions of any underlying files.

ReadOnly  (Inherited from BaseDatabase.)
If true, this database supports transactional read operations with degree 1 isolation. Read operations on the database may request the return of modified but not yet committed data.

ReadUncommitted  (Inherited from BaseDatabase.)
If true, the database was opened with support for multiversion concurrency control.

TableSize  An estimate of the final size of the hash table.

Transactional  (Inherited from BaseDatabase.)
If true, this database has been opened in a transactional mode.

Truncated  (Inherited from BaseDatabase.)
If true, the underlying file was physically truncated upon open, discarding all previous databases it might have held.

Type  (Inherited from BaseDatabase.)
The type of the underlying access method (and file format). This value may be used to determine the type of the database after an Open(String, DatabaseConfig).

UseMVCC  (Inherited from BaseDatabase.)
If true, the database was opened with support for multiversion concurrency control.
See Also

SecondaryHashDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryHashDatabase Class  See Also

The secondary Hash key comparison function. The comparison function is called whenever it is necessary to compare a key specified by the application with a key currently stored in the tree.

Namespace:  BerkeleyDB
Syntax

C#

public EntryComparisonDelegate Compare { get; private set; }

Visual Basic (Declaration)

Public Property Compare As EntryComparisonDelegate

Visual C++

public:
 property EntryComparisonDelegate^ Compare {
    EntryComparisonDelegate^ get ();
    void set (EntryComparisonDelegate^ value);
}
See Also

SecondaryHashDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryHashDatabase Class

See Also

The duplicate data item comparison function.

**Namespace:** [BerkeleyDB](#)

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public EntryComparisonDelegate DupCompare { get; private set; }

Visual Basic (Declaration)

Public Property DupCompare As EntryComparisonDelegate

Visual C++

public:

property EntryComparisonDelegate^ DupCompare {
    EntryComparisonDelegate^ get ();
    void set (EntryComparisonDelegate^ value);
}
See Also

SecondaryHashDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Whether the insertion of duplicate data items in the database is permitted, and whether duplicates items are sorted.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public DuplicatesPolicy Duplicates { get; }

Visual Basic (Declaration)

Public ReadOnly Property Duplicates As DuplicatesPolicy

Visual C++

public:
property DuplicatesPolicy Duplicates {
    DuplicatesPolicy get ();
}

See Also

SecondaryHashDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The desired density within the hash table.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
## Syntax

### C#

```csharp
public uint FillFactor { get; }
```

### Visual Basic (Declaration)

```vbnet
Public ReadOnly Property FillFactor AsUInteger
```

### Visual C++

```cpp
public:
property unsigned int FillFactor {
    unsigned int get ();
}
```
See Also

SecondaryHashDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryHashDatabase Class  See Also

A user-defined hash function; if no hash function is specified, a default hash function is used.

Namespace:  BerkeleyDB
Syntax

C#

public HashFunctionDelegate HashFunction { get; private set; }

Visual Basic (Declaration)

Public Property HashFunction As HashFunctionDelegate

Visual C++

public:
property HashFunctionDelegate^ HashFunction {
    HashFunctionDelegate^ get ();
    void set (HashFunctionDelegate^ value);
}
See Also

SecondaryHashDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
An estimate of the final size of the hash table.

**Namespace:** [BerkeleyDB](https://www.oracle.com/database/berkeley-db/)

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public uint TableSize { get; }

Visual Basic (Declaration)

Public ReadOnly Property TableSize AsUInteger

Visual C++

public:
property unsigned int TableSize {
    unsigned int get ();
}

See Also

SecondaryHashDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
A class representing configuration parameters for SecondaryHashDatabase

Namespace: BerkeleyDB
Syntax

C#

public class SecondaryHashDatabaseConfig : SecondaryDatabaseConfig

Visual Basic (Declaration)

Public Class SecondaryHashDatabaseConfig
   Inherits SecondaryDatabaseConfig

Visual C++

public ref class SecondaryHashDatabaseConfig : public SecondaryData
Inheritance Hierarchy

System..:::Object
BerkeleyDB..:::DatabaseConfig
  BerkeleyDB..:::SecondaryDatabaseConfig
    BerkeleyDB..:::SecondaryHashDatabaseConfig
See Also

SecondaryHashDatabaseConfig Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation
SecondaryHashDatabaseConfig Members

The `SecondaryHashDatabaseConfig` type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SecondaryHashDatabaseConfig</td>
<td>Instantiate a new SecondaryHashDatabaseConfig object</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="#">Object</a>)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a>)</td>
</tr>
<tr>
<td><strong>SetEncryption</strong></td>
<td>Set the password and algorithm used by the Berkeley DB library to perform encryption and decryption. (Inherited from <a href="#">DatabaseConfig</a>)</td>
</tr>
<tr>
<td><strong>SetForeignKeyConstraint</strong></td>
<td>Overloaded. Returns a <a href="#">String</a> that represents the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td></td>
</tr>
</tbody>
</table>
## Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AutoCommit</strong></td>
<td>Enclose the open call within a transaction. If the call succeeds, the open operation will be recoverable and all subsequent database modification operations based on this handle will be transactionally protected. If the call fails, no database will have been created. (Inherited from <a href="#">DatabaseConfig</a>)</td>
</tr>
<tr>
<td><strong>ByteOrder</strong></td>
<td>The byte order for integers in the stored database metadata. The host byte order of the machine where the Berkeley DB library was compiled is the default value. (Inherited from <a href="#">DatabaseConfig</a>)</td>
</tr>
<tr>
<td><strong>CacheSize</strong></td>
<td>The size of the shared memory buffer pool -- that is, the cache. (Inherited from <a href="#">DatabaseConfig</a>)</td>
</tr>
<tr>
<td><strong>Compare</strong></td>
<td>The Secondary Hash key comparison function.</td>
</tr>
<tr>
<td><strong>Creation</strong></td>
<td>The policy for how to handle database creation. If true, do checksum verification of pages read into the cache from the backing filestore. (Inherited from <a href="#">DatabaseConfig</a>)</td>
</tr>
<tr>
<td><strong>DoChecksum</strong></td>
<td>The duplicate data item comparison function.</td>
</tr>
<tr>
<td><strong>DuplicateCompare</strong></td>
<td>The policy for duplicate data items in the database; that is, insertion when the key of the key/data pair being inserted already exists in the database will be successful.</td>
</tr>
<tr>
<td><strong>Duplicates</strong></td>
<td>The Berkeley DB environment within which to create a database. If null, the database will be created stand-alone; that is, it is not part of any Berkeley DB environment. (Inherited from <a href="#">DatabaseConfig</a>)</td>
</tr>
</tbody>
</table>
| **Env**          | The mechanism for reporting error messages to the


- **ErrorFeedback** application.
  (Inherited from [DatabaseConfig](#).)

- **ErrorPrefix**
  The prefix string that appears before error messages issued by Berkeley DB.
  (Inherited from [DatabaseConfig](#).)

- **Feedback**
  (Inherited from [DatabaseConfig](#).)

- **FreeThreaded**
  Cause the database object to be free-threaded; that is, concurrently usable by multiple threads in the address space.
  (Inherited from [DatabaseConfig](#).)

- **HashFunction**
  A user-defined hash function; if no hash function is specified, a default hash function is used.

- **ImmutableKey**
  If true, the secondary key is immutable.
  (Inherited from [SecondaryDatabaseConfig](#).)

- **NoMMap**
  Do not map this database into process memory.
  (Inherited from [DatabaseConfig](#).)

- **NonDurableTxns**
  If true, Berkeley DB will not write log records for this database.
  (Inherited from [DatabaseConfig](#).)

- **Populate**
  If true and the secondary database is empty, walk through Primary and create an index to it in the empty secondary. This operation is potentially very expensive.
  (Inherited from [SecondaryDatabaseConfig](#).)

- **Priority**
  The cache priority for pages referenced by the database.
  (Inherited from [DatabaseConfig](#).)

- **ReadOnly**
  Open the database for reading only. Any attempt to modify items in the database will fail, regardless of the actual permissions of any underlying files.
  (Inherited from [DatabaseConfig](#).)

- **ReadUncommitted**
  Support transactional read operations with degree 1 isolation.
  (Inherited from [DatabaseConfig](#).)

- **Truncate**
  Physically truncate the underlying file, discarding all previous databases it might have held.
UseMVCC

(IInherited from `DatabaseConfig`.)
Open the database with support for multiversion concurrency control.
(IInherited from `DatabaseConfig`.)
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EncryptAlgorithm</td>
<td>The algorithm used to perform encryption and decryption. (Inherited from <a href="#">DatabaseConfig</a>.)</td>
</tr>
<tr>
<td>EncryptionPassword</td>
<td>The password used to perform encryption and decryption. (Inherited from <a href="#">DatabaseConfig</a>.)</td>
</tr>
<tr>
<td>FillFactor</td>
<td>The desired density within the hash table. If no value is specified, the fill factor will be selected dynamically as pages are filled.</td>
</tr>
<tr>
<td>ForeignKeyDatabase</td>
<td>(Inherited from <a href="#">SecondaryDatabaseConfig</a>.)</td>
</tr>
<tr>
<td>ForeignKeyNullifier</td>
<td>(Inherited from <a href="#">SecondaryDatabaseConfig</a>.)</td>
</tr>
<tr>
<td>KeyGen</td>
<td>The delegate that creates the set of secondary keys corresponding to a given primary key and data pair. (Inherited from <a href="#">SecondaryDatabaseConfig</a>.)</td>
</tr>
<tr>
<td>OnForeignKeyDelete</td>
<td>(Inherited from <a href="#">SecondaryDatabaseConfig</a>.)</td>
</tr>
<tr>
<td>PageSize</td>
<td>The size of the pages used to hold items in the database, in bytes. (Inherited from <a href="#">DatabaseConfig</a>.)</td>
</tr>
<tr>
<td>Primary</td>
<td>All updates to Primary will be automatically reflected in the secondary and all reads from the secondary will return corresponding data from Primary. (Inherited from <a href="#">SecondaryDatabaseConfig</a>.)</td>
</tr>
<tr>
<td>TableSize</td>
<td>An estimate of the final size of the hash table.</td>
</tr>
</tbody>
</table>
See Also

SecondaryHashDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Namespace: BerkeleyDB
Syntax

C#

public SecondaryHashDatabaseConfig(
    Database PrimaryDB,
    SecondaryKeyGenDelegate KeyGenFunc
)

Visual Basic (Declaration)

Public Sub New (_
    PrimaryDB As Database, _
    KeyGenFunc As SecondaryKeyGenDelegate _
)

Visual C++

public:
SecondaryHashDatabaseConfig(
    Database^ PrimaryDB,
    SecondaryKeyGenDelegate^ KeyGenFunc
)

Parameters

PrimaryDB
Type: BerkeleyDB::::Database

KeyGenFunc
Type: BerkeleyDB::::SecondaryKeyGenDelegate
See Also

SecondaryHashDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `SecondaryHashDatabaseConfig` type exposes the following members.
### Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AutoCommit</strong></td>
<td>Enclose the open call within a transaction. If the call succeeds, the open operation will be recoverable and all subsequent database modification operations based on this handle will be transactionally protected. If the call fails, no database will have been created. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td><strong>ByteOrder</strong></td>
<td>The byte order for integers in the stored database metadata. The host byte order of the machine where the Berkeley DB library was compiled is the default value. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td><strong>CacheSize</strong></td>
<td>The size of the shared memory buffer pool -- that is, the cache. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td><strong>Compare</strong></td>
<td>The Secondary Hash key comparison function.</td>
</tr>
<tr>
<td><strong>Creation</strong></td>
<td>The policy for how to handle database creation.</td>
</tr>
<tr>
<td><strong>DoChecksum</strong></td>
<td>If true, do checksum verification of pages read into the cache from the backing filestore. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td><strong>DuplicateCompare</strong></td>
<td>The duplicate data item comparison function.</td>
</tr>
<tr>
<td><strong>Duplicates</strong></td>
<td>Policy for duplicate data items in the database; that is, insertion when the key of the key/data pair being inserted already exists in the database will be successful.</td>
</tr>
<tr>
<td><strong>Env</strong></td>
<td>The Berkeley DB environment within which to create a database. If null, the database will be created stand-alone; that is, it is not part of any Berkeley DB environment. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td></td>
<td>The mechanism for reporting error messages to the</td>
</tr>
<tr>
<td><strong>Property</strong></td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ErrorFeedback</td>
<td>The prefix string that appears before error messages issued by Berkeley DB. <em>(Inherited from DatabaseConfig.)</em></td>
</tr>
<tr>
<td>ErrorPrefix</td>
<td>The prefix string that appears before error messages issued by Berkeley DB. <em>(Inherited from DatabaseConfig.)</em></td>
</tr>
<tr>
<td>Feedback</td>
<td>Cause the database object to be free-threaded; that is, concurrently usable by multiple threads in the address space. <em>(Inherited from DatabaseConfig.)</em></td>
</tr>
<tr>
<td>FreeThreaded</td>
<td>Cause the database object to be free-threaded; that is, concurrently usable by multiple threads in the address space. <em>(Inherited from DatabaseConfig.)</em></td>
</tr>
<tr>
<td>HashFunction</td>
<td>A user-defined hash function; if no hash function is specified, a default hash function is used.</td>
</tr>
<tr>
<td>ImmutableKey</td>
<td>If true, the secondary key is immutable. <em>(Inherited from SecondaryDatabaseConfig.)</em></td>
</tr>
<tr>
<td>NoMMap</td>
<td>Do not map this database into process memory. <em>(Inherited from DatabaseConfig.)</em></td>
</tr>
<tr>
<td>NonDurableTxns</td>
<td>If true, Berkeley DB will not write log records for this database. <em>(Inherited from DatabaseConfig.)</em></td>
</tr>
<tr>
<td>Populate</td>
<td>If true and the secondary database is empty, walk through Primary and create an index to it in the empty secondary. This operation is potentially very expensive. <em>(Inherited from SecondaryDatabaseConfig.)</em></td>
</tr>
<tr>
<td>Priority</td>
<td>The cache priority for pages referenced by the database. <em>(Inherited from DatabaseConfig.)</em></td>
</tr>
<tr>
<td>ReadOnly</td>
<td>Open the database for reading only. Any attempt to modify items in the database will fail, regardless of the actual permissions of any underlying files. <em>(Inherited from DatabaseConfig.)</em></td>
</tr>
<tr>
<td>ReadUncommitted</td>
<td>Support transactional read operations with degree 1 isolation. <em>(Inherited from DatabaseConfig.)</em></td>
</tr>
<tr>
<td>Truncate</td>
<td>Physically truncate the underlying file, discarding all previous databases it might have held.</td>
</tr>
</tbody>
</table>
UseMVCC

Open the database with support for multiversion concurrency control.

(Inherited from DatabaseConfig.)
See Also

SecondaryHashDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The Secondary Hash key comparison function.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public EntryComparisonDelegate Compare

Visual Basic (Declaration)

Public Compare As EntryComparisonDelegate

Visual C++

public:
EntryComparisonDelegate^ Compare
Remarks

The comparison function is called whenever it is necessary to compare a key specified by the application with a key currently stored in the tree.

If no comparison function is specified, the keys are compared lexically, with shorter keys collating before longer keys.

If the database already exists, the comparison function must be the same as that historically used to create the database or corruption can occur.
See Also

SecondaryHashDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryHashDatabaseConfig::Creation Field

The policy for how to handle database creation.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public CreatePolicy Creation

Visual Basic (Declaration)

Public Creation As CreatePolicy

Visual C++

public:
CreatePolicy Creation
Remarks

If the database does not already exist and NEVER is set, Open(String, SecondaryHashDatabaseConfig) will fail.
See Also

SecondaryHashDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryHashDatabaseConfig::DuplicateCompare Field

The duplicate data item comparison function.

**Namespace:** BerkeleyDB
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

**C#**

public `EntryComparisonDelegate` DuplicateCompare

**Visual Basic (Declaration)**

Public DuplicateCompare As `EntryComparisonDelegate`

**Visual C++**

public:
`EntryComparisonDelegate` DuplicateCompare
Remarks

The comparison function is called whenever it is necessary to compare a data item specified by the application with a data item currently stored in the database. Setting DuplicateCompare implies setting Duplicates to SORTED.

If no comparison function is specified, the data items are compared lexically, with shorter data items collating before longer data items.

If the database already exists when Open(String, SecondaryHashDatabaseConfig) is called, the delegate must be the same as that historically used to create the database or corruption can occur.
See Also

SecondaryHashDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Policy for duplicate data items in the database; that is, insertion when the key of the key/data pair being inserted already exists in the database will be successful.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
### Syntax

**C#**

```csharp
public DuplicatesPolicy Duplicates
```

**Visual Basic (Declaration)**

```vbnet
Public Duplicates As DuplicatesPolicy
```

**Visual C++**

```c++
public:
DuplicatesPolicy Duplicates
```
Remarks

The ordering of duplicates in the database for **UNSORTED** is determined by the order of insertion, unless the ordering is otherwise specified by use of a cursor operation or a duplicate sort function. The ordering of duplicates in the database for **SORTED** is determined by the duplicate comparison function. If the application does not specify a comparison function using `DuplicateCompare`, a default lexical comparison will be used.

**SORTED** is preferred to **UNSORTED** for performance reasons. **UNSORTED** should only be used by applications wanting to order duplicate data items manually.

If the database already exists, the value of Duplicates must be the same as the existing database or an error will be returned.
See Also

SecondaryHashDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryHashDatabaseConfig::HashFunction Field

A user-defined hash function; if no hash function is specified, a default hash function is used.

Namespace: BerkeleyDB
Syntax

C#
public HashFunctionDelegate HashFunction

Visual Basic (Declaration)
Public HashFunction As HashFunctionDelegate

Visual C++
public:
HashFunctionDelegate^ HashFunction
Remarks

Because no hash function performs equally well on all possible data, the user may find that the built-in hash function performs poorly with a particular data set.

If the database already exists, HashFunction must be the same as that historically used to create the database or corruption can occur.
See Also

SecondaryHashDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
C#  Visual Basic  Visual C++  Include Protected Members  Include Inherited Members

Berkeley DB .NET API Documentation

SecondaryHashDatabaseConfig Methods

SecondaryHashDatabaseConfig Class  See Also

The SecondaryHashDatabaseConfig type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a>.</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type.</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <a href="#">Type</a> of the current instance.</td>
</tr>
<tr>
<td><strong>SetEncryption</strong></td>
<td>Set the password and algorithm used by the Berkeley DB library to perform encryption and decryption.</td>
</tr>
<tr>
<td><strong>SetForeignKeyConstraint</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a <a href="#">String</a> that represents the current <a href="#">Object</a>.</td>
</tr>
</tbody>
</table>
See Also

SecondaryHashDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryHashDatabaseConfig::SetForeignKeyConstraint Method

See Also

SecondaryHashDatabaseConfig Class
# Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SetForeignKeyConstraint(Database, ForeignKeyDeleteAction)</td>
<td>(Inherited from SecondaryDatabaseConfig.)</td>
</tr>
<tr>
<td>SetForeignKeyConstraint(Database, ForeignKeyDeleteAction, ForeignKeyNullifyDelegate)</td>
<td>(Inherited from SecondaryDatabaseConfig.)</td>
</tr>
</tbody>
</table>
See Also

SecondaryHashDatabaseConfig Class
SecondaryHashDatabaseConfig Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `SecondaryHashDatabaseConfig` type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EncryptAlgorithm</td>
<td>The algorithm used to perform encryption and decryption.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">DatabaseConfig</a>)</td>
</tr>
<tr>
<td>EncryptionPassword</td>
<td>The password used to perform encryption and decryption.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">DatabaseConfig</a>)</td>
</tr>
<tr>
<td>FillFactor</td>
<td>The desired density within the hash table. If no value is specified, the fill factor will be selected dynamically as pages are filled.</td>
</tr>
<tr>
<td>ForeignKeyDatabase</td>
<td>(Inherited from <a href="#">SecondaryDatabaseConfig</a>)</td>
</tr>
<tr>
<td>ForeignKeyNullifier</td>
<td>(Inherited from <a href="#">SecondaryDatabaseConfig</a>)</td>
</tr>
<tr>
<td>KeyGen</td>
<td>The delegate that creates the set of secondary keys corresponding to a given primary key and data pair.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">SecondaryDatabaseConfig</a>)</td>
</tr>
<tr>
<td>OnForeignKeyDelete</td>
<td>(Inherited from <a href="#">SecondaryDatabaseConfig</a>)</td>
</tr>
<tr>
<td>PageSize</td>
<td>The size of the pages used to hold items in the database, in bytes.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">DatabaseConfig</a>)</td>
</tr>
<tr>
<td>Primary</td>
<td>All updates to Primary will be automatically reflected in the secondary and all reads from the secondary will return corresponding data from Primary.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">SecondaryDatabaseConfig</a>)</td>
</tr>
<tr>
<td>TableSize</td>
<td>An estimate of the final size of the hash table.</td>
</tr>
</tbody>
</table>
See Also

*SecondaryHashDatabaseConfig Class*
*BerkeleyDB Namespace*

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The desired density within the hash table. If no value is specified, the fill factor will be selected dynamically as pages are filled.

Namespace: BerkeleyDB
Syntax

C#

public uint FillFactor { get; set; }

Visual Basic (Declaration)

Public Property FillFactor As UInteger

Visual C++

public:
property unsigned int FillFactor {
    unsigned int get ();
    void set (unsigned int value);
}
Remarks

The density is an approximation of the number of keys allowed to accumulate in any one bucket, determining when the hash table grows or shrinks. If you know the average sizes of the keys and data in your data set, setting the fill factor can enhance performance. A reasonable rule computing fill factor is to set it to the following:

\[
\frac{\text{pagesize} - 32}{\text{average_key_size} + \text{average_data_size} + 8}
\]

If the database already exists, this setting will be ignored.
See Also

SecondaryHashDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
An estimate of the final size of the hash table.

Namespace: BerkeleyDB
Syntax

C#

public uint TableSize { get; set; }

Visual Basic (Declaration)

Public Property TableSize AsUInteger

Visual C++

public:
property unsigned int TableSize {
  unsigned int get ();
  void set (unsigned int value);
}
Remarks

In order for the estimate to be used when creating the database, FillFactor must also be set. If the estimate or fill factor are not set or are set too low, hash tables will still expand gracefully as keys are entered, although a slight performance degradation may be noticed.

If the database already exists, this setting will be ignored.
See Also

SecondaryHashDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The function that creates the set of secondary keys corresponding to a given primary key and data pair.

**Namespace:**  BerkeleyDB  
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
## Syntax

### C#

```csharp
public delegate DatabaseEntry SecondaryKeyGenDelegate(
    DatabaseEntry key,
    DatabaseEntry data
)
```

### Visual Basic (Declaration)

```vbnet
Public Delegate Function SecondaryKeyGenDelegate ( _
    key As DatabaseEntry, _
    data As DatabaseEntry _
) As DatabaseEntry
```

### Visual C++

```cpp
public delegate DatabaseEntry^ SecondaryKeyGenDelegate(
    DatabaseEntry^ key, 
    DatabaseEntry^ data
)
```

## Parameters

- **key**
  - Type: `BerkeleyDB::DatabaseEntry`
  - The primary key

- **data**
  - Type: `BerkeleyDB::DatabaseEntry`
  - The primary data item

## Return Value

The secondary key(s)
See Also

BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
A class representing a SecondaryQueueDatabase. The Queue format supports fast access to fixed-length records accessed sequentially or by logical record number.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

**C#**

```csharp
public class SecondaryQueueDatabase : SecondaryDatabase
```

**Visual Basic (Declaration)**

```vbnet
Public Class SecondaryQueueDatabase
    Inherits SecondaryDatabase
```

**Visual C++**

```cpp
public ref class SecondaryQueueDatabase : public SecondaryDatabase
```
Inheritance Hierarchy

System...:::Object
BerkeleyDB...:::BaseDatabase
BerkeleyDB...:::SecondaryDatabase
BerkeleyDB...:::SecondaryQueueDatabase
See Also

SecondaryQueueDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The **SecondaryQueueDatabase** type exposes the following members.
Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Close</td>
<td>Overloaded. Although closing a database will close any open cursors, it is recommended that applications explicitly close all their Cursor objects before closing the database. The reason why is that when the cursor is explicitly closed, the memory allocated for it is reclaimed; however, this will not happen if you close a database while cursors are still opened. The same rule, for the same reasons, hold true for Transaction objects. Simply make sure you resolve all your transaction objects before closing your database handle. Because key/data pairs are cached in memory, applications should make a point to always either close database handles or sync their data to disk (using Sync()()() before exiting, to ensure that any data cached in main memory are reflected in the underlying file system. When called on a database that is the primary database for a secondary index, the primary database should be closed only after all secondary indices referencing it have been closed. When multiple threads are using the object concurrently, only a single thread may call the Close method. The object may not be accessed again after Close is called, regardless of its outcome.</td>
</tr>
</tbody>
</table>
Cursor
Overloaded.
Delete
Overloaded.
Dispose
Release the resources held by this object, and close the database if it's still open.
(Inherited from BaseDatabase.)
Equals
Determines whether the specified Object is equal to the current Object.
(Inherited from Object.)
Exists
Overloaded.
Get
Overloaded.
GetBoth
Overloaded.
GetHashCode
Serves as a hash function for a particular type.
(Inherited from Object.)
GetType
Gets the Type of the current instance.
(Inherited from Object.)
Open
Overloaded.
PrintFastStats
The statistical information is described by the BTreeStats, HashStats, QueueStats, and RecnoStats classes.
(PrintedStats
Overloaded.
The statistical information is described by the BTreeStats, HashStats, QueueStats, and RecnoStats classes.
SecondaryCursor
Overloaded.
Sync
Flush any cached information to disk.
(Inherited from BaseDatabase.)
ToString
Returns a String that represents the current Object.
(Inherited from Object.)
Truncate
When called on a database configured with secondary indices, Truncate will truncate the primary database and all secondary indices. A count of the records discarded from the primary database is returned.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AutoCommit</td>
<td>If true, all database modification operations based on this object will be transactionally protected. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>CacheSize</td>
<td>The size of the shared memory buffer pool -- that is, the cache. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>Creation</td>
<td>The CreatePolicy with which this database was opened. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>DatabaseName</td>
<td>The name of this database, if it has one. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>DoChecksum</td>
<td>If true, do checksum verification of pages read into the cache from the backing filestore. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>EncryptAlgorithm</td>
<td>The algorithm used by the Berkeley DB library to perform encryption and decryption. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>Encrypted</td>
<td>If true, encrypt all data stored in the database. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>Endianness</td>
<td>The database byte order. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>ErrorFeedback</td>
<td>The mechanism for reporting detailed error messages to the application. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>ErrorPrefix</td>
<td>The prefix string that appears before error messages issued by Berkeley DB. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>ExtentSize</td>
<td>The size of the extents used to hold pages in a QueueDatabase, specified as a number of pages.</td>
</tr>
<tr>
<td>Feedback</td>
<td>Monitor progress within long running operations. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>FileName</strong></td>
<td>The filename of this database, if it has one. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>FreeThreaded</strong></td>
<td>If true, the object is free-threaded; that is, concurrently usable by multiple threads in the address space. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>HasMultiple</strong></td>
<td>If true, the object references a physical file supporting multiple databases. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>InHostOrder</strong></td>
<td>If true, the underlying database files were created on an architecture of the same byte order as the current one. This information may be used to determine whether application data needs to be adjusted for this architecture or not. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>KeyGen</strong></td>
<td>The delegate that creates the set of secondary keys corresponding to a given primary key and data pair. (Inherited from SecondaryDatabase.)</td>
</tr>
<tr>
<td><strong>Length</strong></td>
<td>The length of records in the database.</td>
</tr>
<tr>
<td><strong>NoMMap</strong></td>
<td>If true, this database is not mapped into process memory.</td>
</tr>
<tr>
<td><strong>NonDurableTxns</strong></td>
<td>See MMapSize for further information. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>Nullifier</strong></td>
<td>If true, Berkeley DB will not write log records for this database. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>PadByte</strong></td>
<td>The padding character for short, fixed-length records.</td>
</tr>
<tr>
<td><strong>Pagesize</strong></td>
<td>The database's current page size. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>Priority</strong></td>
<td>The cache priority for pages referenced by this object. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>Attribute</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>ReadOnly</strong></td>
<td>If true, this database has been opened for reading only. Any attempt to modify items in the database will fail, regardless of the actual permissions of any underlying files. (Inherited from <code>BaseDatabase</code>.)</td>
</tr>
<tr>
<td><strong>ReadUncommitted</strong></td>
<td>If true, this database supports transactional read operations with degree 1 isolation. Read operations on the database may request the return of modified but not yet committed data. (Inherited from <code>BaseDatabase</code>.)</td>
</tr>
<tr>
<td><strong>Transactional</strong></td>
<td>If true, this database has been opened in a transactional mode. (Inherited from <code>BaseDatabase</code>.)</td>
</tr>
<tr>
<td><strong>Truncated</strong></td>
<td>If true, the underlying file was physically truncated upon open, discarding all previous databases it might have held. (Inherited from <code>BaseDatabase</code>.)</td>
</tr>
<tr>
<td><strong>Type</strong></td>
<td>The type of the underlying access method (and file format). This value may be used to determine the type of the database after an <code>Open(String, DatabaseConfig)</code> function. (Inherited from <code>BaseDatabase</code>.)</td>
</tr>
<tr>
<td><strong>UseMVCC</strong></td>
<td>If true, the database was opened with support for multiversion concurrency control. (Inherited from <code>BaseDatabase</code>.)</td>
</tr>
</tbody>
</table>
See Also

SecondaryQueueDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `SecondaryQueueDatabase` type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Close</td>
<td>Overloaded.</td>
</tr>
</tbody>
</table>

Although closing a database will close any open cursors, it is recommended that applications explicitly close all their Cursor objects before closing the database. The reason why is that when the cursor is explicitly closed, the memory allocated for it is reclaimed; however, this will not happen if you close a database while cursors are still opened.

The same rule, for the same reasons, hold true for **Transaction** objects. Simply make sure you resolve all your transaction objects before closing your database handle.

Because key/data pairs are cached in memory, applications should make a point to always either close database handles or sync their data to disk (using `Sync()` before exiting, to ensure that any data cached in main memory are reflected in the underlying file system.

When called on a database that is the primary database for a secondary index, the primary database should be closed only after all secondary indices referencing it have been closed.

When multiple threads are using the object concurrently, only a single thread may call the Close method.

The object may not be accessed again after Close is called, regardless of its outcome.
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Dispose</strong></td>
<td>Release the resources held by this object, and close the database if it's still open. (Inherited from <code>BaseDatabase</code>.)</td>
</tr>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <code>Object</code> is equal to the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>Exists</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>Get</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>GetBoth</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <code>Type</code> of the current instance. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>Open</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>PrintFastStats</strong></td>
<td>The statistical information is described by the <code>BTreeStats</code>, <code>HashStats</code>, <code>QueueStats</code>, and <code>RecnoStats</code> classes.</td>
</tr>
<tr>
<td><strong>PrintStats</strong></td>
<td>The statistical information is described by the <code>BTreeStats</code>, <code>HashStats</code>, <code>QueueStats</code>, and <code>RecnoStats</code> classes.</td>
</tr>
<tr>
<td><strong>SecondaryCursor</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>Sync</strong></td>
<td>Flush any cached information to disk. (Inherited from <code>BaseDatabase</code>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a <code>String</code> that represents the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>Truncate</strong></td>
<td>When called on a database configured with secondary indices, Truncate will truncate the primary database and all secondary indices. A count of the records discarded from the primary database is returned.</td>
</tr>
</tbody>
</table>
See Also

SecondaryQueueDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Although closing a database will close any open cursors, it is recommended that applications explicitly close all their Cursor objects before closing the database. The reason why is that when the cursor is explicitly closed, the memory allocated for it is reclaimed; however, this will not happen if you close a database while cursors are still opened.

The same rule, for the same reasons, hold true for Transaction objects. Simply make sure you resolve all your transaction objects before closing your database handle.

Because key/data pairs are cached in memory, applications should make a point to always either close database handles or sync their data to disk (using Sync() method) before exiting, to ensure that any data cached in main memory are reflected in the underlying file system.

When called on a database that is the primary database for a secondary index, the primary database should be closed only after all secondary indices referencing it have been closed.

When multiple threads are using the object concurrently, only a single thread may call the Close method.

The object may not be accessed again after Close is called, regardless of its outcome.
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Close()()</code></td>
<td>Flush any cached database information to disk, close any open <code>Cursor()()</code> objects, free any allocated resources, and close any underlying files. (Inherited from <code>BaseDatabase</code>.)</td>
</tr>
<tr>
<td><code>Close(Boolean)</code></td>
<td>Optionally flush any cached database information to disk, close any open <code>Cursor()()</code> objects, free any allocated resources, and close any underlying files. (Inherited from <code>BaseDatabase</code>.)</td>
</tr>
</tbody>
</table>
See Also

SecondaryQueueDatabase Class
SecondaryQueueDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryQueueDatabase Class  See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Cursor()</code></td>
<td>Create a database cursor.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>BaseDatabase</code>.)</td>
</tr>
<tr>
<td></td>
<td>Create a database cursor with the given configuration.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>BaseDatabase</code>.)</td>
</tr>
<tr>
<td><code>Cursor(CursorConfig)</code></td>
<td>Create a transactionally protected database cursor.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>BaseDatabase</code>.)</td>
</tr>
<tr>
<td><code>Cursor(Transaction)</code></td>
<td>Create a transactionally protected database cursor with the given configuration.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>BaseDatabase</code>.)</td>
</tr>
</tbody>
</table>
See Also

SecondaryQueueDatabase Class
SecondaryQueueDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryQueueDatabase Class

See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Delete(DatabaseEntry)</strong></td>
<td>Remove key/data pairs from the database. The key/data pair associated with key is discarded from the database. In the presence of duplicate key values, all records associated with the designated key will be discarded. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td><strong>Delete(DatabaseEntry, Transaction)</strong></td>
<td>Remove key/data pairs from the database. The key/data pair associated with key is discarded from the database. In the presence of duplicate key values, all records associated with the designated key will be discarded. (Inherited from BaseDatabase.)</td>
</tr>
</tbody>
</table>
See Also

SecondaryQueueDatabase Class
SecondaryQueueDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation
SecondaryQueueDatabase...: Exists Method
SecondaryQueueDatabase Class  See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exists(DatabaseEntry)</td>
<td>Check whether key appears in the database. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>Exists(DatabaseEntry, Transaction)</td>
<td>Check whether key appears in the database. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>Exists(DatabaseEntry, Transaction, LockingInfo)</td>
<td>Check whether key appears in the database. (Inherited from BaseDatabase.)</td>
</tr>
</tbody>
</table>
See Also

SecondaryQueueDatabase Class
SecondaryQueueDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Get(DatabaseEntry)</td>
<td>Retrieve a key/data pair from the database. In the presence of duplicate key values, Get will return the first data item for key. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>Get(DatabaseEntry, Transaction)</td>
<td>Retrieve a key/data pair from the database. In the presence of duplicate key values, Get will return the first data item for key. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>Get(DatabaseEntry, Transaction, LockingInfo)</td>
<td>Retrieve a key/data pair from the database. In the presence of duplicate key values, Get will return the first data item for key. (Inherited from BaseDatabase.)</td>
</tr>
</tbody>
</table>
See Also

SecondaryQueueDatabase Class
SecondaryQueueDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
**SecondaryQueueDatabase Class**  
**See Also**
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>GetBoth(DatabaseEntry, DatabaseEntry)</code></td>
<td>Retrieve a key/data pair from the database which matches key and data. (Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
<tr>
<td><code>GetBoth(DatabaseEntry, DatabaseEntry, Transaction)</code></td>
<td>Retrieve a key/data pair from the database which matches key and data. (Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
<tr>
<td><code>GetBoth(DatabaseEntry, DatabaseEntry, Transaction, LockingInfo)</code></td>
<td>Retrieve a key/data pair from the database which matches key and data. (Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
</tbody>
</table>
See Also

SecondaryQueueDatabase Class
SecondaryQueueDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryQueueDatabase Class  See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open(String, SecondaryQueueDatabaseConfig)</td>
<td>Instantiate a new SecondaryQueueDatabase object, open the database represented by Filename and associate the database with the <strong>primary index</strong>.</td>
</tr>
<tr>
<td>Open(String, SecondaryQueueDatabaseConfig, Transaction)</td>
<td>Instantiate a new SecondaryQueueDatabase object, open the database represented by Filename and associate the database with the <strong>primary index</strong>.</td>
</tr>
</tbody>
</table>
See Also

SecondaryQueueDatabase Class
SecondaryQueueDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Instantiate a new SecondaryQueueDatabase object, open the database represented by Filename and associate the database with the primary index.

**Namespace:**  BerkeleyDB  
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll)  
**Version:**  4.8.24.0
**Syntax**

**C#**

```csharp
public static SecondaryQueueDatabase Open(
    string Filename,
    SecondaryQueueDatabaseConfig cfg
)
```

**Visual Basic (Declaration)**

```vbnet
Public Shared Function Open ( _
    Filename As String, _
    cfg As SecondaryQueueDatabaseConfig _
) As SecondaryQueueDatabase
```

**Visual C++**

```cpp
public:
static SecondaryQueueDatabase^ Open(
    String^ Filename,
    SecondaryQueueDatabaseConfig^ cfg
)
```

**Parameters**

**Filename**

Type: `System::String`

The name of an underlying file that will be used to back the database. In-memory databases never intended to be preserved on disk may be created by setting this parameter to null.

**cfg**

Type: `BerkeleyDB::SecondaryQueueDatabaseConfig`

The database's configuration

**Return Value**
A new, open database object
Remarks

If Filename is null, the database is strictly temporary and cannot be opened by any other thread of control, thus the database can only be accessed by sharing the single database object that created it, in circumstances where doing so is safe.

If AutoCommit is set, the operation will be implicitly transaction protected. Note that transactionally protected operations on a database object requires the object itself be transactionally protected during its open.
See Also

SecondaryQueueDatabase Class
Open Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryQueueDatabase Class  See Also

Instantiate a new SecondaryQueueDatabase object, open the database represented by Filename and associate the database with the primary index.

Namespace:  BerkeleyDB
### Syntax

**C#**

```csharp
public static SecondaryQueueDatabase Open(
    string Filename,
    SecondaryQueueDatabaseConfig cfg,
    Transaction txn
)
```

**Visual Basic (Declaration)**

```vbnet
Public Shared Function Open (_
    Filename As String, _
    cfg As SecondaryQueueDatabaseConfig, _
    txn As Transaction _
) As SecondaryQueueDatabase
```

**Visual C++**

```cpp
public:
static SecondaryQueueDatabase^ Open(
    String^ Filename,
    SecondaryQueueDatabaseConfig^ cfg,
    Transaction^ txn
)
```

### Parameters

**Filename**

Type: `System::String`

The name of an underlying file that will be used to back the database. In-memory databases never intended to be preserved on disk may be created by setting this parameter to null.

**cfg**

Type: `BerkeleyDB::SecondaryQueueDatabaseConfig`

The database’s configuration
txn

Type: **BerkeleyDB::::Transaction**

If the operation is part of an application-specified transaction, txn is a Transaction object returned from `BeginTransaction()`; if the operation is part of a Berkeley DB Concurrent Data Store group, txn is a handle returned from `BeginCDSGroup()`; otherwise null.

**Return Value**

A new, open database object
Remarks

If Filename is null, the database is strictly temporary and cannot be opened by any other thread of control, thus the database can only be accessed by sharing the single database object that created it, in circumstances where doing so is safe.

If txn is null, but AutoCommit is set, the operation will be implicitly transaction protected. Note that transactionally protected operations on a database object requires the object itself be transactionally protected during its open. Also note that the transaction must be committed before the object is closed.
See Also

SecondaryQueueDatabase Class
Open Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The statistical information is described by the BTreeStats, HashStats, QueueStats, and RecnoStats classes.
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PrintFastStats()()</td>
<td>Display the database statistical information which does not require traversal of the database. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>PrintFastStats(Boolean)</td>
<td>Display the database statistical information which does not require traversal of the database. (Inherited from BaseDatabase.)</td>
</tr>
</tbody>
</table>
See Also

SecondaryQueueDatabase Class
SecondaryQueueDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The statistical information is described by the BTreeStats, HashStats, QueueStats, and RecnoStats classes.
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>PrintStats()</code></td>
<td>Display the database statistical information. (Inherited from <code>BaseDatabase</code>.)</td>
</tr>
<tr>
<td><code>PrintStats(Boolean)</code></td>
<td>Display the database statistical information. (Inherited from <code>BaseDatabase</code>.)</td>
</tr>
</tbody>
</table>
See Also

SecondaryQueueDatabase Class
SecondaryQueueDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryQueueDatabase Class  See Also
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>SecondaryCursor()</code></td>
<td>Create a secondary database cursor. (Inherited from <a href="#">SecondaryDatabase</a>.)</td>
</tr>
<tr>
<td><code>SecondaryCursor(CursorConfig)</code></td>
<td>Create a secondary database cursor with the given configuration. (Inherited from <a href="#">SecondaryDatabase</a>.)</td>
</tr>
<tr>
<td><code>SecondaryCursor(Transaction)</code></td>
<td>Create a transactionally protected secondary database cursor. (Inherited from <a href="#">SecondaryDatabase</a>.)</td>
</tr>
<tr>
<td><code>SecondaryCursor(CursorConfig, Transaction)</code></td>
<td>Create a transactionally protected secondary database cursor with the given configuration. (Inherited from <a href="#">SecondaryDatabase</a>.)</td>
</tr>
</tbody>
</table>
See Also

SecondaryQueueDatabase Class  
SecondaryQueueDatabase Members  
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
When called on a database configured with secondary indices, Truncate will truncate the primary database and all secondary indices. A count of the records discarded from the primary database is returned.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Truncate()</strong></td>
<td>Empty the database, discarding all records it contains.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
<tr>
<td><strong>Truncate(Transaction)</strong></td>
<td>Empty the database, discarding all records it contains.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
</tbody>
</table>
See Also

SecondaryQueueDatabase Class
SecondaryQueueDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryQueueDatabase Properties

See Also

The SecondaryQueueDatabase type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AutoCommit</strong></td>
<td>If true, all database modification operations based on this object will be transactionally protected.</td>
</tr>
<tr>
<td><strong>CacheSize</strong></td>
<td>The size of the shared memory buffer pool -- that is, the cache.</td>
</tr>
<tr>
<td><strong>Creation</strong></td>
<td>The CreatePolicy with which this database was opened.</td>
</tr>
<tr>
<td><strong>DatabaseName</strong></td>
<td>The name of this database, if it has one.</td>
</tr>
<tr>
<td><strong>DoChecksum</strong></td>
<td>If true, do checksum verification of pages read into the cache from the backing filestore.</td>
</tr>
<tr>
<td><strong>EncryptAlgorithm</strong></td>
<td>The algorithm used by the Berkeley DB library to perform encryption and decryption.</td>
</tr>
<tr>
<td><strong>Encrypted</strong></td>
<td>If true, encrypt all data stored in the database.</td>
</tr>
<tr>
<td><strong>Endianness</strong></td>
<td>The database byte order.</td>
</tr>
<tr>
<td><strong>ErrorFeedback</strong></td>
<td>The mechanism for reporting detailed error messages to the application.</td>
</tr>
<tr>
<td><strong>ErrorPrefix</strong></td>
<td>The prefix string that appears before error messages issued by Berkeley DB.</td>
</tr>
<tr>
<td><strong>ExtentSize</strong></td>
<td>The size of the extents used to hold pages in a QueueDatabase, specified as a number of pages.</td>
</tr>
<tr>
<td><strong>Feedback</strong></td>
<td>Monitor progress within long running operations.</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>FileName</strong></td>
<td>The filename of this database, if it has one. (Inherited from <a href="#">BaseDatabase</a>) If true, the object is free-threaded; that is, concurrently usable by multiple threads in the address space. (Inherited from <a href="#">BaseDatabase</a>)</td>
</tr>
<tr>
<td><strong>FreeThreaded</strong></td>
<td>If true, the object is free-threaded; that is, concurrently usable by multiple threads in the address space. (Inherited from <a href="#">BaseDatabase</a>)</td>
</tr>
<tr>
<td><strong>HasMultiple</strong></td>
<td>If true, the object references a physical file supporting multiple databases. (Inherited from <a href="#">BaseDatabase</a>) If true, the underlying database files were created on an architecture of the same byte order as the current one. This information may be used to determine whether application data needs to be adjusted for this architecture or not. (Inherited from <a href="#">BaseDatabase</a>)</td>
</tr>
<tr>
<td><strong>InHostOrder</strong></td>
<td>If true, Berkeley DB will not write log records for this database. (Inherited from <a href="#">BaseDatabase</a>) If true, the underlying database files were created on an architecture of the same byte order as the current one. This information may be used to determine whether application data needs to be adjusted for this architecture or not. (Inherited from <a href="#">BaseDatabase</a>)</td>
</tr>
<tr>
<td><strong>KeyGen</strong></td>
<td>The delegate that creates the set of secondary keys corresponding to a given primary key and data pair. (Inherited from <a href="#">SecondaryDatabase</a>)</td>
</tr>
<tr>
<td><strong>Length</strong></td>
<td>The length of records in the database. If true, this database is not mapped into process memory. See <a href="#">MMapSize</a> for further information. (Inherited from <a href="#">BaseDatabase</a>)</td>
</tr>
<tr>
<td><strong>NoMMap</strong></td>
<td>See <a href="#">MMapSize</a> for further information. (Inherited from <a href="#">BaseDatabase</a>)</td>
</tr>
<tr>
<td><strong>NonDurableTxns</strong></td>
<td>If true, Berkeley DB will not write log records for this database. (Inherited from <a href="#">BaseDatabase</a>)</td>
</tr>
<tr>
<td><strong>Nullifier</strong></td>
<td>(Inherited from <a href="#">SecondaryDatabase</a>) The padding character for short, fixed-length records.</td>
</tr>
<tr>
<td><strong>Pagesize</strong></td>
<td>The database's current page size. (Inherited from <a href="#">BaseDatabase</a>) The cache priority for pages referenced by this object.</td>
</tr>
<tr>
<td><strong>Priority</strong></td>
<td>(Inherited from <a href="#">BaseDatabase</a>) The cache priority for pages referenced by this object.</td>
</tr>
</tbody>
</table>
If true, this database has been opened for reading only. Any attempt to modify items in the database will fail, regardless of the actual permissions of any underlying files.
(Inherited from BaseDatabase.)

If true, this database supports transactional read operations with degree 1 isolation. Read operations on the database may request the return of modified but not yet committed data.
(Inherited from BaseDatabase.)

If true, this database has been opened in a transactional mode.
(Inherited from BaseDatabase.)

If true, the underlying file was physically truncated upon open, discarding all previous databases it might have held.
(Inherited from BaseDatabase.)

The type of the underlying access method (and file format). This value may be used to determine the type of the database after an Open(String, DatabaseConfig).
(Inherited from BaseDatabase.)

If true, the database was opened with support for multiversion concurrency control.
(Inherited from BaseDatabase.)
See Also

SecondaryQueueDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The size of the extents used to hold pages in a `QueueDatabase`, specified as a number of pages.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public uint ExtentSize { get; }

Visual Basic (Declaration)

Public ReadOnly Property ExtentSize As UInteger

Visual C++

public:
property unsigned int ExtentSize {
unsigned int get ();
}
See Also

SecondaryQueueDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The length of records in the database.

Namespace:  BerkeleyDB
Syntax

C#

public uint Length { get; }

Visual Basic (Declaration)

Public ReadOnly Property Length AsUInteger

Visual C++

public:
property unsigned int Length {
    unsigned int get ();
}

See Also

SecondaryQueueDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The padding character for short, fixed-length records.

**Namespace:**  BerkeleyDB

**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public int PadByte { get; }

Visual Basic (Declaration)

Public ReadOnly Property PadByte As Integer

Visual C++

public:
property int PadByte {
    int get ();
}

See Also

SecondaryQueueDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
A class representing configuration parameters for SecondaryQueueDatabase

**Namespace:** BerkeleyDB
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public class SecondaryQueueDatabaseConfig : SecondaryDatabaseConfig

Visual Basic (Declaration)

Public Class SecondaryQueueDatabaseConfig
    Inherits SecondaryDatabaseConfig

Visual C++

public ref class SecondaryQueueDatabaseConfig : public SecondaryDatabaseConfig
Inheritance Hierarchy

System:::Object
BerkeleyDB:::DatabaseConfig
   BerkeleyDB:::SecondaryDatabaseConfig
      BerkeleyDB:::SecondaryQueueDatabaseConfig
See Also

SecondaryQueueDatabaseConfig Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `SecondaryQueueDatabaseConfig` type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>SecondaryQueueDatabaseConfig</code></td>
<td>Instantiate a new <code>SecondaryQueueDatabaseConfig</code> object</td>
</tr>
</tbody>
</table>
### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a>.</td>
</tr>
<tr>
<td>(Inherited from <a href="#">Object</a>.)</td>
<td></td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type.</td>
</tr>
<tr>
<td>(Inherited from <a href="#">Object</a>.)</td>
<td></td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <a href="#">Type</a> of the current instance.</td>
</tr>
<tr>
<td>(Inherited from <a href="#">Object</a>.)</td>
<td></td>
</tr>
<tr>
<td><strong>SetEncryption</strong></td>
<td>Set the password and algorithm used by the Berkeley DB library to perform encryption and decryption.</td>
</tr>
<tr>
<td>(Inherited from <a href="#">DatabaseConfig</a>.)</td>
<td></td>
</tr>
<tr>
<td><strong>SetForeignKeyConstraint</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a <a href="#">String</a> that represents the current <a href="#">Object</a>.</td>
</tr>
<tr>
<td>(Inherited from <a href="#">Object</a>.)</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>AutoCommit</td>
<td>Enclose the open call within a transaction. If the call succeeds, the open operation will be recoverable and all subsequent database modification operations based on this handle will be transactionally protected. If the call fails, no database will have been created. (Inherited from <code>DatabaseConfig</code>.)</td>
</tr>
<tr>
<td>ByteOrder</td>
<td>The byte order for integers in the stored database metadata. The host byte order of the machine where the Berkeley DB library was compiled is the default value. (Inherited from <code>DatabaseConfig</code>.)</td>
</tr>
<tr>
<td>CacheSize</td>
<td>The size of the shared memory buffer pool -- that is, the cache. (Inherited from <code>DatabaseConfig</code>.)</td>
</tr>
<tr>
<td>Creation</td>
<td>The policy for how to handle database creation.</td>
</tr>
<tr>
<td>DoChecksum</td>
<td>If true, do checksum verification of pages read into the cache from the backing filestore. (Inherited from <code>DatabaseConfig</code>.)</td>
</tr>
<tr>
<td>Env</td>
<td>The Berkeley DB environment within which to create a database. If null, the database will be created stand-alone; that is, it is not part of any Berkeley DB environment. (Inherited from <code>DatabaseConfig</code>.)</td>
</tr>
<tr>
<td>ErrorFeedback</td>
<td>The mechanism for reporting error messages to the application. (Inherited from <code>DatabaseConfig</code>.)</td>
</tr>
<tr>
<td>ErrorPrefix</td>
<td>The prefix string that appears before error messages issued by Berkeley DB. (Inherited from <code>DatabaseConfig</code>.)</td>
</tr>
<tr>
<td>Feedback</td>
<td>(Inherited from <code>DatabaseConfig</code>.)</td>
</tr>
</tbody>
</table>
**FreeThreaded**  
Cause the database object to be free-threaded; that is, concurrently usable by multiple threads in the address space.  
(Inherited from [DatabaseConfig](#).)

**ImmutableKey**  
If true, the secondary key is immutable.  
(Inherited from [SecondaryDatabaseConfig](#).)

**NoMMap**  
Do not map this database into process memory.  
(Inherited from [DatabaseConfig](#).)

**NoDurableTxns**  
If true, Berkeley DB will not write log records for this database.  
(Inherited from [DatabaseConfig](#).)

**Populate**  
If true and the secondary database is empty, walk through Primary and create an index to it in the empty secondary. This operation is potentially very expensive.  
(Inherited from [SecondaryDatabaseConfig](#).)

**Priority**  
The cache priority for pages referenced by the database.  
(Inherited from [DatabaseConfig](#).)

**ReadOnly**  
Open the database for reading only. Any attempt to modify items in the database will fail, regardless of the actual permissions of any underlying files.  
(Inherited from [DatabaseConfig](#).)

**ReadUncommitted**  
Support transactional read operations with degree 1 isolation.  
(Inherited from [DatabaseConfig](#).)

**Truncate**  
Physically truncate the underlying file, discarding all previous databases it might have held.  
(Inherited from [DatabaseConfig](#).)

**UseMVCC**  
Open the database with support for multiversion concurrency control.  
(Inherited from [DatabaseConfig](#).)
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EncryptAlgorithm</td>
<td>The algorithm used to perform encryption and decryption.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">DatabaseConfig</a>.)</td>
</tr>
<tr>
<td>EncryptionPassword</td>
<td>The password used to perform encryption and decryption.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">DatabaseConfig</a>.)</td>
</tr>
<tr>
<td>ExtentSize</td>
<td>The size of the extents used to hold pages in a <a href="#">SecondaryQueueDatabase</a>, specified as a number of pages.</td>
</tr>
<tr>
<td>ForeignKeyDatabase</td>
<td>(Inherited from <a href="#">SecondaryDatabaseConfig</a>.)</td>
</tr>
<tr>
<td>ForeignKeyNullifier</td>
<td>(Inherited from <a href="#">SecondaryDatabaseConfig</a>.)</td>
</tr>
<tr>
<td>KeyGen</td>
<td>The delegate that creates the set of secondary keys corresponding to a given primary key and data pair.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">SecondaryDatabaseConfig</a>.)</td>
</tr>
<tr>
<td>Length</td>
<td>Specify the length of records in the database.</td>
</tr>
<tr>
<td>OnForeignKeyDelete</td>
<td>(Inherited from <a href="#">SecondaryDatabaseConfig</a>.)</td>
</tr>
<tr>
<td>PadByte</td>
<td>The padding character for short, fixed-length records.</td>
</tr>
<tr>
<td>PageSize</td>
<td>The size of the pages used to hold items in the database, in bytes.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">DatabaseConfig</a>.)</td>
</tr>
<tr>
<td>Primary</td>
<td>All updates to Primary will be automatically reflected in the secondary and all reads from the secondary will return corresponding data from Primary.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">SecondaryDatabaseConfig</a>.)</td>
</tr>
</tbody>
</table>
See Also

SecondaryQueueDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation
SecondaryQueueDatabaseConfig Constructor

**SecondaryQueueDatabaseConfig Class**  **See Also**

Instantiate a new SecondaryQueueDatabaseConfig object

**Namespace:**  [BerkeleyDB](#)
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
### Syntax

**C#**

```csharp
public SecondaryQueueDatabaseConfig(
    Database PrimaryDB,
    SecondaryKeyGenDelegate KeyGenFunc
)
```

**Visual Basic (Declaration)**

```vbnet
Public Sub New (_
    PrimaryDB As Database, _
    KeyGenFunc As SecondaryKeyGenDelegate _
)
```

**Visual C++**

```cpp
public:
SecondaryQueueDatabaseConfig(
    Database^ PrimaryDB,
    SecondaryKeyGenDelegate^ KeyGenFunc
)
```

### Parameters

**PrimaryDB**
- Type: [BerkeleyDB::Database](#)

**KeyGenFunc**
- Type: [BerkeleyDB::SecondaryKeyGenDelegate](#)
See Also

SecondaryQueueDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `SecondaryQueueDatabaseConfig` type exposes the following members.
## Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AutoCommit</strong></td>
<td>Enclose the open call within a transaction. If the call succeeds, the open operation will be recoverable and all subsequent database modification operations based on this handle will be transactionally protected. If the call fails, no database will have been created. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td><strong>ByteOrder</strong></td>
<td>The byte order for integers in the stored database metadata. The host byte order of the machine where the Berkeley DB library was compiled is the default value. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td><strong>CacheSize</strong></td>
<td>The size of the shared memory buffer pool -- that is, the cache. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td><strong>Creation</strong></td>
<td>The policy for how to handle database creation.</td>
</tr>
<tr>
<td><strong>DoChecksum</strong></td>
<td>If true, do checksum verification of pages read into the cache from the backing filestore. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td><strong>Env</strong></td>
<td>The Berkeley DB environment within which to create a database. If null, the database will be created stand-alone; that is, it is not part of any Berkeley DB environment. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td><strong>ErrorFeedback</strong></td>
<td>The mechanism for reporting error messages to the application. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td><strong>ErrorPrefix</strong></td>
<td>The prefix string that appears before error messages issued by Berkeley DB. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td><strong>Feedback</strong></td>
<td>(Inherited from DatabaseConfig.)</td>
</tr>
</tbody>
</table>
- **FreeThreaded**: Cause the database object to be free-threaded; that is, concurrently usable by multiple threads in the address space. (Inherited from [DatabaseConfig](#).)

- **ImmutableKey**: If true, the secondary key is immutable. (Inherited from [SecondaryDatabaseConfig](#).)

- **NoMMap**: Do not map this database into process memory. (Inherited from [DatabaseConfig](#).)

- **NonDurableTxns**: If true, Berkeley DB will not write log records for this database. (Inherited from [DatabaseConfig](#).)

- **Populate**: If true and the secondary database is empty, walk through Primary and create an index to it in the empty secondary. This operation is potentially very expensive. (Inherited from [SecondaryDatabaseConfig](#).)

- **Priority**: The cache priority for pages referenced by the database. (Inherited from [DatabaseConfig](#).)

- **ReadOnly**: Open the database for reading only. Any attempt to modify items in the database will fail, regardless of the actual permissions of any underlying files. (Inherited from [DatabaseConfig](#).)

- **ReadUncommitted**: Support transactional read operations with degree 1 isolation. (Inherited from [DatabaseConfig](#).)

- **Truncate**: Physically truncate the underlying file, discarding all previous databases it might have held. (Inherited from [DatabaseConfig](#).)

- **UseMVCC**: Open the database with support for multiversion concurrency control. (Inherited from [DatabaseConfig](#).)
See Also

SecondaryQueueDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryQueueDatabaseConfig::Creation Field

The policy for how to handle database creation.

Namespace: BerkeleyDB
Syntax

C#
public CreatePolicy Creation

Visual Basic (Declaration)
Public Creation As CreatePolicy

Visual C++
public:
CreatePolicy Creation
Remarks

If the database does not already exist and NEVER is set, Open(String, SecondaryQueueDatabaseConfig) will fail.
See Also

SecondaryQueueDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The SecondaryQueueDatabaseConfig type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <strong>Object</strong> is equal to the current <strong>Object</strong>. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <strong>Type</strong> of the current instance. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>SetEncryption</strong></td>
<td>Set the password and algorithm used by the Berkeley DB library to perform encryption and decryption. (Inherited from <strong>DatabaseConfig</strong>.)</td>
</tr>
<tr>
<td><strong>SetForeignKeyConstraint</strong></td>
<td>Overloaded. Returns a <strong>String</strong> that represents the current <strong>Object</strong>. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td></td>
</tr>
</tbody>
</table>
See Also

SecondaryQueueDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryQueueDatabaseConfig::SetForeignKeyConstraint Method

SecondaryQueueDatabaseConfig Class  See Also
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SetForeignKeyConstraint(Database, ForeignKeyDeleteAction)</td>
<td>(Inherited from SecondaryDatabaseConfig.)</td>
</tr>
<tr>
<td>SetForeignKeyConstraint(Database, ForeignKeyDeleteAction, ForeignKeyNullifyDelegate)</td>
<td>(Inherited from SecondaryDatabaseConfig.)</td>
</tr>
</tbody>
</table>
See Also

SecondaryQueueDatabaseConfig Class
SecondaryQueueDatabaseConfig Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation
SecondaryQueueDatabaseConfig Properties
SecondaryQueueDatabaseConfig Class  See Also

The SecondaryQueueDatabaseConfig type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EncryptAlgorithm</td>
<td>The algorithm used to perform encryption and decryption. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td>EncryptionPassword</td>
<td>The password used to perform encryption and decryption. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td>ExtentSize</td>
<td>The size of the extents used to hold pages in a SecondaryQueueDatabase, specified as a number of pages.</td>
</tr>
<tr>
<td>ForeignKeyDatabase</td>
<td>(Inherited from SecondaryDatabaseConfig.)</td>
</tr>
<tr>
<td>ForeignKeyNullifier</td>
<td>(Inherited from SecondaryDatabaseConfig.)</td>
</tr>
<tr>
<td>KeyGen</td>
<td>The delegate that creates the set of secondary keys corresponding to a given primary key and data pair. (Inherited from SecondaryDatabaseConfig.)</td>
</tr>
<tr>
<td>Length</td>
<td>Specify the length of records in the database.</td>
</tr>
<tr>
<td>OnForeignKeyDelete</td>
<td>(Inherited from SecondaryDatabaseConfig.)</td>
</tr>
<tr>
<td>PadByte</td>
<td>The padding character for short, fixed-length records.</td>
</tr>
<tr>
<td>PageSize</td>
<td>The size of the pages used to hold items in the database, in bytes. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td>Primary</td>
<td>All updates to Primary will be automatically reflected in the secondary and all reads from the secondary will return corresponding data from Primary. (Inherited from SecondaryDatabaseConfig.)</td>
</tr>
</tbody>
</table>
See Also

SecondaryQueueDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryQueueDatabaseConfig::ExtentSize Property

SecondaryQueueDatabaseConfig Class  See Also

The size of the extents used to hold pages in a SecondaryQueueDatabase, specified as a number of pages.

Namespace:  BerkeleyDB
Syntax

C#

public uint ExtentSize { get; set; }

Visual Basic (Declaration)

Public Property ExtentSize As UInteger

Visual C++

public:
property unsigned int ExtentSize {
unsigned int get ();
void set (unsigned int value);
}

Remarks

Each extent is created as a separate physical file. If no extent size is set, the default behavior is to create only a single underlying database file.

For information on tuning the extent size, see Selecting a extent size in the Programmer's Reference Guide.

If the database already exists, this setting will be ignored.
See Also

SecondaryQueueDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Specify the length of records in the database.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public uint Length { get; set; }

Visual Basic (Declaration)

Public Property Length As UInteger

Visual C++

public:
    property unsigned int Length {
        unsigned int get ();
        void set (unsigned int value);
    }
}
Remarks

The record length must be enough smaller than PageSize that at least one record plus the database page's metadata information can fit on each database page.

Any records added to the database that are less than Length bytes long are automatically padded (see PadByte for more information).

Any attempt to insert records into the database that are greater than Length bytes long will cause the call to fail immediately and return an error.

If the database already exists, this setting will be ignored.
See Also

SecondaryQueueDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryQueueDatabaseConfig
.
.
::
.
.
PadByte Property

SecondaryQueueDatabaseConfig Class  See Also

The padding character for short, fixed-length records.

Namespace:  BerkeleyDB
**Syntax**

**C#**

```csharp
public int PadByte { get; set; }
```

**Visual Basic (Declaration)**

```
Public Property PadByte As Integer
```

**Visual C++**

```c++
public:
property int PadByte {
    int get ();
    void set (int value);
}
```
Remarks

If no pad character is specified, space characters (that is, ASCII 0x20) are used for padding.

If the database already exists, this setting will be ignored.
See Also

SecondaryQueueDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryRecnoDatabase Class

A class representing a RecnoDatabase. The Recno format supports fixed- or variable-length records, accessed sequentially or by logical record number, and optionally backed by a flat text file.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public class SecondaryRecnoDatabase : SecondaryDatabase

Visual Basic (Declaration)

Public Class SecondaryRecnoDatabase _
Inherits SecondaryDatabase

Visual C++

public ref class SecondaryRecnoDatabase : public SecondaryDatabase
Inheritance Hierarchy

System:::Object
BerkeleyDB:::BaseDatabase
BerkeleyDB:::SecondaryDatabase
BerkeleyDB:::SecondaryRecnoDatabase
See Also

SecondaryRecnoDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The **SecondaryRecnoDatabase** type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Close</td>
<td>Overloaded.</td>
</tr>
</tbody>
</table>

Although closing a database will close any open cursors, it is recommended that applications explicitly close all their Cursor objects before closing the database. The reason why is that when the cursor is explicitly closed, the memory allocated for it is reclaimed; however, this will not happen if you close a database while cursors are still opened.

The same rule, for the same reasons, hold true for `Transaction` objects. Simply make sure you resolve all your transaction objects before closing your database handle.

Because key/data pairs are cached in memory, applications should make a point to always either close database handles or sync their data to disk (using `SyncOOO`) before exiting, to ensure that any data cached in main memory are reflected in the underlying file system.

When called on a database that is the primary database for a secondary index, the primary database should be closed only after all secondary indices referencing it have been closed.

When multiple threads are using the object concurrently, only a single thread may call the Close method.

The object may not be accessed again after Close is called, regardless of its outcome.
- **Cursor** Overloaded.
- **Delete** Overloaded.
- **Dispose** Release the resources held by this object, and close the database if it's still open. (Inherited from BaseDatabase.)
- **Equals** Determines whether the specified Object is equal to the current Object. (Inherited from Object.)
- **Exists** Overloaded.
- **Get** Overloaded.
- **GetBoth** Overloaded.
- **GetHashCode** Serves as a hash function for a particular type. (Inherited from Object.)
- **GetType** Gets the Type of the current instance. (Inherited from Object.)
- **Open** Overloaded.
- **PrintFastStats** The statistical information is described by the BTreeStats, HashStats, QueueStats, and RecnoStats classes. Overloaded.
- **PrintStats** The statistical information is described by the BTreeStats, HashStats, QueueStats, and RecnoStats classes.
- **SecondaryCursor** Overloaded.
- **Sync** Flush any cached information to disk. (Inherited from BaseDatabase.)
- **ToString** Returns a String that represents the current Object. (Inherited from Object.) Overloaded.
- **Truncate** When called on a database configured with secondary indices, Truncate will truncate the primary database and all secondary indices. A count of the records discarded from the primary database is returned.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AutoCommit</td>
<td>If true, all database modification operations based on this object will be transactionally protected. (Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
<tr>
<td>BackingFile</td>
<td>The underlying source file for the Recno access method. The size of the shared memory buffer pool -- that is, the cache. (Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
<tr>
<td>CacheSize</td>
<td>The CreatePolicy with which this database was opened. (Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
<tr>
<td>Creation</td>
<td>The name of this database, if it has one. (Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
<tr>
<td>DatabaseName</td>
<td>The delimiting byte used to mark the end of a record in <a href="#">BackingFile</a>. If true, do checksum verification of pages read into the cache from the backing filestore. (Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
<tr>
<td>EncryptAlgorithm</td>
<td>The algorithm used by the Berkeley DB library to perform encryption and decryption. (Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
<tr>
<td>Encrypted</td>
<td>If true, encrypt all data stored in the database. (Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
<tr>
<td>Endianness</td>
<td>The database byte order. (Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
<tr>
<td>ErrorFeedback</td>
<td>The mechanism for reporting detailed error messages to the application. (Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
<tr>
<td>ErrorPrefix</td>
<td>The prefix string that appears before error messages issued by Berkeley DB. (Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Feedback</td>
<td>Monitor progress within long running operations. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>FileName</td>
<td>The filename of this database, if it has one. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>FreeThreaded</td>
<td>If true, the object is free-threaded; that is, concurrently usable by multiple threads in the address space. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>HasMultiple</td>
<td>If true, the object references a physical file supporting multiple databases. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>InHostOrder</td>
<td>If true, the underlying database files were created on an architecture of the same byte order as the current one. This information may be used to determine whether application data needs to be adjusted for this architecture or not. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>KeyGen</td>
<td>The delegate that creates the set of secondary keys corresponding to a given primary key and data pair. (Inherited from SecondaryDatabase.)</td>
</tr>
<tr>
<td>Length</td>
<td>If true, this database is not mapped into process memory. If using fixed-length, not byte-delimited records, the length of the records.</td>
</tr>
<tr>
<td>NoMMap</td>
<td>See MMapSize for further information. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>NonDurableTxns</td>
<td>If true, Berkeley DB will not write log records for this database. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>Nullifier</td>
<td>(Inherited from SecondaryDatabase.)</td>
</tr>
<tr>
<td>PadByte</td>
<td>The padding character for short, fixed-length records.</td>
</tr>
<tr>
<td>Pagesize</td>
<td>The database's current page size. (Inherited from BaseDatabase.)</td>
</tr>
</tbody>
</table>
**Priority**
The cache priority for pages referenced by this object.
(Inherited from BaseDatabase.)
If true, this database has been opened for reading only. Any attempt to modify items in the database will fail, regardless of the actual permissions of any underlying files.
(Inherited from BaseDatabase.)

**ReadOnly**
If true, this database has been opened for reading only. Any attempt to modify items in the database will fail, regardless of the actual permissions of any underlying files.
(Inherited from BaseDatabase.)

**ReadUncommitted**
If true, this database supports transactional read operations with degree 1 isolation. Read operations on the database may request the return of modified but not yet committed data.
(Inherited from BaseDatabase.)

**Renumber**
If true, the logical record numbers are mutable, and change as records are added to and deleted from the database.
If true, any BackingFile file will be read in its entirety when Open(String, SecondaryRecnoDatabaseConfig) is called. If false, BackingFile may be read lazily.

**Snapshot**
If true, any BackingFile file will be read in its entirety when Open(String, SecondaryRecnoDatabaseConfig) is called. If false, BackingFile may be read lazily.
If true, this database has been opened in a transactional mode.
(Inherited from BaseDatabase.)

**Transactional**
If true, the underlying file was physically truncated upon open, discarding all previous databases it might have held.
(Inherited from BaseDatabase.)

**Truncated**
The type of the underlying access method (and file format). This value may be used to determine the type of the database after an Open(String, DatabaseConfig).
(Inherited from BaseDatabase.)

**Type**
If true, the database was opened with support for multiversion concurrency control.
(Inherited from BaseDatabase.)

**UseMVCC**
See Also

SecondaryRecnoDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The **SecondaryRecnoDatabase** type exposes the following members.
Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overloaded.</td>
<td>Although closing a database will close any open cursors, it is recommended that applications explicitly close all their Cursor objects before closing the database. The reason why is that when the cursor is explicitly closed, the memory allocated for it is reclaimed; however, this will not happen if you close a database while cursors are still opened. The same rule, for the same reasons, hold true for Transaction objects. Simply make sure you resolve all your transaction objects before closing your database handle. Because key/data pairs are cached in memory, applications should make a point to always either close database handles or sync their data to disk (using SyncQOO) before exiting, to ensure that any data cached in main memory are reflected in the underlying file system. When called on a database that is the primary database for a secondary index, the primary database should be closed only after all secondary indices referencing it have been closed. When multiple threads are using the object concurrently, only a single thread may call the Close method. The object may not be accessed again after Close is called, regardless of its outcome.</td>
</tr>
</tbody>
</table>

- Close
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cursor</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>Delete</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>Dispose</td>
<td>Release the resources held by this object, and close the database if it's still open. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified Object is equal to the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Exists</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>Get</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>GetBoth</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as a hash function for a particular type. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>Open</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>PrintFastStats</td>
<td>The statistical information is described by the BTreeStats, HashStats, QueueStats, and RecnoStats classes. (Overloaded.)</td>
</tr>
<tr>
<td>PrintStats</td>
<td>The statistical information is described by the BTreeStats, HashStats, QueueStats, and RecnoStats classes.</td>
</tr>
<tr>
<td>SecondaryCursor</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>Sync</td>
<td>Flush any cached information to disk. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a String that represents the current Object. (Inherited from Object.) (Overloaded.)</td>
</tr>
<tr>
<td>Truncate</td>
<td>When called on a database configured with secondary indices, Truncate will truncate the primary database and all secondary indices. A count of the records discarded from the primary database is returned.</td>
</tr>
</tbody>
</table>
See Also

SecondaryRecnoDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Although closing a database will close any open cursors, it is recommended that applications explicitly close all their Cursor objects before closing the database. The reason why is that when the cursor is explicitly closed, the memory allocated for it is reclaimed; however, this will not happen if you close a database while cursors are still opened.

The same rule, for the same reasons, hold true for Transaction objects. Simply make sure you resolve all your transaction objects before closing your database handle.

Because key/data pairs are cached in memory, applications should make a point to always either close database handles or sync their data to disk (using Sync() before exiting, to ensure that any data cached in main memory are reflected in the underlying file system.

When called on a database that is the primary database for a secondary index, the primary database should be closed only after all secondary indices referencing it have been closed.

When multiple threads are using the object concurrently, only a single thread may call the Close method.

The object may not be accessed again after Close is called, regardless of its outcome.
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Close()</td>
<td>Flush any cached database information to disk, close any open <code>Cursor()</code> objects, free any allocated resources, and close any underlying files. (Inherited from <code>BaseDatabase</code>.)</td>
</tr>
<tr>
<td>Close(Boolean)</td>
<td>Optionally flush any cached database information to disk, close any open <code>Cursor()</code> objects, free any allocated resources, and close any underlying files. (Inherited from <code>BaseDatabase</code>.)</td>
</tr>
</tbody>
</table>
See Also

SecondaryRecnoDatabase Class
SecondaryRecnoDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Cursor()</code></td>
<td>Create a database cursor. (Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
<tr>
<td><code>Cursor(CursorConfig)</code></td>
<td>Create a database cursor with the given configuration. (Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
<tr>
<td><code>Cursor(Transaction)</code></td>
<td>Create a transactionally protected database cursor. (Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
<tr>
<td><code>Cursor(CursorConfig, Transaction)</code></td>
<td>Create a transactionally protected database cursor with the given configuration. (Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
</tbody>
</table>
See Also

SecondaryRecnoDatabase Class
SecondaryRecnoDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryRecnoDatabase Method

SecondaryRecnoDatabase Class  See Also
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delete(DatabaseEntry)</td>
<td>Remove key/data pairs from the database. The key/data pair associated with key is discarded from the database. In the presence of duplicate key values, all records associated with the designated key will be discarded. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>Delete(DatabaseEntry, Transaction)</td>
<td>Remove key/data pairs from the database. The key/data pair associated with key is discarded from the database. In the presence of duplicate key values, all records associated with the designated key will be discarded. (Inherited from BaseDatabase.)</td>
</tr>
</tbody>
</table>
See Also

SecondaryRecnoDatabase Class
SecondaryRecnoDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryRecnoDatabase...::Exists Method
SecondaryRecnoDatabase Class  See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exists(DatabaseEntry)</td>
<td>Check whether key appears in the database. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>Exists(DatabaseEntry, Transaction)</td>
<td>Check whether key appears in the database. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>Exists(DatabaseEntry, Transaction, LockingInfo)</td>
<td>Check whether key appears in the database. (Inherited from BaseDatabase.)</td>
</tr>
</tbody>
</table>
See Also

SecondaryRecnoDatabase Class
SecondaryRecnoDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryRecnoDatabase::Get Method

SecondaryRecnoDatabase Class  See Also
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Get(DatabaseEntry)</td>
<td>Retrieve a key/data pair from the database. In the presence of duplicate key values, Get will return the first data item for key. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>Get(DatabaseEntry, Transaction)</td>
<td>Retrieve a key/data pair from the database. In the presence of duplicate key values, Get will return the first data item for key. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>Get(DatabaseEntry, Transaction, LockingInfo)</td>
<td>Retrieve a key/data pair from the database. In the presence of duplicate key values, Get will return the first data item for key. (Inherited from BaseDatabase.)</td>
</tr>
</tbody>
</table>
See Also

SecondaryRecnoDatabase Class
SecondaryRecnoDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryRecnoDatabase Class  See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetBoth(DatabaseEntry, DatabaseEntry)</td>
<td>Retrieve a key/data pair from the database which matches key and data. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>GetBoth(DatabaseEntry, DatabaseEntry, Transaction)</td>
<td>Retrieve a key/data pair from the database which matches key and data. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>GetBoth(DatabaseEntry, DatabaseEntry, Transaction, LockingInfo)</td>
<td>Retrieve a key/data pair from the database which matches key and data. (Inherited from BaseDatabase.)</td>
</tr>
</tbody>
</table>
See Also

SecondaryRecnoDatabase Class
SecondaryRecnoDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
C#  Visual Basic  Visual C++
Include Protected Members
Include Inherited Members
Berkeley DB .NET API Documentation
SecondaryRecnoDatabase...:::Open Method
SecondaryRecnoDatabase Class  See Also
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open(String, SecondaryRecnoDatabaseConfig)</td>
<td>Instantiate a new SecondaryRecnoDatabase object, open the database represented by Filename and associate the database with the primary index.</td>
</tr>
<tr>
<td>Open(String, SecondaryRecnoDatabaseConfig, Transaction)</td>
<td>Instantiate a new SecondaryRecnoDatabase object, open the database represented by Filename and associate the database with the primary index.</td>
</tr>
<tr>
<td>Open(String, String, SecondaryRecnoDatabaseConfig)</td>
<td>Instantiate a new SecondaryRecnoDatabase object, open the database represented by Filename and associate the database with the primary index.</td>
</tr>
<tr>
<td>Open(String, String, SecondaryRecnoDatabaseConfig, Transaction)</td>
<td>Instantiate a new SecondaryRecnoDatabase object, open the database represented by Filename and associate the database with the primary index.</td>
</tr>
</tbody>
</table>
See Also

SecondaryRecnoDatabase Class
SecondaryRecnoDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryRecnoDatabase....Open Method (String, SecondaryRecnoDatabaseConfig)

Instantiate a new SecondaryRecnoDatabase object, open the database represented by Filename and associate the database with the primary index.

Namespace: BerkeleyDB
## Syntax

### C#

```csharp
public static SecondaryRecnoDatabase Open(
    string Filename,
    SecondaryRecnoDatabaseConfig cfg
)
```

### Visual Basic (Declaration)

```vbnet
Public Shared Function Open (
    Filename As String,
    cfg As SecondaryRecnoDatabaseConfig
) As SecondaryRecnoDatabase
```

### Visual C++

```cpp
public:
static SecondaryRecnoDatabase^ Open(
    String^ Filename,
    SecondaryRecnoDatabaseConfig^ cfg
)
```

## Parameters

**Filename**

Type: `System::String`

The name of an underlying file that will be used to back the database. In-memory databases never intended to be preserved on disk may be created by setting this parameter to null.

**cfg**

Type: `BerkeleyDB::SecondaryRecnoDatabaseConfig`

The database's configuration

## Return Value
A new, open database object
Remarks

If Filename is null, the database is strictly temporary and cannot be opened by any other thread of control, thus the database can only be accessed by sharing the single database object that created it, in circumstances where doing so is safe.

If AutoCommit is set, the operation will be implicitly transaction protected. Note that transactionally protected operations on a database object requires the object itself be transactionally protected during its open.
See Also

SecondaryRecnoDatabase Class
Open Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryRecnoDatabase.

::

Open Method (String, SecondaryRecnoDatabaseConfig, Transaction)

See Also

Instantiate a new SecondaryRecnoDatabase object, open the database represented by Filename and associate the database with the primary index.

Namespace: BerkeleyDB
Syntax

C#

public static SecondaryRecnoDatabase Open(
    string Filename,
    SecondaryRecnoDatabaseConfig cfg,
    Transaction txn
)

Visual Basic (Declaration)

Public Shared Function Open ( _
    Filename As String, _
    cfg As SecondaryRecnoDatabaseConfig, _
    txn As Transaction _
) As SecondaryRecnoDatabase

Visual C++

public:
    static SecondaryRecnoDatabase ^ Open(
        String ^ Filename,
        SecondaryRecnoDatabaseConfig ^ cfg,
        Transaction ^ txn
    )

Parameters

Filename
    Type: System::String
    The name of an underlying file that will be used to back the database. In-memory databases never intended to be preserved on disk may be created by setting this parameter to null.

cfg
    Type: BerkeleyDB::SecondaryRecnoDatabaseConfig
    The database’s configuration
txn

Type: BerkeleyDB::Transaction
If the operation is part of an application-specified transaction, txn is a Transaction object returned from BeginTransaction(); if the operation is part of a Berkeley DB Concurrent Data Store group, txn is a handle returned from BeginCDSGroup(); otherwise null.

Return Value

A new, open database object
Remarks

If Filename is null, the database is strictly temporary and cannot be opened by any other thread of control, thus the database can only be accessed by sharing the single database object that created it, in circumstances where doing so is safe.

If txn is null, but AutoCommit is set, the operation will be implicitly transaction protected. Note that transactionally protected operations on a database object requires the object itself be transactionally protected during its open. Also note that the transaction must be committed before the object is closed.
See Also

SecondaryRecnoDatabase Class
Open Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryRecnoDatabase..::..Open Method (String, String, SecondaryRecnoDatabaseConfig)

SecondaryRecnoDatabase Class  See Also

Instantiate a new SecondaryRecnoDatabase object, open the database represented by Filename and associate the database with the primary index.

Namespace:  BerkeleyDB
Syntax

C#

```csharp
public static SecondaryRecnoDatabase Open(
    string Filename,
    string DatabaseName,
    SecondaryRecnoDatabaseConfig cfg)
```

Visual Basic (Declaration)

```vbnet
Public Shared Function Open (_
    Filename As String, _
    DatabaseName As String, _
    cfg As SecondaryRecnoDatabaseConfig _
) As SecondaryRecnoDatabase
```

Visual C++

```cpp
public:
static SecondaryRecnoDatabase^ Open(
    String^ Filename,
    String^ DatabaseName,
    SecondaryRecnoDatabaseConfig^ cfg
)
```

Parameters

Filename
   Type: System::String
   The name of an underlying file that will be used to back the database. In-memory databases never intended to be preserved on disk may be created by setting this parameter to null.

DatabaseName
   Type: System::String
   This parameter allows applications to have multiple databases in a single file. Although no DatabaseName needs to be specified, it is an error to
attempt to open a second database in a file that was not initially created using a database name.

cfg
  Type: BerkeleyDB::<SecondaryRecnoDatabaseConfig>
The database's configuration

**Return Value**

A new, open database object
Remarks

If both Filename and DatabaseName are null, the database is strictly temporary and cannot be opened by any other thread of control, thus the database can only be accessed by sharing the single database object that created it, in circumstances where doing so is safe. If Filename is null and DatabaseName is non-null, the database can be opened by other threads of control and will be replicated to client sites in any replication group.

If AutoCommit is set, the operation will be implicitly transaction protected. Note that transactionally protected operations on a database object requires the object itself be transactionally protected during its open.
See Also

SecondaryRecnoDatabase Class
Open Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryRecnoDatabase::Open Method (String, String, SecondaryRecnoDatabaseConfig, Transaction)

See Also

Instantiate a new SecondaryRecnoDatabase object, open the database represented by Filename and associate the database with the primary index.

Namespace: BerkeleyDB
### Syntax

**C#**

```csharp
public static SecondaryRecnoDatabase Open(
    string Filename,
    string DatabaseName,
    SecondaryRecnoDatabaseConfig cfg,
    Transaction txn
)
```

**Visual Basic (Declaration)**

```vbnet
Public Shared Function Open ( 
    Filename As String, 
    DatabaseName As String, 
    cfg As SecondaryRecnoDatabaseConfig, 
    txn As Transaction 
) As SecondaryRecnoDatabase
```

**Visual C++**

```cpp
public:
static SecondaryRecnoDatabase^ Open( 
    String^ Filename, 
    String^ DatabaseName, 
    SecondaryRecnoDatabaseConfig^ cfg, 
    Transaction^ txn
)
```

### Parameters

**Filename**

- **Type:** `System::::String`
- The name of an underlying file that will be used to back the database. In-memory databases never intended to be preserved on disk may be created by setting this parameter to null.

**DatabaseName**

- **Type:** `System::::String`
This parameter allows applications to have multiple databases in a single file. Although no DatabaseName needs to be specified, it is an error to attempt to open a second database in a file that was not initially created using a database name.

**cfg**
Type: BerkeleyDB::SecondaryRecnoDatabaseConfig
The database's configuration

**txn**
Type: BerkeleyDB::Transaction
If the operation is part of an application-specified transaction, txn is a Transaction object returned from BeginTransaction(); if the operation is part of a Berkeley DB Concurrent Data Store group, txn is a handle returned from BeginCDSGroup(); otherwise null.

**Return Value**
A new, open database object
Remarks

If both Filename and DatabaseName are null, the database is strictly temporary and cannot be opened by any other thread of control, thus the database can only be accessed by sharing the single database object that created it, in circumstances where doing so is safe. If Filename is null and DatabaseName is non-null, the database can be opened by other threads of control and will be replicated to client sites in any replication group.

If txn is null, but AutoCommit is set, the operation will be implicitly transaction protected. Note that transactionally protected operations on a database object requires the object itself be transactionally protected during its open. Also note that the transaction must be committed before the object is closed.
See Also

SecondaryRecnoDatabase Class
Open Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The statistical information is described by the BTreeStats, HashStats, QueueStats, and RecnoStats classes.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PrintFastStats()()()</td>
<td>Display the database statistical information which does not require traversal of the database. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>PrintFastStats(Boolean)</td>
<td>Display the database statistical information which does not require traversal of the database. (Inherited from BaseDatabase.)</td>
</tr>
</tbody>
</table>
See Also

SecondaryRecnoDatabase Class
SecondaryRecnoDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The statistical information is described by the BTreeStats, HashStats, QueueStats, and RecnoStats classes.
### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PrintStats()</td>
<td>Display the database statistical information. (Inherited from BaseDatabase.)</td>
</tr>
<tr>
<td>PrintStats(Boolean)</td>
<td>Display the database statistical information. (Inherited from BaseDatabase.)</td>
</tr>
</tbody>
</table>
See Also

SecondaryRecnoDatabase Class
SecondaryRecnoDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryRecnoDatabase Class  See Also
### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SecondaryCursor()</strong></td>
<td>Create a secondary database cursor.</td>
</tr>
<tr>
<td>(Inherited from <a href="#">SecondaryDatabase</a>)</td>
<td>(Inherited from <a href="#">SecondaryDatabase</a>)</td>
</tr>
<tr>
<td><strong>SecondaryCursor(CursorConfig)</strong></td>
<td>Create a secondary database cursor with the given configuration.</td>
</tr>
<tr>
<td>(Inherited from <a href="#">SecondaryDatabase</a>)</td>
<td>(Inherited from <a href="#">SecondaryDatabase</a>)</td>
</tr>
<tr>
<td><strong>SecondaryCursor(Transaction)</strong></td>
<td>Create a transactionally protected secondary database cursor.</td>
</tr>
<tr>
<td>(Inherited from <a href="#">SecondaryDatabase</a>)</td>
<td>(Inherited from <a href="#">SecondaryDatabase</a>)</td>
</tr>
<tr>
<td><strong>SecondaryCursor(CursorConfig,</strong></td>
<td>Create a transactionally protected secondary database cursor with the</td>
</tr>
<tr>
<td>Transaction)**</td>
<td>given configuration.</td>
</tr>
<tr>
<td>(Inherited from <a href="#">SecondaryDatabase</a>)</td>
<td>(Inherited from <a href="#">SecondaryDatabase</a>)</td>
</tr>
</tbody>
</table>
See Also

SecondaryRecnoDatabase Class
SecondaryRecnoDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryRecnoDatabase...:..Truncate Method

When called on a database configured with secondary indices, Truncate will truncate the primary database and all secondary indices. A count of the records discarded from the primary database is returned.
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>TruncateQQQQ</code></td>
<td>Empty the database, discarding all records it contains.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
<tr>
<td><code>Truncate(Transaction)</code></td>
<td>Empty the database, discarding all records it contains.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">BaseDatabase</a>.)</td>
</tr>
</tbody>
</table>
See Also

SecondaryRecnoDatabase Class
SecondaryRecnoDatabase Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryRecnoDatabase Properties

See Also

The SecondaryRecnoDatabase type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AutoCommit</td>
<td>If true, all database modification operations based on this object will be transactionally protected.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">BaseDatabase</a>).</td>
</tr>
<tr>
<td>BackingFile</td>
<td>The underlying source file for the Recno access method.</td>
</tr>
<tr>
<td></td>
<td>The size of the shared memory buffer pool -- that is, the cache.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">BaseDatabase</a>).</td>
</tr>
<tr>
<td>CacheSize</td>
<td>The CreatePolicy with which this database was opened.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">BaseDatabase</a>).</td>
</tr>
<tr>
<td>Creation</td>
<td>The name of this database, if it has one.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">BaseDatabase</a>).</td>
</tr>
<tr>
<td>DatabaseName</td>
<td>The delimiting byte used to mark the end of a record in <a href="#">BackingFile</a>.</td>
</tr>
<tr>
<td></td>
<td>If true, do checksum verification of pages read into the cache from the backing filestore.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">BaseDatabase</a>).</td>
</tr>
<tr>
<td>Delimiter</td>
<td>The algorithm used by the Berkeley DB library to perform encryption and decryption.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">BaseDatabase</a>).</td>
</tr>
<tr>
<td>DoChecksum</td>
<td>If true, encrypt all data stored in the database.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">BaseDatabase</a>).</td>
</tr>
<tr>
<td>EncryptAlgorithm</td>
<td>The database byte order.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">BaseDatabase</a>).</td>
</tr>
<tr>
<td>Encrypted</td>
<td>The mechanism for reporting detailed error messages to the application.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">BaseDatabase</a>).</td>
</tr>
<tr>
<td>Endianness</td>
<td>The prefix string that appears before error messages issued by Berkeley DB.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">BaseDatabase</a>).</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Feedback</td>
<td>Monitor progress within long running operations. (Inherited from <code>BaseDatabase</code>.)</td>
</tr>
<tr>
<td>FileName</td>
<td>The filename of this database, if it has one. (Inherited from <code>BaseDatabase</code>.)</td>
</tr>
<tr>
<td>FreeThreaded</td>
<td>If true, the object is free-threaded; that is, concurrently usable by multiple threads in the address space. (Inherited from <code>BaseDatabase</code>.)</td>
</tr>
<tr>
<td>HasMultiple</td>
<td>If true, the object references a physical file supporting multiple databases. (Inherited from <code>BaseDatabase</code>.)</td>
</tr>
<tr>
<td>InHostOrder</td>
<td>If true, the underlying database files were created on an architecture of the same byte order as the current one. This information may be used to determine whether application data needs to be adjusted for this architecture or not. (Inherited from <code>BaseDatabase</code>.)</td>
</tr>
<tr>
<td>KeyGen</td>
<td>The delegate that creates the set of secondary keys corresponding to a given primary key and data pair. (Inherited from <code>SecondaryDatabase</code>.)</td>
</tr>
<tr>
<td>Length</td>
<td>If true, this database is not mapped into process memory. If using fixed-length, not byte-delimited records, the length of the records.</td>
</tr>
<tr>
<td>NoMMap</td>
<td>See <code>MMapSize</code> for further information. (Inherited from <code>BaseDatabase</code>.)</td>
</tr>
<tr>
<td>NonDurableTxns</td>
<td>If true, Berkeley DB will not write log records for this database. (Inherited from <code>BaseDatabase</code>.)</td>
</tr>
<tr>
<td>Nullifier</td>
<td>(Inherited from <code>SecondaryDatabase</code>.)</td>
</tr>
<tr>
<td>PadByte</td>
<td>The padding character for short, fixed-length records.</td>
</tr>
<tr>
<td>Pagesize</td>
<td>The database's current page size. (Inherited from <code>BaseDatabase</code>.)</td>
</tr>
</tbody>
</table>
**Priority**
The cache priority for pages referenced by this object.
(Inherited from BaseDatabase.)
If true, this database has been opened for reading only. Any attempt to modify items in the database will fail, regardless of the actual permissions of any underlying files.
(Inherited from BaseDatabase.)

**ReadOnly**
If true, this database has been opened for reading only. Any attempt to modify items in the database will fail, regardless of the actual permissions of any underlying files.
(Inherited from BaseDatabase.)
If true, this database supports transactional read operations with degree 1 isolation. Read operations on the database may request the return of modified but not yet committed data.
(Inherited from BaseDatabase.)

**ReadUncommitted**
If true, the logical record numbers are mutable, and change as records are added to and deleted from the database.
If true, any BackingFile file will be read in its entirety when Open(String, SecondaryRecnoDatabaseConfig) is called. If false, BackingFile may be read lazily.

**Renumber**
If true, any BackingFile file will be read in its entirety when Open(String, SecondaryRecnoDatabaseConfig) is called. If false, BackingFile may be read lazily.
If true, the logical record numbers are mutable, and change as records are added to and deleted from the database.

**Snapshot**
If true, this database has been opened in a transactional mode.
(Inherited from BaseDatabase.)
If true, the underlying file was physically truncated upon open, discarding all previous databases it might have held.
(Inherited from BaseDatabase.)

**Transactional**
The type of the underlying access method (and file format). This value may be used to determine the type of the database after an Open(String, DatabaseConfig).
(Inherited from BaseDatabase.)

**Truncated**
If true, the database was opened with support for multiversion concurrency control.
(Inherited from BaseDatabase.)
See Also

SecondaryRecnoDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryRecnoDatabase...:::BackingFile Property

The underlying source file for the Recno access method.

Namespace: BerkeleyDB
### Syntax

#### C#

```csharp
public string BackingFile { get; }
```

#### Visual Basic (Declaration)

```vbnet
Public ReadOnly Property BackingFile As String
```

#### Visual C++

```cpp
public:
property String^ BackingFile {
    String^ get ();
}
```
See Also

SecondaryRecnoDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The delimiting byte used to mark the end of a record in `BackingFile`.

**Namespace:**  BerkeleyDB  
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll)  Version: 4.8.24.0
Syntax

C#

public int Delimiter { get; }

Visual Basic (Declaration)

Public ReadOnly Property Delimiter As Integer

Visual C++

public:
    property int Delimiter {
        int get ();
    }
}
See Also

SecondaryRecnoDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If using fixed-length, not byte-delimited records, the length of the records.

**Namespace:** BerkeleyDB
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public uint Length { get; }

Visual Basic (Declaration)

Public ReadOnly Property Length AsUInteger

Visual C++

public:
property unsigned int Length {
    unsigned int get ();
}

See Also

SecondaryRecnoDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The padding character for short, fixed-length records.

**Namespace:** BerkeleyDB
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

```csharp
public int PadByte { get; }
```

Visual Basic (Declaration)

```vbnet
Public ReadOnly Property PadByte As Integer
```

Visual C++

```cpp
public:
property int PadByte {
    int get ();
}
```
See Also

SecondaryRecordDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, the logical record numbers are mutable, and change as records are added to and deleted from the database.

**Namespace:** [BerkeleyDB](#)

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool Renumber { get; }

Visual Basic (Declaration)

Public ReadOnly Property Renumber As Boolean

Visual C++

public:
property bool Renumber {
    bool get ();
}
}
See Also

SecondaryRecnoDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryRecnoDatabase::Snapshot Property

SecondaryRecnoDatabase Class  See Also

If true, any BackingFile file will be read in its entirety when Open(String, SecondaryRecnoDatabaseConfig) is called. If false, BackingFile may be read lazily.

Namespace: BerkeleyDB
Syntax

C#

public bool Snapshot { get; }

Visual Basic (Declaration)

Public Readonly Property Snapshot As Boolean

Visual C++

public:
property bool Snapshot {
    bool get ();
}

}
See Also

SecondaryRecnoDatabase Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryRecnoDatabaseConfig Class

A class representing configuration parameters for RecnoDatabase

Namespace: BerkeleyDB
Syntax

C#

```csharp
public class SecondaryRecnoDatabaseConfig : SecondaryDatabaseConfig
```

Visual Basic (Declaration)

```vbnet
Public Class SecondaryRecnoDatabaseConfig
    Inherits SecondaryDatabaseConfig
```

Visual C++

```cpp
public ref class SecondaryRecnoDatabaseConfig : public SecondaryDatabaseConfig
```
Inheritance Hierarchy

System:::Object
BerkeleyDB:::DatabaseConfig
   BerkeleyDB:::SecondaryDatabaseConfig
      BerkeleyDB:::SecondaryRecnoDatabaseConfig
See Also

SecondaryRecnoDatabaseConfig Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The **SecondaryRecnoDatabaseConfig** type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>SecondaryRecnoDatabaseConfig</code></td>
<td>Instantiate a new <code>SecondaryRecnoDatabaseConfig</code> object</td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td>SetEncryption</td>
<td>Set the password and algorithm used by the Berkeley DB library to perform encryption and decryption. (Inherited from <a href="#">DatabaseConfig</a>.)</td>
</tr>
<tr>
<td>SetForeignKeyConstraint</td>
<td>Overloaded.  Returns a <a href="#">String</a> that represents the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td>ToString</td>
<td></td>
</tr>
</tbody>
</table>
### Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AutoCommit</strong></td>
<td>Enclose the open call within a transaction. If the call succeeds, the open operation will be recoverable and all subsequent database modification operations based on this handle will be transactionally protected. If the call fails, no database will have been created. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td><strong>BackingFile</strong></td>
<td>The underlying source file for the Recno access method.</td>
</tr>
<tr>
<td><strong>ByteOrder</strong></td>
<td>The byte order for integers in the stored database metadata. The host byte order of the machine where the Berkeley DB library was compiled is the default value. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td><strong>CacheSize</strong></td>
<td>The size of the shared memory buffer pool -- that is, the cache. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td><strong>Creation</strong></td>
<td>The policy for how to handle database creation.</td>
</tr>
<tr>
<td><strong>DoChecksum</strong></td>
<td>If true, do checksum verification of pages read into the cache from the backing filestore. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td><strong>Env</strong></td>
<td>The Berkeley DB environment within which to create a database. If null, the database will be created stand-alone; that is, it is not part of any Berkeley DB environment. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td><strong>ErrorFeedback</strong></td>
<td>The mechanism for reporting error messages to the application. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td><strong>ErrorPrefix</strong></td>
<td>The prefix string that appears before error messages issued by Berkeley DB. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Feedback</strong></td>
<td>(Inherited from <a href="#">DatabaseConfig</a>) Cause the database object to be free-threaded; that is, concurrently usable by multiple threads in the address space. (Inherited from <a href="#">DatabaseConfig</a>)</td>
</tr>
<tr>
<td><strong>FreeThreaded</strong></td>
<td>If true, the secondary key is immutable. (Inherited from <a href="#">SecondaryDatabaseConfig</a>)</td>
</tr>
<tr>
<td><strong>ImmutableKey</strong></td>
<td>Do not map this database into process memory. (Inherited from <a href="#">DatabaseConfig</a>)</td>
</tr>
<tr>
<td><strong>NoMMap</strong></td>
<td>If true, Berkeley DB will not write log records for this database. (Inherited from <a href="#">DatabaseConfig</a>)</td>
</tr>
<tr>
<td><strong>NonDurableTxns</strong></td>
<td>If true and the secondary database is empty, walk through Primary and create an index to it in the empty secondary. This operation is potentially very expensive. (Inherited from <a href="#">SecondaryDatabaseConfig</a>)</td>
</tr>
<tr>
<td><strong>Populate</strong></td>
<td>The cache priority for pages referenced by the database. (Inherited from <a href="#">DatabaseConfig</a>)</td>
</tr>
<tr>
<td><strong>Priority</strong></td>
<td>Open the database for reading only. Any attempt to modify items in the database will fail, regardless of the actual permissions of any underlying files. (Inherited from <a href="#">DatabaseConfig</a>)</td>
</tr>
<tr>
<td><strong>ReadOnly</strong></td>
<td>Support transactional read operations with degree 1 isolation. (Inherited from <a href="#">DatabaseConfig</a>)</td>
</tr>
<tr>
<td><strong>ReadUncommitted</strong></td>
<td>Cause the logical record numbers to be mutable, and change as records are added to and deleted from the database.</td>
</tr>
<tr>
<td><strong>Renumber</strong></td>
<td>If true, any <a href="#">BackingFile</a> file will be read in its entirety when <a href="#">Open(String, SecondaryRecnoDatabaseConfig)</a> is called. If false, <a href="#">BackingFile</a> may be read lazily.</td>
</tr>
<tr>
<td><strong>Snapshot</strong></td>
<td>Physically truncate the underlying file, discarding all previous databases it might have held. (Inherited from <a href="#">DatabaseConfig</a>)</td>
</tr>
</tbody>
</table>
UseMVCC

Open the database with support for multiversion concurrency control.
(Inherited from DatabaseConfig.)
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Delimiter</strong></td>
<td>The delimiting byte used to mark the end of a record in <a href="#">BackingFile</a>. The algorithm used to perform encryption and decryption. (Inherited from <a href="#">DatabaseConfig</a>.)</td>
</tr>
<tr>
<td><strong>EncryptAlgorithm</strong></td>
<td></td>
</tr>
<tr>
<td><strong>EncryptionPassword</strong></td>
<td>The password used to perform encryption and decryption. (Inherited from <a href="#">DatabaseConfig</a>.)</td>
</tr>
<tr>
<td><strong>ForeignKeyDatabase</strong></td>
<td>(Inherited from <a href="#">SecondaryDatabaseConfig</a>.)</td>
</tr>
<tr>
<td><strong>ForeignKeyNullfier</strong></td>
<td>(Inherited from <a href="#">SecondaryDatabaseConfig</a>.)</td>
</tr>
<tr>
<td><strong>KeyGen</strong></td>
<td>The delegate that creates the set of secondary keys corresponding to a given primary key and data pair. (Inherited from <a href="#">SecondaryDatabaseConfig</a>.)</td>
</tr>
<tr>
<td><strong>Length</strong></td>
<td>Specify that the records are fixed-length, not byte-delimited, and are of length Length.</td>
</tr>
<tr>
<td><strong>OnForeignKeyDelete</strong></td>
<td>(Inherited from <a href="#">SecondaryDatabaseConfig</a>.)</td>
</tr>
<tr>
<td><strong>PadByte</strong></td>
<td>The padding character for short, fixed-length records.</td>
</tr>
<tr>
<td><strong>PageSize</strong></td>
<td>The size of the pages used to hold items in the database, in bytes. (Inherited from <a href="#">DatabaseConfig</a>.)</td>
</tr>
<tr>
<td><strong>Primary</strong></td>
<td>All updates to Primary will be automatically reflected in the secondary and all reads from the secondary will return corresponding data from Primary. (Inherited from <a href="#">SecondaryDatabaseConfig</a>.)</td>
</tr>
</tbody>
</table>
See Also

**SecondaryRecnoDatabaseConfig Class**

**BerkeleyDB Namespace**

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Instantiate a new SecondaryRecnoDatabaseConfig object

**Namespace:**  [BerkeleyDB](http://example.com/BerkeleyDB)

**Assembly:**  `libdb_dotnet48` (in `libdb_dotnet48.dll`) Version: 4.8.24.0
Syntax

C#

public SecondaryRecnoDatabaseConfig(
    Database PrimaryDB,
    SecondaryKeyGenDelegate KeyGenFunc
)

Visual Basic (Declaration)

Public Sub New ( _
    PrimaryDB As Database, _
    KeyGenFunc As SecondaryKeyGenDelegate _
)

Visual C++

public:
SecondaryRecnoDatabaseConfig(
    Database^ PrimaryDB,
    SecondaryKeyGenDelegate^ KeyGenFunc
)

Parameters

PrimaryDB
Type: BerkeleyDB::Database

KeyGenFunc
Type: BerkeleyDB::SecondaryKeyGenDelegate
See Also

SecondaryRecnoDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The SecondaryRecnoDatabaseConfig type exposes the following members.
## Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AutoCommit</strong></td>
<td>Enclose the open call within a transaction. If the call succeeds, the open operation will be recoverable and all subsequent database modification operations based on this handle will be transactionally protected. If the call fails, no database will have been created. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td><strong>BackingFile</strong></td>
<td>The underlying source file for the Recno access method.</td>
</tr>
<tr>
<td><strong>ByteOrder</strong></td>
<td>The byte order for integers in the stored database metadata. The host byte order of the machine where the Berkeley DB library was compiled is the default value. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td><strong>CacheSize</strong></td>
<td>The size of the shared memory buffer pool -- that is, the cache. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td><strong>Creation</strong></td>
<td>The policy for how to handle database creation. If true, do checksum verification of pages read into the cache from the backing filestore. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td><strong>DoChecksum</strong></td>
<td>The Berkeley DB environment within which to create a database. If null, the database will be created stand-alone; that is, it is not part of any Berkeley DB environment. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td><strong>Env</strong></td>
<td>The mechanism for reporting error messages to the application. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td><strong>ErrorFeedback</strong></td>
<td>The prefix string that appears before error messages issued by Berkeley DB. (Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td>Feature</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Feedback</strong></td>
<td>Cause the database object to be free-threaded; that is, concurrently usable by multiple threads in the address space.</td>
</tr>
<tr>
<td><strong>FreeThreaded</strong></td>
<td>(Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td><strong>ImmutableKey</strong></td>
<td>If true, the secondary key is immutable.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from SecondaryDatabaseConfig.)</td>
</tr>
<tr>
<td><strong>NoMMap</strong></td>
<td>Do not map this database into process memory.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td><strong>NonDurableTxns</strong></td>
<td>If true, Berkeley DB will not write log records for this database.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td><strong>Populate</strong></td>
<td>If true and the secondary database is empty, walk through Primary and create an index to it in the empty secondary. This operation is potentially very expensive.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from SecondaryDatabaseConfig.)</td>
</tr>
<tr>
<td><strong>Priority</strong></td>
<td>The cache priority for pages referenced by the database.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td><strong>ReadOnly</strong></td>
<td>Open the database for reading only. Any attempt to modify items in the database will fail, regardless of the actual permissions of any underlying files.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td><strong>ReadUncommitted</strong></td>
<td>Support transactional read operations with degree 1 isolation.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from DatabaseConfig.)</td>
</tr>
<tr>
<td><strong>Renumber</strong></td>
<td>Cause the logical record numbers to be mutable, and change as records are added to and deleted from the database.</td>
</tr>
<tr>
<td></td>
<td>If true, any BackingFile file will be read in its entirety when Open(String, SecondaryRecnoDatabaseConfig) is called. If false, BackingFile may be read lazily.</td>
</tr>
<tr>
<td><strong>Snapshot</strong></td>
<td>Physically truncate the underlying file, discarding all previous databases it might have held.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from DatabaseConfig.)</td>
</tr>
</tbody>
</table>
UseMVCC

Open the database with support for multiversion concurrency control.
(Inherited from DatabaseConfig.)
See Also

SecondaryRecnoDatabaseConfig Class  BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryRecnoDatabaseConfig Class  See Also

The underlying source file for the Recno access method.

Namespace: BerkeleyDB
Syntax

C#

public string BackingFile

Visual Basic (Declaration)

Public BackingFile As String

Visual C++

public:
String^ BackingFile
Remarks

The purpose of the source file is to provide fast access and modification to databases that are normally stored as flat text files.

The source parameter specifies an underlying flat text database file that is read to initialize a transient record number index. In the case of variable length records, the records are separated, as specified by Delimiter. For example, standard UNIX byte stream files can be interpreted as a sequence of variable length records separated by newline characters.

In addition, when cached data would normally be written back to the underlying database file (for example, CloseQQQ or SyncQQQ), the in-memory copy of the database will be written back to the source file.

By default, the backing source file is read lazily; that is, records are not read from the file until they are requested by the application. If multiple processes (not threads) are accessing a Recno database concurrently, and are either inserting or deleting records, the backing source file must be read in its entirety before more than a single process accesses the database, and only that process should specify the backing source file as part of the Open(String, SecondaryRecnoDatabaseConfig) call. See Snapshot for more information.

Reading and writing the backing source file specified by source cannot be transaction-protected because it involves filesystem operations that are not part of the Db transaction methodology. For this reason, if a temporary database is used to hold the records, it is possible to lose the contents of the source file, for example, if the system crashes at the right instant. If a file is used to hold the database, normal database recovery on that file can be used to prevent information loss, although it is still possible that the contents of source will be lost if the system crashes.

The source file must already exist (but may be zero-length) when Open(String, SecondaryRecnoDatabaseConfig) is called.

It is not an error to specify a read-only source file when creating a database, nor is it an error to modify the resulting database. However, any attempt to write the
changes to the backing source file using either the \texttt{Sync()} or \texttt{Close()} methods will fail, of course. Use \texttt{Close(Boolean)} to stop it from attempting to write the changes to the backing file; instead, they will be silently discarded.

For all of the previous reasons, the source file is generally used to specify databases that are read-only for Berkeley DB applications; and that are either generated on the fly by software tools or modified using a different mechanism — for example, a text editor.

If the database already exists, BackingFile must be the same as that historically used to create the database or corruption can occur.
See Also

SecondaryRecnoDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The policy for how to handle database creation.

**Namespace:** [BerkeleyDB](https://www.berkeley-db.com)

**Assembly:** [libdb_dotnet48](https://www.libdb.com) (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#
public CreatePolicy Creation

Visual Basic (Declaration)
Public Creation As CreatePolicy

Visual C++
public:
CreatePolicy Creation
Remarks

If the database does not already exist and NEVER is set, Open(String, SecondaryRecnoDatabaseConfig) will fail.
See Also

SecondaryRecnoDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryRecnoDatabaseConfig Class  See Also

Cause the logical record numbers to be mutable, and change as records are added to and deleted from the database.

Namespace: BerkeleyDB
Syntax

C#

public bool Renumber

Visual Basic (Declaration)

Public Renumber As Boolean

Visual C++

public:
bool Renumber
Remarks

For example, the deletion of record number 4 causes records numbered 5 and greater to be renumbered downward by one. If a cursor was positioned to record number 4 before the deletion, it will refer to the new record number 4, if any such record exists, after the deletion. If a cursor was positioned after record number 4 before the deletion, it will be shifted downward one logical record, continuing to refer to the same record as it did before.

Using `Put(DatabaseEntry, DatabaseEntry)` or `Put(DatabaseEntry, DatabaseEntry, UInt32)` to create new records will cause the creation of multiple records if the record number is more than one greater than the largest record currently in the database. For example, creating record 28, when record 25 was previously the last record in the database, will create records 26 and 27 as well as 28. Attempts to retrieve records that were created in this manner will throw a `KeyEmptyException`.

If a created record is not at the end of the database, all records following the new record will be automatically renumbered upward by one. For example, the creation of a new record numbered 8 causes records numbered 8 and greater to be renumbered upward by one. If a cursor was positioned to record number 8 or greater before the insertion, it will be shifted upward one logical record, continuing to refer to the same record as it did before.

For these reasons, concurrent access to a `SecondaryRecnoDatabase` with this setting specified may be largely meaningless, although it is supported.

If the database already exists, this setting must be the same as the existing database or an exception will be thrown.
See Also

SecondaryRecnoDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, any `BackingFile` file will be read in its entirety when `Open(String, SecondaryRecnoDatabaseConfig)` is called. If false, `BackingFile` may be read lazily.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll)  
**Version:** 4.8.24.0
Syntax

C#

public bool Snapshot

Visual Basic (Declaration)

Public Snapshot As Boolean

Visual C++

public:
bool Snapshot
See Also

SecondaryRecnoDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `SecondaryRecnoDatabaseConfig` type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>SetEncryption</strong></td>
<td>Set the password and algorithm used by the Berkeley DB library to perform encryption and decryption. (Inherited from <a href="#">DatabaseConfig</a>.)</td>
</tr>
<tr>
<td><strong>SetForeignKeyConstraint</strong></td>
<td>Overloaded.</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a <a href="#">String</a> that represents the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
</tbody>
</table>
See Also

SecondaryRecnoDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryRecnoDatabaseConfig::SetForeignKeyConstraint Method
SecondaryRecnoDatabaseConfig Class  See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>SetForeignKeyConstraint(Database, ForeignKeyDeleteAction)</code></td>
<td>(Inherited from <code>SecondaryDatabaseConfig</code>.)</td>
</tr>
<tr>
<td><code>SetForeignKeyConstraint(Database, ForeignKeyDeleteAction, ForeignKeyNullifyDelegate)</code></td>
<td>(Inherited from <code>SecondaryDatabaseConfig</code>.)</td>
</tr>
</tbody>
</table>
See Also

SecondaryRecnoDatabaseConfig Class
SecondaryRecnoDatabaseConfig Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `SecondaryRecnoDatabaseConfig` type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Delimiter</strong></td>
<td>The delimiting byte used to mark the end of a record in <a href="#">BackingFile</a>. The algorithm used to perform encryption and decryption. (Inherited from <a href="#">DatabaseConfig</a>.)</td>
</tr>
<tr>
<td><strong>EncryptAlgorithm</strong></td>
<td>(Inherited from <a href="#">DatabaseConfig</a>.) The algorithm used to perform encryption and decryption.</td>
</tr>
<tr>
<td><strong>EncryptionPassword</strong></td>
<td>The password used to perform encryption and decryption. (Inherited from <a href="#">DatabaseConfig</a>.)</td>
</tr>
<tr>
<td><strong>ForeignKeyDatabase</strong></td>
<td>(Inherited from <a href="#">SecondaryDatabaseConfig</a>.) The delegate that creates the set of secondary keys corresponding to a given primary key and data pair. (Inherited from <a href="#">SecondaryDatabaseConfig</a>.)</td>
</tr>
<tr>
<td><strong>ForeignKeyNullifier</strong></td>
<td>(Inherited from <a href="#">SecondaryDatabaseConfig</a>.)</td>
</tr>
<tr>
<td><strong>KeyGen</strong></td>
<td>The delegate that creates the set of secondary keys corresponding to a given primary key and data pair. (Inherited from <a href="#">SecondaryDatabaseConfig</a>.)</td>
</tr>
<tr>
<td><strong>Length</strong></td>
<td>Specify that the records are fixed-length, not byte-delimited, and are of length Length. (Inherited from <a href="#">SecondaryDatabaseConfig</a>.)</td>
</tr>
<tr>
<td><strong>OnForeignKeyDelete</strong></td>
<td>(Inherited from <a href="#">SecondaryDatabaseConfig</a>.) The padding character for short, fixed-length records.</td>
</tr>
<tr>
<td><strong>PadByte</strong></td>
<td>The padding character for short, fixed-length records.</td>
</tr>
<tr>
<td><strong>PageSize</strong></td>
<td>The size of the pages used to hold items in the database, in bytes. (Inherited from <a href="#">DatabaseConfig</a>.) All updates to Primary will be automatically reflected in the secondary and all reads from the secondary will return corresponding data from Primary. (Inherited from <a href="#">SecondaryDatabaseConfig</a>.)</td>
</tr>
</tbody>
</table>
See Also

SecondaryRecnoDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryRecnoDatabaseConfig::Delimiter Property

The delimiting byte used to mark the end of a record in BackingFile.

Namespace: BerkeleyDB
Syntax

C#

public int Delimiter { get; set; }

Visual Basic (Declaration)

Public Property Delimiter As Integer

Visual C++

public:
property int Delimiter {
    int get ();
    void set (int value);
}

Remarks

This byte is used for variable length records if `BackingFile` is set. If `BackingFile` is specified and no delimiting byte was specified, newline characters (that is, ASCII 0x0a) are interpreted as end-of-record markers.

If the database already exists, this setting will be ignored.
See Also

SecondaryRecnoDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Specify that the records are fixed-length, not byte-delimited, and are of length Length.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

**C#**

public uint Length { get; set; }

**Visual Basic (Declaration)**

Public Property Length AsUInteger

**Visual C++**

public:
property unsigned int Length {
    unsigned int get ();
    void set (unsigned int value);
}
Remarks

Any records added to the database that are less than Length bytes long are automatically padded (see PadByte for more information).

Any attempt to insert records into the database that are greater than Length bytes long will cause the call to fail immediately and return an error.

If the database already exists, this setting will be ignored.
See Also

SecondaryRecnoDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SecondaryRecnoDatabaseConfig Class  See Also

The padding character for short, fixed-length records.

**Namespace:** BerkeleyDB
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public int PadByte { get; set; }

Visual Basic (Declaration)

Public Property PadByte As Integer

Visual C++

public:
property int PadByte {
        int get ();
        void set (int value);
    }

Remarks

If no pad character is specified, space characters (that is, ASCII 0x20) are used for padding.

If the database already exists, this setting will be ignored.
See Also

SecondaryRecnoDatabaseConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
A class that provides an arbitrary number of persistent objects that return an increasing or decreasing sequence of integers.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public class Sequence : IDisposable

Visual Basic (Declaration)

Public Class Sequence  Implements IDisposable

Visual C++

public ref class Sequence : IDisposable
Inheritance Hierarchy

System..:::Object
BerkeleyDB..:::Sequence
See Also

Sequence Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `Sequence` type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequence</td>
<td>Overloaded.</td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Close</td>
<td>Close the sequence handle. Any unused cached values are lost. Release the resources held by this object, and close the sequence if it's still open.</td>
</tr>
<tr>
<td>Dispose</td>
<td>Determines whether the specified Object is equal to the current Object. (Inherited from Object.) Overloaded.</td>
</tr>
<tr>
<td>Equals</td>
<td>If there are enough cached values in the sequence handle then they will be returned. Otherwise the next value will be fetched from the database and incremented (decremented) by enough to cover the delta and the next batch of cached values. For maximum concurrency a non-zero cache size should be specified prior to opening the sequence handle and NoSync should be specified for each Get method call. By default, sequence ranges do not wrap; to cause the sequence to wrap around the beginning or end of its range, set SequenceConfig.Wrap to true.</td>
</tr>
<tr>
<td>Get</td>
<td>If P:BackingDatabase was opened in a transaction, calling Get may result in changes to the sequence object; these changes will be automatically committed in a transaction internal to the Berkeley DB library. If the thread of control calling Get has an active transaction, which holds locks on the same database as the one in which the sequence object is stored, it is possible for a thread of control calling Get to self-deadlock because the active transaction's locks conflict with the internal transaction's locks. For this reason, it is often preferable</td>
</tr>
</tbody>
</table>
for sequence objects to be stored in their own database.

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetHashCode</td>
<td>Serves as a hash function for a particular type. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>PrintStats</td>
<td>The diagnostic information is described by SequenceStats.</td>
</tr>
<tr>
<td>Remove</td>
<td>Overloaded.</td>
</tr>
<tr>
<td>Stats</td>
<td>The DB_SEQUENCE-&gt;stat() method cannot be transaction-protected. For this reason, it should be called in a thread of control that has no open cursors or active transactions.</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a String that represents the current Object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BackingDatabase</td>
<td>The database used by the sequence.</td>
</tr>
<tr>
<td>Cachesize</td>
<td>The current cache size.</td>
</tr>
<tr>
<td>Decrement</td>
<td>If true, the sequence will be decremented.</td>
</tr>
<tr>
<td>Increment</td>
<td>If true, the sequence will be incremented. This is the default.</td>
</tr>
<tr>
<td>Key</td>
<td>The key for the sequence.</td>
</tr>
<tr>
<td>Max</td>
<td>The maximum value in the sequence.</td>
</tr>
<tr>
<td>Min</td>
<td>The minimum value in the sequence.</td>
</tr>
<tr>
<td>Wrap</td>
<td>If true, the sequence should wrap around when it is incremented (decremented) past the specified maximum (minimum) value.</td>
</tr>
</tbody>
</table>
See Also

Sequence Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Sequence Constructor

Sequence Class  See Also
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Sequence(SequenceConfig)</code></td>
<td>Instantiate a new Sequence object.</td>
</tr>
<tr>
<td><code>Sequence(SequenceConfig, Transaction)</code></td>
<td>Instantiate a new Sequence object.</td>
</tr>
</tbody>
</table>
See Also

Sequence Class
Sequence Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Instantiate a new Sequence object.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public Sequence(
    SequenceConfig cfg
)

Visual Basic (Declaration)

Public Sub New (  
    cfg As SequenceConfig  
)

Visual C++

public:
    Sequence(
        SequenceConfig^ cfg
    )

Parameters

cfg
    Type: BerkeleyDB::::SequenceConfig
    Configuration parameters for the Sequence
If txn is null and the operation occurs in a transactional database, the operation will be implicitly transaction protected.
See Also

Sequence Class
Sequence Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation
Sequence Constructor (SequenceConfig, Transaction)

**Sequence Class**  **See Also**

Instantiate a new Sequence object.

**Namespace:**  BerkeleyDB
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

```csharp
public Sequence(
    SequenceConfig cfg,
    Transaction txn
)
```

Visual Basic (Declaration)

```vbnet
Public Sub New (_
    cfg As SequenceConfig, _
    txn As Transaction _
)
```

Visual C++

```cpp
public:
    Sequence(
        SequenceConfig^ cfg,
        Transaction^ txn
    )
```

Parameters

cfg

Type: BerkeleyDB::::SequenceConfig
Configuration parameters for the Sequence

txn

Type: BerkeleyDB::::Transaction
If the operation is part of an application-specified transaction, txn is a Transaction object returned from BeginTransaction(); if the operation is part of a Berkeley DB Concurrent Data Store group, txn is a handle returned from BeginCDSGroup(); otherwise null.
Remarks

If txn is null and the operation occurs in a transactional database, the operation will be implicitly transaction protected.
See Also

Sequence Class
Sequence Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The **Sequence** type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Close</td>
<td>Close the sequence handle. Any unused cached values are lost.</td>
</tr>
<tr>
<td>Dispose</td>
<td>Release the resources held by this object, and close the sequence if it's still open.</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.) Overloaded.</td>
</tr>
<tr>
<td>Get</td>
<td>If there are enough cached values in the sequence handle then they will be returned. Otherwise the next value will be fetched from the database and incremented (decremented) by enough to cover the delta and the next batch of cached values.</td>
</tr>
<tr>
<td></td>
<td>For maximum concurrency a non-zero cache size should be specified prior to opening the sequence handle and NoSync should be specified for each Get method call.</td>
</tr>
<tr>
<td></td>
<td>By default, sequence ranges do not wrap; to cause the sequence to wrap around the beginning or end of its range, set SequenceConfig.Wrap to true.</td>
</tr>
<tr>
<td></td>
<td>If P:BackingDatabase was opened in a transaction, calling Get may result in changes to the sequence object; these changes will be automatically committed in a transaction internal to the Berkeley DB library. If the thread of control calling Get has an active transaction, which holds locks on the same database as the one in which the sequence object is stored, it is possible for a thread of control calling Get to self-deadlock because the active transaction's locks conflict with the internal transaction's locks. For this reason, it is often preferable</td>
</tr>
</tbody>
</table>
for sequence objects to be stored in their own database.

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetHashCode</td>
<td>Serves as a hash function for a particular type.</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>PrintStats</td>
<td>The diagnostic information is described by SequenceStats.</td>
</tr>
<tr>
<td>Remove</td>
<td>Overloaded.</td>
</tr>
<tr>
<td></td>
<td>Overloaded.</td>
</tr>
<tr>
<td>Stats</td>
<td>In the presence of multiple threads or processes accessing an active sequence, the information returned by DB_SEQUENCE-&gt;stat() may be out-of-date.</td>
</tr>
<tr>
<td></td>
<td>The DB_SEQUENCE-&gt;stat() method cannot be transaction-protected. For this reason, it should be called in a thread of control that has no open cursors or active transactions.</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a String that represents the current Object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>
See Also

Sequence Class  
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Sequence...:::Close Method

See Also

Close the sequence handle. Any unused cached values are lost.

Namespace: BerkeleyDB
Syntax

**C#**

```csharp
public void Close()
```

**Visual Basic (Declaration)**

```vbnet
Public Sub Close
```

**Visual C++**

```cpp
public:
void Close()
```
See Also

Sequence Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Release the resources held by this object, and close the sequence if it's still open.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public void Dispose()

Visual Basic (Declaration)

Public Sub Dispose

Visual C++

public:
virtual void Dispose() sealed

Implements

IDisposable:::Dispose()()
See Also

Sequence Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If there are enough cached values in the sequence handle then they will be returned. Otherwise the next value will be fetched from the database and incremented (decremented) by enough to cover the delta and the next batch of cached values.

For maximum concurrency a non-zero cache size should be specified prior to opening the sequence handle and NoSync should be specified for each Get method call.

By default, sequence ranges do not wrap; to cause the sequence to wrap around the beginning or end of its range, set SequenceConfig.Wrap to true.

If P:BackingDatabase was opened in a transaction, calling Get may result in changes to the sequence object; these changes will be automatically committed in a transaction internal to the Berkeley DB library. If the thread of control calling Get has an active transaction, which holds locks on the same database as the one in which the sequence object is stored, it is possible for a thread of control calling Get to self-deadlock because the active transaction's locks conflict with the internal transaction's locks. For this reason, it is often preferable for sequence objects to be stored in their own database.
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Get(Int32)</td>
<td>Return the next available element in the sequence and change the sequence value by Delta.</td>
</tr>
<tr>
<td>Get(Int32, Transaction)</td>
<td>Return the next available element in the sequence and change the sequence value by Delta.</td>
</tr>
<tr>
<td>Get(Int32, Boolean)</td>
<td>Return the next available element in the sequence and change the sequence value by Delta.</td>
</tr>
</tbody>
</table>
See Also

Sequence Class
Sequence Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Return the next available element in the sequence and change the sequence value by Delta.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
### Syntax

**C#**

```csharp
public long Get(
    int Delta
)
```

**Visual Basic (Declaration)**

```vbnet
Public Function Get ( _
    Delta As Integer _
) As Long
```

**Visual C++**

```cpp
public:
    long long Get(
        int Delta
    )
```

### Parameters

**Delta**
- **Type:** `System::::Int32`
  - The amount by which to increment the sequence value. Must be greater than 0.

### Return Value

The next available element in the sequence.
See Also

Sequence Class
Get Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Return the next available element in the sequence and change the sequence value by Delta.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
## Syntax

### C#

```csharp
public long Get(
    int Delta,
    Transaction txn
)
```

### Visual Basic (Declaration)

```vbnet
Public Function Get ( _
    Delta As Integer, _
    txn As Transaction _
) As Long
```

### Visual C++

```cpp
public:
    long long Get(
        int Delta,
        Transaction^ txn
    )
```

## Parameters

**Delta**

Type: `System::::Int32`

The amount by which to increment the sequence value. Must be greater than 0.

**txn**

Type: `BerkeleyDB::::Transaction`

If the operation is part of an application-specified transaction, txn is a Transaction object returned from `BeginTransaction()();` if the operation is part of a Berkeley DB Concurrent Data Store group, txn is a handle returned from `BeginCDSGroup()();` otherwise null. Must be null if the sequence was opened with a non-zero cache size.
**Return Value**

The next available element in the sequence.
See Also

Sequence Class
Get Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Return the next available element in the sequence and change the sequence value by Delta.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public long Get(
        int Delta,
        bool NoSync
    )

Visual Basic (Declaration)

Public Function Get ( _
        Delta As Integer, _
        NoSync As Boolean _
    ) As Long

Visual C++

public:
    long long Get(
            int Delta,
            bool NoSync
    )

Parameters

Delta
Type: System::::Int32
The amount by which to increment the sequence value. Must be greater than 0.

NoSync
Type: System::::Boolean
If true, and if the operation is implicitly transaction protected, do not synchronously flush the log when the transaction commits.

Return Value
The next available element in the sequence.
See Also

Sequence Class
Get Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The diagnostic information is described by SequenceStats.
### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PrintStats()</td>
<td>Print diagnostic information.</td>
</tr>
<tr>
<td>PrintStats(Boolean)</td>
<td>Print diagnostic information.</td>
</tr>
</tbody>
</table>
See Also

Sequence Class
Sequence Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Print diagnostic information.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

**C#**

public void PrintStats()

**Visual Basic (Declaration)**

Public Sub PrintStats

**Visual C++**

public:
void PrintStats()
See Also

Sequence Class
PrintStats Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Print diagnostic information.

**Namespace:** [BerkeleyDB](#)

**Assembly:** `libdb_dotnet48` (in `libdb_dotnet48.dll`) Version: 4.8.24.0
Syntax

C#

public void PrintStats(  
    bool ClearStats
)

Visual Basic (Declaration)

Public Sub PrintStats (  
    ClearStats As Boolean  
)

Visual C++

public:
void PrintStats(  
    bool ClearStats
)

Parameters

ClearStats
    Type: System:: Boolean
    If true, reset statistics after printing.
See Also

Sequence Class
PrintStats Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
C#  Visual Basic  Visual C++
Include Protected Members  Include Inherited Members
Berkeley DB .NET API Documentation
Sequence...:.Remove Method
**Sequence Class**  **See Also**
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Remove()()</td>
<td>Remove the sequence from the database.</td>
</tr>
<tr>
<td>Remove(Transaction)</td>
<td>Remove the sequence from the database.</td>
</tr>
<tr>
<td>Remove(Boolean)</td>
<td>Remove the sequence from the database.</td>
</tr>
</tbody>
</table>
See Also

Sequence Class
Sequence Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Remove the sequence from the database.

Namespace:   BerkeleyDB
Syntax

C#
public void Remove()

Visual Basic (Declaration)
Public Sub Remove

Visual C++
public:
void Remove()
See Also

Sequence Class
Remove Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Remove the sequence from the database.

**Namespace:**  BerkeleyDB  
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

```csharp
public void Remove(
    Transaction txn
)
```

Visual Basic (Declaration)

```vbnet
Public Sub Remove (
    _
    txn As Transaction _
)
```

Visual C++

```cpp
public:
void Remove(
    Transaction^ txn
)
```

Parameters

txn

Type: BerkeleyDB::Transaction

If the operation is part of an application-specified transaction, txn is a Transaction object returned from BeginTransaction(); if the operation is part of a Berkeley DB Concurrent Data Store group, txn is a handle returned from BeginCDSGroup(); otherwise null.
See Also

Sequence Class
Remove Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Remove the sequence from the database.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
## Syntax

### C#

```csharp
public void Remove(
    bool NoSync
)
```

### Visual Basic (Declaration)

```vbnet
Public Sub Remove (_,
    NoSync As Boolean _
)
```

### Visual C++

```cpp
public:
void Remove(
    bool NoSync
)
```

## Parameters

NoSync
Type: `System::::Boolean`
If true, and if the operation is implicitly transaction protected, do not synchronously flush the log when the transaction commits.
See Also

Sequence Class
Remove Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
In the presence of multiple threads or processes accessing an active sequence, the information returned by DB_SEQUENCE->stat() may be out-of-date.

The DB_SEQUENCE->stat() method cannot be transaction-protected. For this reason, it should be called in a thread of control that has no open cursors or active transactions.
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stats()()</td>
<td>Return statistical information for this sequence.</td>
</tr>
<tr>
<td>Stats(Boolean)</td>
<td>Return statistical information for this sequence.</td>
</tr>
</tbody>
</table>


See Also

Sequence Class
Sequence Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Return statistical information for this sequence.

**Namespace:** [BerkeleyDB](https://berkeleydb.com)

**Assembly:** `libdb_dotnet48` (in `libdb_dotnet48.dll`) Version: 4.8.24.0
Syntax

C#

public SequenceStats Stats()

Visual Basic (Declaration)

Public Function Stats As SequenceStats

Visual C++

public:
SequenceStats^ Stats()

Return Value

Statistical information for this sequence.
See Also

Sequence Class
Stats Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Return statistical information for this sequence.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public SequenceStats Stats(
    bool clear
)

Visual Basic (Declaration)

Public Function Stats ( _
    clear As Boolean _
) As SequenceStats

Visual C++

public:
    SequenceStats^ Stats(
    bool clear
)

Parameters

clear
    Type: System::::Boolean
    If true, reset statistics.

Return Value

Statistical information for this sequence.
See Also

Sequence Class
Stats Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `Sequence` type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BackingDatabase</td>
<td>The database used by the sequence.</td>
</tr>
<tr>
<td>Cachesize</td>
<td>The current cache size.</td>
</tr>
<tr>
<td>Decrement</td>
<td>If true, the sequence will be decremented.</td>
</tr>
<tr>
<td>Increment</td>
<td>If true, the sequence will be incremented. This is the default.</td>
</tr>
<tr>
<td>Key</td>
<td>The key for the sequence.</td>
</tr>
<tr>
<td>Max</td>
<td>The maximum value in the sequence.</td>
</tr>
<tr>
<td>Min</td>
<td>The minimum value in the sequence.</td>
</tr>
<tr>
<td>Wrap</td>
<td>If true, the sequence should wrap around when it is incremented (decremented) past the specified maximum (minimum) value.</td>
</tr>
</tbody>
</table>
See Also

Sequence Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The database used by the sequence.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public Database BackingDatabase { get; }

Visual Basic (Declaration)

Public ReadOnly Property BackingDatabase As Database

Visual C++

public:
property Database^ BackingDatabase {
    Database^ get ();
}
See Also

Sequence Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The current cache size.

**Namespace:**  [BerkeleyDB](https://berkley-db.org)
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
### Syntax

**C#**

```csharp
public int Cachesize { get; }
```

**Visual Basic (Declaration)**

```vbnet
Public ReadOnly Property Cachesize As Integer
```

**Visual C++**

```cpp
public:
    property int Cachesize { 
        int get ();
    }
```
See Also

Sequence Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, the sequence will be decremented.

**Namespace:**  BerkeleyDB

**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool Decrement { get; }

Visual Basic (Declaration)

Public ReadOnly Property Decrement As Boolean

Visual C++

public:
property bool Decrement {
    bool get ();
}

See Also

Sequence Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, the sequence will be incremented. This is the default.

Namespace: BerkeleyDB
Syntax

C#

public bool Increment { get; }

Visual Basic (Declaration)

Public ReadOnly Property Increment As Boolean

Visual C++

public:
property bool Increment {
    bool get ();
}
}
See Also

Sequence Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The key for the sequence.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public DatabaseEntry Key { get; }

Visual Basic (Declaration)

Public ReadOnly Property Key As DatabaseEntry

Visual C++

public:
property DatabaseEntry^ Key {
    DatabaseEntry^ get ();
}

See Also

Sequence Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The maximum value in the sequence.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public long Max { get; }

Visual Basic (Declaration)

Public ReadOnly Property Max As Long

Visual C++

public:
property long long Max {
    long long get ();
}
See Also

Sequence Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The minimum value in the sequence.

**Namespace:** [BerkeleyDB](https://www.oracle.com/database/database-technologies/berkeley-db-net)

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public  long  Min  {  get;  }

Visual Basic (Declaration)

Public ReadOnly Property Min As  Long

Visual C++

public:
property  long long  Min  {
    long long  get  ();
}


See Also

Sequence Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, the sequence should wrap around when it is incremented (decremented) past the specified maximum (minimum) value.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool Wrap { get; }

Visual Basic (Declaration)

Public ReadOnly Property Wrap As Boolean

Visual C++

public:
  property bool Wrap {
    bool get ();
  }
}
See Also

Sequence Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SequenceConfig Class

See Also

Configuration properties for a Sequence

Namespace: BerkeleyDB
Syntax

C#

public class SequenceConfig

Visual Basic (Declaration)

Public Class SequenceConfig

Visual C++

public ref class SequenceConfig
Inheritance Hierarchy

System..:::Object
BerkeleyDB..:::SequenceConfig
See Also

SequenceConfig Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `SequenceConfig` type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SequenceConfig</td>
<td>Initializes a new instance of the <a href="#">SequenceConfig</a> class</td>
</tr>
</tbody>
</table>
# Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a> is equal to the current <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the <a href="https://docs.microsoft.com/en-us/dotnet/api/system.type">Type</a> of the current instance. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.)</td>
</tr>
<tr>
<td>SetRange</td>
<td>Set the minimum and maximum values in the sequence.</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a <a href="https://docs.microsoft.com/en-us/dotnet/api/system.string">String</a> that represents the current <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.</td>
</tr>
</tbody>
</table>
## Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BackingDatabase</td>
<td>An open database which holds the persistent data for the sequence.</td>
</tr>
<tr>
<td>Creation</td>
<td>The policy for how to handle sequence creation. If true, the object returned by the Sequence constructor will be free-threaded; that is, usable by multiple threads within a single address space. Note that if multiple threads create multiple sequences using the same BackingDatabase, that database must have also been opened free-threaded.</td>
</tr>
<tr>
<td>FreeThreaded</td>
<td>The record in the database that stores the persistent sequence data. If true, the sequence should wrap around when it is incremented (decremented) past the specified maximum (minimum) value.</td>
</tr>
<tr>
<td>key</td>
<td></td>
</tr>
<tr>
<td>Wrap</td>
<td></td>
</tr>
</tbody>
</table>
### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CacheSize</td>
<td>The number of elements cached by a sequence handle.</td>
</tr>
<tr>
<td>Decrement</td>
<td>If true, the sequence will be decremented.</td>
</tr>
<tr>
<td>Increment</td>
<td>If true, the sequence will be incremented. This is the default.</td>
</tr>
<tr>
<td>InitialValue</td>
<td>The initial value for a sequence.</td>
</tr>
<tr>
<td>Max</td>
<td>The maximum value in the sequence.</td>
</tr>
<tr>
<td>Min</td>
<td>The minimum value in the sequence.</td>
</tr>
</tbody>
</table>
See Also

SequenceConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SequenceConfig Constructor

Initializes a new instance of the SequenceConfig class

Namespace: BerkeleyDB
Syntax

C#

public SequenceConfig()

Visual Basic (Declaration)

Public Sub New

Visual C++

public:
SequenceConfig()
See Also

SequenceConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The **SequenceConfig** type exposes the following members.
# Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BackingDatabase</td>
<td>An open database which holds the persistent data for the sequence.</td>
</tr>
<tr>
<td>Creation</td>
<td>The policy for how to handle sequence creation.</td>
</tr>
<tr>
<td></td>
<td>If true, the object returned by the Sequence constructor will be free-threaded; that is, usable by multiple threads within a single address space. Note that if multiple threads create multiple sequences using the same BackingDatabase, that database must have also been opened free-threaded.</td>
</tr>
<tr>
<td>FreeThreaded</td>
<td>The record in the database that stores the persistent sequence data.</td>
</tr>
<tr>
<td></td>
<td>If true, the sequence should wrap around when it is incremented (decremented) past the specified maximum (minimum) value.</td>
</tr>
<tr>
<td>key</td>
<td></td>
</tr>
<tr>
<td>Wrap</td>
<td></td>
</tr>
</tbody>
</table>
See Also

SequenceConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
An open database which holds the persistent data for the sequence.

**Namespace:**  BerkeleyDB
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

**C#**

public Database BackingDatabase

**Visual Basic (Declaration)**

Public BackingDatabase As Database

**Visual C++**

public:
Database BackingDatabase
Remarks

The database may be of any type, but must not have been configured to support duplicate data items.

If P:BackingDatabase was opened in a transaction, calling Get may result in changes to the sequence object; these changes will be automatically committed in a transaction internal to the Berkeley DB library. If the thread of control calling Get has an active transaction, which holds locks on the same database as the one in which the sequence object is stored, it is possible for a thread of control calling Get to self-deadlock because the active transaction's locks conflict with the internal transaction's locks. For this reason, it is often preferable for sequence objects to be stored in their own database.
See Also

SequenceConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The policy for how to handle sequence creation.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public CreatePolicy Creation

Visual Basic (Declaration)

Public Creation As CreatePolicy

Visual C++

public:
CreatePolicy Creation
Remarks

If the sequence does not already exist and NEVER is set, the Sequence constructor will fail.
See Also

SequenceConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, the object returned by the Sequence constructor will be free-threaded; that is, usable by multiple threads within a single address space. Note that if multiple threads create multiple sequences using the same BackingDatabase, that database must have also been opened free-threaded.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool FreeThreaded

Visual Basic (Declaration)

Public FreeThreaded As Boolean

Visual C++

public:
bool FreeThreaded
See Also

SequenceConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The record in the database that stores the persistent sequence data.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public DatabaseEntry key

Visual Basic (Declaration)

Public key As DatabaseEntry

Visual C++

public:
DatabaseEntry^ key
See Also

SequenceConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, the sequence should wrap around when it is incremented (decremented) past the specified maximum (minimum) value.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool Wrap

Visual Basic (Declaration)

Public Wrap As Boolean

Visual C++

public:
    bool Wrap
See Also

SequenceConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `SequenceConfig` type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>SetRange</strong></td>
<td>Set the minimum and maximum values in the sequence.</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a <a href="#">String</a> that represents the current <a href="#">Object</a>.              (Inherited from <a href="#">Object</a>.)</td>
</tr>
</tbody>
</table>
See Also

SequenceConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Set the minimum and maximum values in the sequence.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
### Syntax

#### C#

```csharp
public void SetRange(
    long Min,
    long Max
)
```

#### Visual Basic (Declaration)

```vbnet
Public Sub SetRange ( _
    Min As Long, _
    Max As Long _
)
```

#### Visual C++

```cpp
public:
void SetRange(
    long long Min,
    long long Max
)
```

### Parameters

**Min**
- Type: `System::::Int64`
- The minimum value in the sequence.

**Max**
- Type: `System::::Int64`
- The maximum value in the sequence.
See Also

SequenceConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `SequenceConfig` type exposes the following members.
### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CacheSize</td>
<td>The number of elements cached by a sequence handle.</td>
</tr>
<tr>
<td>Decrement</td>
<td>If true, the sequence will be decremented.</td>
</tr>
<tr>
<td>Increment</td>
<td>If true, the sequence will be incremented. This is the default.</td>
</tr>
<tr>
<td>InitialValue</td>
<td>The initial value for a sequence.</td>
</tr>
<tr>
<td>Max</td>
<td>The maximum value in the sequence.</td>
</tr>
<tr>
<td>Min</td>
<td>The minimum value in the sequence.</td>
</tr>
</tbody>
</table>
See Also

SequenceConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The number of elements cached by a sequence handle.

**Namespace:** [BerkeleyDB](https://github.com/BerkeleyDB)  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public int CacheSize { get; set; }

Visual Basic (Declaration)

Public Property CacheSize As Integer

Visual C++

public:
property int CacheSize {
    int get ();
    void set (int value);
}
See Also

SequenceConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, the sequence will be decremented.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool Decrement { get; set; }

Visual Basic (Declaration)

Public Property Decrement As Boolean

Visual C++

public:
property bool Decrement {
    bool get ();
    void set (bool value);
}
See Also

SequenceConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, the sequence will be incremented. This is the default.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool Increment { get; set; }

Visual Basic (Declaration)

Public Property Increment As Boolean

Visual C++

public:
property bool Increment {
bool get ();
void set (bool value);
}
See Also

SequenceConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SequenceConfig:::InitialValue Property

The initial value for a sequence.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public long InitialValue { get; set; }

Visual Basic (Declaration)

Public Property InitialValue As Long

Visual C++

public:
property long long InitialValue {
    long long get ();
    void set (long long value);
}
See Also

SequenceConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SequenceConfig.

Max Property

The maximum value in the sequence.

Namespace: BerkeleyDB
Syntax

C#

public long Max { get; }

Visual Basic (Declaration)

Public ReadOnly Property Max As Long

Visual C++

public:
property long long Max {
    long long get ();
}


See Also

SequenceConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The minimum value in the sequence.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public long Min { get; }

Visual Basic (Declaration)

Public ReadOnly Property Min As Long

Visual C++

public:
property long long Min {
    long long get ();
}

See Also

SequenceConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Statistical information about a Sequence

**Namespace:** BerkeleyDB
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#  
public class SequenceStats

Visual Basic (Declaration)  
Public Class SequenceStats  

Visual C++  
public ref class SequenceStats
Inheritance Hierarchy

System:::Object
BerkeleyDB:::SequenceStats
See Also

SequenceStats Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation

SequenceStats Members

SequenceStats Class Methods Properties See Also

The SequenceStats type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified <a href="/object">Object</a> is equal to the current <a href="/object">Object</a>. (Inherited from <a href="/object">Object</a>.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="/object">Object</a>.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the <a href="/type">Type</a> of the current instance. (Inherited from <a href="/object">Object</a>.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a <a href="/string">String</a> that represents the current <a href="/object">Object</a>. (Inherited from <a href="/object">Object</a>.)</td>
</tr>
</tbody>
</table>
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CachedValue</td>
<td>Current cached value.</td>
</tr>
<tr>
<td>CacheSize</td>
<td>Cache size.</td>
</tr>
<tr>
<td>Flags</td>
<td>Flag value.</td>
</tr>
<tr>
<td>LastCachedValue</td>
<td>Last cached value.</td>
</tr>
<tr>
<td>LockNoWait</td>
<td>Sequence lock granted after wait.</td>
</tr>
<tr>
<td>LockWait</td>
<td>Sequence lock granted w/o wait.</td>
</tr>
<tr>
<td>Max</td>
<td>Maximum value.</td>
</tr>
<tr>
<td>Min</td>
<td>Minimum value.</td>
</tr>
<tr>
<td>StoredValue</td>
<td>Current value in db.</td>
</tr>
</tbody>
</table>
See Also

SequenceStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `SequenceStats` type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified Object is equal to the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a String that represents the current Object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>
See Also

SequenceStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `SequenceStats` type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CachedValue</td>
<td>Current cached value.</td>
</tr>
<tr>
<td>CacheSize</td>
<td>Cache size.</td>
</tr>
<tr>
<td>Flags</td>
<td>Flag value.</td>
</tr>
<tr>
<td>LastCachedValue</td>
<td>Last cached value.</td>
</tr>
<tr>
<td>LockNoWait</td>
<td>Sequence lock granted after wait.</td>
</tr>
<tr>
<td>LockWait</td>
<td>Sequence lock granted w/o wait.</td>
</tr>
<tr>
<td>Max</td>
<td>Maximum value.</td>
</tr>
<tr>
<td>Min</td>
<td>Minimum value.</td>
</tr>
<tr>
<td>StoredValue</td>
<td>Current value in db.</td>
</tr>
</tbody>
</table>
See Also

SequenceStats Class  
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Current cached value.

Namespace: BerkeleyDB
**Syntax**

**C#**

```csharp
public long CachedValue { get; }
```

**Visual Basic (Declaration)**

```
Public ReadOnly Property CachedValue As Long
```

**Visual C++**

```cpp
public:
    property long long CachedValue {
        long long get ();
    }
```
See Also

SequenceStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Cache size.

**Namespace:** [BerkeleyDB](http://example.com/BerkeleyDB)

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public int CacheSize { get; }

Visual Basic (Declaration)

Public ReadOnly Property CacheSize As Integer

Visual C++

public:
property int CacheSize {
    int get ();
}

See Also

SequenceStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Flag value.

Namespace: BerkeleyDB
Syntax

C#

public uint Flags { get; }

Visual Basic (Declaration)

Public ReadOnly Property Flags As UInteger

Visual C++

public:
    property unsigned int Flags {
        unsigned int get ();
    }

See Also

SequenceStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Last cached value.

**Namespace:**  BerkeleyDB  
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll)  Version: 4.8.24.0
### Syntax

#### C#

```csharp
public long LastCachedValue { get; }
```

#### Visual Basic (Declaration)

```vbnet
Public ReadOnly Property LastCachedValue As Long
```

#### Visual C++

```cpp
public:
    property long long LastCachedValue {
        long long get ();
    }
```


See Also

SequenceStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Sequence lock granted after wait.

Namespace:  BerkeleyDB
Syntax

C#

public ulong LockNoWait { get; }

Visual Basic (Declaration)

Public ReadOnly Property LockNoWait As ULong

Visual C++

public:
property unsigned long long LockNoWait {
    unsigned long long get ();
}
See Also

SequenceStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
SequenceStats.LockWait Property

SequenceStats Class  See Also

Sequence lock granted w/o wait.

Namespace:  BerkeleyDB
Syntax

C#

public ulong LockWait { get; }

Visual Basic (Declaration)

Public ReadOnly Property LockWait As ULong

Visual C++

public:
property unsigned long long LockWait {
    unsigned long long get ();
}

See Also

SequenceStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Maximum value.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
## Syntax

### C#

```csharp
public long Max { get; }
```

### Visual Basic (Declaration)

```
Public ReadOnly Property Max As Long
```

### Visual C++

```cpp
public:
property long long Max {
    long long get ();
}
```
See Also

SequenceStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Minimum value.

Namespace: BerkeleyDB
Syntax

C#

```csharp
public long Min { get; }
```

Visual Basic (Declaration)

```vbnet
Public Readonly Property Min As Long
```

Visual C++

```cpp
public:
    property long long Min {
        get { return long long get (); }
    }
```

See Also

SequenceStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Current value in db.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
**Syntax**

**C#**

```csharp
public long StoredValue { get; }
```

**Visual Basic (Declaration)**

```vbnet
Public ReadOnly Property StoredValue As Long
```

**Visual C++**

```cpp
public:
property long long StoredValue {
    long long get ();
}
```
See Also

SequenceStats Class  
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
A function which returns a unique identifier pair for a thread of control in a Berkeley DB application.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
**Syntax**

**C#**

public delegate DbThreadID SetThreadIDDelegate()

**Visual Basic (Declaration)**

Public Delegate Function SetThreadIDDelegate As DbThreadID

**Visual C++**

public delegate DbThreadID^ SetThreadIDDelegate()

**Return Value**

A DbThreadID object describing the current thread of control
See Also

BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
A function which returns an identifier pair for a thread of control formatted for display.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
**Syntax**

**C#**

```csharp
public delegate string SetThreadNameDelegate(
    DbThreadID info
)
```

**Visual Basic (Declaration)**

```vbnet
Public Delegate Function SetThreadNameDelegate ( _
    info As DbThreadID _
) As String
```

**Visual C++**

```cpp
public delegate String^ SetThreadNameDelegate(
    DbThreadID^ info
)
```

**Parameters**

info

Type: BerkeleyDB::::DbThreadID

The thread of control to format

**Return Value**

The formatted identifier pair
See Also

BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
ThreadIsAliveDelegate Delegate

See Also

A function which returns whether the thread of control, identified by `info`, is still running.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
**Syntax**

**C#**

```csharp
public delegate bool ThreadIsAliveDelegate(
    DbThreadID info,
    bool procOnly
)
```

**Visual Basic (Declaration)**

```vbnet
Public Delegate Function ThreadIsAliveDelegate ( _
    info As DbThreadID, _
    procOnly As Boolean _
) As Boolean
```

**Visual C++**

```cpp
public delegate bool ThreadIsAliveDelegate(
    DbThreadID^ info,
    bool procOnly
)
```

**Parameters**

- **info**
  Type: BerkeleyDB::::DbThreadID
  The thread of control to check

- **procOnly**
  Type: System::::Boolean
  If true, return only if the process is alive, and the threadID portion of info should be ignored.

**Return Value**

True if the thread is alive, false otherwise.
See Also

BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
A class representing Berkeley DB transactions

**Namespace:**  [BerkeleyDB](#)

**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
**Syntax**

**C#**

```csharp
public class Transaction
```

**Visual Basic (Declaration)**

```vbnet
Public Class Transaction
```

**Visual C++**

```cpp
public ref class Transaction
```
Remarks

Calling `Abort()`, `Commit()` or `Discard()` will release the resources held by the created object.

Transactions may only span threads if they do so serially; that is, each transaction must be active in only a single thread of control at a time. This restriction holds for parents of nested transactions as well; no two children may be concurrently active in more than one thread of control at any one time.

Cursors may not span transactions; that is, each cursor must be opened and closed within a single transaction.

A parent transaction may not issue any Berkeley DB operations — except for `BeginTransaction()`, `Abort()` and `Commit()` — while it has active child transactions (child transactions that have not yet been committed or aborted).
Inheritance Hierarchy

System::Object
BerkeleyDB::Transaction
See Also

Transaction Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The **Transaction** type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Abort</strong></td>
<td>Cause an abnormal termination of the transaction. Overloaded. In the case of nested transactions, if the transaction is a parent transaction, committing the parent transaction causes all unresolved children of the parent to be committed. In the case of nested transactions, if the transaction is a child transaction, its locks are not released, but are acquired by its parent. Although the commit of the child transaction will succeed, the actual resolution of the child transaction is postponed until the parent transaction is committed or aborted; that is, if its parent transaction commits, it will be committed; and if its parent transaction aborts, it will be aborted. All cursors opened within the transaction must be closed before the transaction is committed.</td>
</tr>
<tr>
<td><strong>Commit</strong></td>
<td>Free up all the per-process resources associated with the specified Transaction instance, neither committing nor aborting the transaction. Determines whether the specified Object is equal to the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>Discard</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>Equals</strong></td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Set the timeout value for locks for this transaction.</td>
</tr>
<tr>
<td><strong>GetTxnTimeout</strong></td>
<td>Set the timeout value for transactions for this</td>
</tr>
<tr>
<td><strong>Prepare</strong></td>
<td>Initiate the beginning of a two-phase commit.</td>
</tr>
<tr>
<td><strong>SetLockTimeout</strong></td>
<td></td>
</tr>
</tbody>
</table>
transaction.

- **ToString**

Returns a **String** that represents the current **Object**.
(Inherited from **Object**.)
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>GlobalIdLength</code></td>
<td>The size of the global transaction ID</td>
</tr>
</tbody>
</table>
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id</td>
<td>The unique transaction id associated with this transaction. The transaction's name. The name is returned by <code>TransactionSystemStats()</code> and displayed by <code>PrintTransactionSystemStats()</code>.</td>
</tr>
</tbody>
</table>
See Also

Transaction Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `Transaction` type exposes the following members.
Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>GlobalIdLength</strong></td>
<td>The size of the global transaction ID</td>
</tr>
</tbody>
</table>
See Also

Transaction Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The size of the global transaction ID

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public static uint GlobalIdLength

Visual Basic (Declaration)

Public Shared GlobalIdLength As UInteger

Visual C++

public:
static unsigned int GlobalIdLength
See Also

Transaction Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The **Transaction** type exposes the following members.
# Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Abort</strong></td>
<td>Cause an abnormal termination of the transaction. Overloaded. In the case of nested transactions, if the transaction is a parent transaction, committing the parent transaction causes all unresolved children of the parent to be committed. In the case of nested transactions, if the transaction is a child transaction, its locks are not released, but are acquired by its parent. Although the commit of the child transaction will succeed, the actual resolution of the child transaction is postponed until the parent transaction is committed or aborted; that is, if its parent transaction commits, it will be committed; and if its parent transaction aborts, it will be aborted. All cursors opened within the transaction must be closed before the transaction is committed.</td>
</tr>
<tr>
<td><strong>Commit</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Discard</strong></td>
<td>Free up all the per-process resources associated with the specified Transaction instance, neither committing nor aborting the transaction. Determines whether the specified Object is equal to the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>Equals</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Prepare</strong></td>
<td>Initiate the beginning of a two-phase commit.</td>
</tr>
<tr>
<td><strong>SetLockTimeout</strong></td>
<td>Set the timeout value for locks for this transaction.</td>
</tr>
<tr>
<td><strong>SetTxnTimeout</strong></td>
<td>Set the timeout value for transactions for this transaction.</td>
</tr>
</tbody>
</table>
- **ToString**

  Returns a `String` that represents the current `Object`.
  (Inherited from `Object`.)
See Also

Transaction Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Abort Method

Transaction Class  See Also

Cause an abnormal termination of the transaction.

Namespace:  BerkeleyDB
Syntax

C#

public void Abort()

Visual Basic (Declaration)

Public Sub Abort

Visual C++

public:
void Abort()
Remarks

Before Abort returns, any locks held by the transaction will have been released.

In the case of nested transactions, aborting a parent transaction causes all children (unresolved or not) of the parent transaction to be aborted.

All cursors opened within the transaction must be closed before the transaction is aborted.
See Also

Transaction Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
In the case of nested transactions, if the transaction is a parent transaction, committing the parent transaction causes all unresolved children of the parent to be committed. In the case of nested transactions, if the transaction is a child transaction, its locks are not released, but are acquired by its parent. Although the commit of the child transaction will succeed, the actual resolution of the child transaction is postponed until the parent transaction is committed or aborted; that is, if its parent transaction commits, it will be committed; and if its parent transaction aborts, it will be aborted.

All cursors opened within the transaction must be closed before the transaction is committed.
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Commit()()</code></td>
<td>End the transaction.</td>
</tr>
<tr>
<td><code>Commit(Boolean)</code></td>
<td>End the transaction.</td>
</tr>
</tbody>
</table>
See Also

Transaction Class
Transaction Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
End the transaction.

Namespace:  BerkeleyDB
Syntax

C#

public void Commit()

Visual Basic (Declaration)

Public Sub Commit

Visual C++

public:
void Commit()
See Also

Transaction Class
Commit Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
End the transaction.

**Namespace:**  BerkeleyDB  
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll)  Version: 4.8.24.0
Syntax

C#

```csharp
public void Commit(
    bool syncLog
)
```

Visual Basic (Declaration)

```vbnet
Public Sub Commit (_
    syncLog As Boolean _
)
```

Visual C++

```cpp
public:
void Commit(
    bool syncLog
)
```

Parameters

syncLog
Type: System::Boolean
If true, synchronously flush the log.
**Remarks**

Synchronously flushing the log is the default for Berkeley DB environments unless `TxnNoSync` was specified. Synchronous log flushing may also be set or unset for a single transaction using `BeginTransaction()`. The value of `syncLog` overrides both of those settings.
See Also

Transaction Class
Commit Overload
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Free up all the per-process resources associated with the specified Transaction instance, neither committing nor aborting the transaction.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public void Discard()

Visual Basic (Declaration)

Public Sub Discard

Visual C++

public:
void Discard()
Remarks

This call may be used only after calls to `Recover(UInt32, Boolean)` when there are multiple global transaction managers recovering transactions in a single Berkeley DB environment. Any transactions returned by `Recover(UInt32, Boolean)` that are not handled by the current global transaction manager should be discarded using Discard.
See Also

Transaction Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Initiate the beginning of a two-phase commit.

**Namespace:**  BerkeleyDB
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
**Syntax**

**C#**

```csharp
public void Prepare(
    byte[] globalId
)
```

**Visual Basic (Declaration)**

```vbnet
Public Sub Prepare (_
    globalId As Byte() _
)
```

**Visual C++**

```cpp
public:
void Prepare(
    array<unsigned char>^ globalId
)
```

**Parameters**

globalId

Type: array< System::::Byte >[](1)[]

The global transaction ID by which this transaction will be known. This global transaction ID will be returned in calls to `Recover(UInt32, Boolean)` telling the application which global transactions must be resolved.
Remarks

In a distributed transaction environment, Berkeley DB can be used as a local transaction manager. In this case, the distributed transaction manager must send prepare messages to each local manager. The local manager must then call Prepare and await its successful return before responding to the distributed transaction manager. Only after the distributed transaction manager receives successful responses from all of its prepare messages should it issue any commit messages.

In the case of nested transactions, preparing the parent causes all unresolved children of the parent transaction to be committed. Child transactions should never be explicitly prepared. Their fate will be resolved along with their parent's during global recovery.
See Also

Transaction Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Set the timeout value for locks for this transaction.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public void SetLockTimeout(
    uint timeout
)

Visual Basic (Declaration)

Public Sub SetLockTimeout (_
    timeout As UInteger _
)

Visual C++

public:
void SetLockTimeout(_
    unsigned int timeout
)

Parameters

timeout
Type: System::::UInt32
An unsigned 32-bit number of microseconds, limiting the maximum timeout to roughly 71 minutes. A value of 0 disables timeouts for the transaction.
Remarks

Timeouts are checked whenever a thread of control blocks on a lock or when deadlock detection is performed. This timeout is for any single lock request. As timeouts are only checked when the lock request first blocks or when deadlock detection is performed, the accuracy of the timeout depends on how often deadlock detection is performed.

Timeout values may be specified for the database environment as a whole. See LockTimeout for more information.
See Also

Transaction Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Set the timeout value for transactions for this transaction.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

**C#**

```csharp
public void SetTxnTimeout(
    uint timeout
)
```

**Visual Basic (Declaration)**

```vbnet
Public Sub SetTxnTimeout ( _
    timeout As UInteger _
)
```

**Visual C++**

```cpp
public:
void SetTxnTimeout(
    unsigned int timeout
)
```

**Parameters**

timeout
Type: `System::::UInt32`
An unsigned 32-bit number of microseconds, limiting the maximum timeout to roughly 71 minutes. A value of 0 disables timeouts for the transaction.
Remarks

Timeouts are checked whenever a thread of control blocks on a lock or when deadlock detection is performed. This timeout is for the life of the transaction. As timeouts are only checked when the lock request first blocks or when deadlock detection is performed, the accuracy of the timeout depends on how often deadlock detection is performed.

Timeout values may be specified for the database environment as a whole. See TxnTimeout for more information.
See Also

Transaction Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `Transaction` type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id</td>
<td>The unique transaction id associated with this transaction. The transaction's name. The name is returned by <code>TransactionSystemStats()</code> and displayed by <code>PrintTransactionSystemStats()</code>.</td>
</tr>
<tr>
<td>Name</td>
<td><code>TransactionSystemStats()</code></td>
</tr>
</tbody>
</table>
See Also

Transaction Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The unique transaction id associated with this transaction.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public uint Id { get; }

Visual Basic (Declaration)

Public ReadOnly Property Id As UInteger

Visual C++

public:
property unsigned int Id {
    unsigned int get ();
};
See Also

Transaction Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The transaction's name. The name is returned by `TransactionSystemStats()` and displayed by `PrintTransactionSystemStats()`.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public string Name { get; set; }

Visual Basic (Declaration)

Public Property Name As String

Visual C++

public:
property String^ Name {
    String^ get ();
    void set (String^ value);
}
Remarks

If the database environment has been configured for logging and the Berkeley DB library was built in Debug mode (or with DIAGNOSTIC defined), a debugging log record is written including the transaction ID and the name.
See Also

Transaction Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
A class representing configuration parameters for a Transaction.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public class TransactionConfig

Visual Basic (Declaration)

Public Class TransactionConfig

Visual C++

public ref class TransactionConfig
Inheritance Hierarchy

System::Object
BerkeleyDB::TransactionConfig
See Also

TransactionConfig Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `TransactionConfig` type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TransactionConfig</td>
<td>Instantiate a new TransactionConfig object</td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <code>Object</code> is equal to the current <code>Object</code>.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <code>Type</code> of the current instance.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a <code>String</code> that represents the current <code>Object</code>.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.</td>
</tr>
</tbody>
</table>
### Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IsolationDegree</td>
<td>The degree of isolation for this transaction. If true and a lock is unavailable for any Berkeley DB operation performed in the context of a transaction, cause the operation to throw a DeadlockException (or LockNotGrantedException if configured with TimeNotGranted).</td>
</tr>
<tr>
<td>NoWait</td>
<td>If true, this transaction will execute with snapshot isolation.</td>
</tr>
<tr>
<td>Snapshot</td>
<td>Log sync behavior on transaction commit or prepare.</td>
</tr>
</tbody>
</table>
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LockTimeout</td>
<td>The timeout value for locks for the transaction.</td>
</tr>
<tr>
<td>Name</td>
<td>The transaction's name. The name is returned by <code>TransactionSystemStats()</code>, and displayed by <code>PrintTransactionSystemStats()</code></td>
</tr>
<tr>
<td>TxnTimeout</td>
<td>The timeout value for locks for the transaction.</td>
</tr>
</tbody>
</table>
See Also

TransactionConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Instantiate a new TransactionConfig object

Namespace: BerkeleyDB
Syntax

C#

public TransactionConfig()

Visual Basic (Declaration)

Public Sub New

Visual C++

public:
TransactionConfig()
See Also

TransactionConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `TransactionConfig` type exposes the following members.
## Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>IsolationDegree</strong></td>
<td>The degree of isolation for this transaction. If true and a lock is unavailable for any Berkeley DB operation performed in the context of a transaction, cause the operation to throw a <code>DeadlockException</code> (or <code>LockNotGrantedException</code> if configured with <code>TimeNotGranted</code>).</td>
</tr>
<tr>
<td><strong>NoWait</strong></td>
<td>If true, this transaction will execute with snapshot isolation.</td>
</tr>
<tr>
<td><strong>Snapshot</strong></td>
<td>Log sync behavior on transaction commit or prepare.</td>
</tr>
</tbody>
</table>
See Also

TransactionConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The degree of isolation for this transaction

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public Isolation IsolationDegree

Visual Basic (Declaration)

Public IsolationDegree As Isolation

Visual C++

public:
Isolation IsolationDegree
See Also

TransactionConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true and a lock is unavailable for any Berkeley DB operation performed in the context of a transaction, cause the operation to throw a `DeadlockException` (or `LockNotGrantedException` if configured with `TimeNotGranted`).

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

**C#**

```csharp
public bool NoWait
```

**Visual Basic (Declaration)**

```vbnet
Public NoWait As Boolean
```

**Visual C++**

```cpp
public:
    bool NoWait
```
Remarks

This setting overrides the behavior specified by TxnNoWait.
See Also

TransactionConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
If true, this transaction will execute with snapshot isolation.

**Namespace:**  [BerkeleyDB](#)

**Assembly:**  `libdb_dotnet48` (in `libdb_dotnet48.dll`) Version: 4.8.24.0
Syntax

C#

public bool Snapshot

Visual Basic (Declaration)

Public Snapshot As Boolean

Visual C++

public:
bool Snapshot
Remarks

For databases with UseMVCC set, data values will be read as they are when the transaction begins, without taking read locks. Silently ignored for operations on databases with UseMVCC not set on the underlying database (read locks are acquired).

A DeadlockException will be thrown from update operations if a snapshot transaction attempts to update data which was modified after the snapshot transaction read it.
See Also

TransactionConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Log sync behavior on transaction commit or prepare.

**Namespace:**  BerkeleyDB

**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public TransactionConfig:::LogFlush SyncAction

Visual Basic (Declaration)

Public SyncAction As TransactionConfig:::LogFlush

Visual C++

public:
TransactionConfig:::LogFlush SyncAction
Remarks

This setting overrides the behavior specified by TxnNoSync and TxnWriteNoSync.
See Also

TransactionConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `TransactionConfig` type exposes the following members.
### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="#">Object</a>)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a>)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a <a href="#">String</a> that represents the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>)</td>
</tr>
</tbody>
</table>
See Also

TransactionConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `TransactionConfig` type exposes the following members.
# Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>LockTimeout</strong></td>
<td>The timeout value for locks for the transaction.</td>
</tr>
<tr>
<td><strong>Name</strong></td>
<td>The transaction's name. The name is returned by <code>TransactionSystemStats()</code> and displayed by <code>PrintTransactionSystemStats()</code>.</td>
</tr>
<tr>
<td><strong>TxnTimeout</strong></td>
<td>The timeout value for locks for the transaction.</td>
</tr>
</tbody>
</table>
See Also

TransactionConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The timeout value for locks for the transaction.

**Namespace:**  [BerkeleyDB](#)

**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#
public uint LockTimeout { get; set; }

Visual Basic (Declaration)
Public Property LockTimeout AsUInteger

Visual C++
public:
property unsigned int LockTimeout {
    unsigned int get ();
    void set (unsigned int value);
}
Remarks

Timeouts are checked whenever a thread of control blocks on a lock or when deadlock detection is performed. This timeout is for any single lock request. As timeouts are only checked when the lock request first blocks or when deadlock detection is performed, the accuracy of the timeout depends on how often deadlock detection is performed.

Timeout values may be specified for the database environment as a whole. See LockTimeout for more information.
See Also

TransactionConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The transaction's name. The name is returned by `TransactionSystemStats()` and displayed by `PrintTransactionSystemStats()`.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public string Name { get; set; }

Visual Basic (Declaration)

Public Property Name As String

Visual C++

public:
property String^ Name {
    String^ get ();
    void set (String^ value);
}
Remarks

If the database environment has been configured for logging and the Berkeley DB library was built in Debug mode (or with DIAGNOSTIC defined), a debugging log record is written including the transaction ID and the name.
See Also

TransactionConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
TransactionConfig...:TxnTimeout Property

The timeout value for locks for the transaction.

Namespace: BerkeleyDB
Syntax

C#

public uint TxnTimeout { get; set; }

Visual Basic (Declaration)

Public Property TxnTimeout As UInteger

Visual C++

public:
property unsigned int TxnTimeout {
    unsigned int get ();
    void set (unsigned int value);
}
Remarks

Timeouts are checked whenever a thread of control blocks on a lock or when deadlock detection is performed. This timeout is for the life of the transaction. As timeouts are only checked when the lock request first blocks or when deadlock detection is performed, the accuracy of the timeout depends on how often deadlock detection is performed.

Timeout values may be specified for the database environment as a whole. See TxnTimeout for more information.
See Also

TransactionConfig Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Specifies the log flushing behavior on transaction commit

**Namespace:**  [BerkeleyDB](#)

**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public enum LogFlush

Visual Basic (Declaration)

Public Enumeration LogFlush

Visual C++

public enum class LogFlush
<table>
<thead>
<tr>
<th>Member name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEFAULT</td>
<td>Use Berkeley DB’s default behavior of syncing the log on commit.</td>
</tr>
<tr>
<td>NOSYNC</td>
<td>Berkeley DB will not write or synchronously flush the log on transaction commit or prepare.</td>
</tr>
<tr>
<td>WRITE_NOSYNC</td>
<td>Berkeley DB will write, but will not synchronously flush, the log on transaction commit or prepare.</td>
</tr>
<tr>
<td>SYNC</td>
<td>Berkeley DB will synchronously flush the log on transaction commit or prepare.</td>
</tr>
</tbody>
</table>
See Also

BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Statistical information about the transaction subsystem

**Namespace:**  BerkeleyDB

**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#
public class TransactionStats

Visual Basic (Declaration)
Public Class TransactionStats

Visual C++
public ref class TransactionStats
Inheritance Hierarchy

System...:::Object
BerkeleyDB...:::TransactionStats
See Also

TransactionStats Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The **TransactionStats** type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified <code>Object</code> is equal to the current <code>Object</code>.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as a hash function for a particular type.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the <code>Type</code> of the current instance.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a <code>String</code> that represents the current <code>Object</code>.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
</tbody>
</table>
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aborted</td>
<td>Number of aborted transactions</td>
</tr>
<tr>
<td>Active</td>
<td>Number of active transactions</td>
</tr>
<tr>
<td>Begun</td>
<td>Number of begun transactions</td>
</tr>
<tr>
<td>Committed</td>
<td>Number of committed transactions</td>
</tr>
<tr>
<td>LastCheckpoint</td>
<td>LSN of the last checkpoint</td>
</tr>
<tr>
<td>LastCheckpointTime</td>
<td>Time of last checkpoint</td>
</tr>
<tr>
<td>LastID</td>
<td>Last transaction id given out</td>
</tr>
<tr>
<td>MaxActive</td>
<td>Maximum active transactions</td>
</tr>
<tr>
<td>MaxSnapshot</td>
<td>Maximum snapshot transactions</td>
</tr>
<tr>
<td>MaxTransactions</td>
<td>Maximum txns possible</td>
</tr>
<tr>
<td>RegionLockNoWait</td>
<td>Region lock granted without wait.</td>
</tr>
<tr>
<td>RegionLockWait</td>
<td>Region lock granted after wait.</td>
</tr>
<tr>
<td>RegionSize</td>
<td>Region size.</td>
</tr>
<tr>
<td>Restored</td>
<td>Number of restored transactions after recovery.</td>
</tr>
<tr>
<td>Snapshot</td>
<td>Number of snapshot transactions</td>
</tr>
<tr>
<td>Transactions</td>
<td>List of active transactions</td>
</tr>
</tbody>
</table>
See Also

TransactionStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `TransactionStats` type exposes the following members.
### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="#">Object</a> is equal to the current Object. (Inherited from <a href="#">Object</a>).</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="#">Object</a>).</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the Type of the current instance. (Inherited from <a href="#">Object</a>).</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a String that represents the current Object. (Inherited from <a href="#">Object</a>).</td>
</tr>
</tbody>
</table>
See Also

TransactionStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `TransactionStats` type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aborted</td>
<td>Number of aborted transactions</td>
</tr>
<tr>
<td>Active</td>
<td>Number of active transactions</td>
</tr>
<tr>
<td>Begun</td>
<td>Number of begun transactions</td>
</tr>
<tr>
<td>Committed</td>
<td>Number of committed transactions</td>
</tr>
<tr>
<td>LastCheckpoint</td>
<td>LSN of the last checkpoint</td>
</tr>
<tr>
<td>LastCheckpointTime</td>
<td>Time of last checkpoint</td>
</tr>
<tr>
<td>LastID</td>
<td>Last transaction id given out</td>
</tr>
<tr>
<td>MaxActive</td>
<td>Maximum active transactions</td>
</tr>
<tr>
<td>MaxSnapshot</td>
<td>Maximum snapshot transactions</td>
</tr>
<tr>
<td>MaxTransactions</td>
<td>Maximum txns possible</td>
</tr>
<tr>
<td>RegionLockNoWait</td>
<td>Region lock granted without wait.</td>
</tr>
<tr>
<td>RegionLockWait</td>
<td>Region lock granted after wait.</td>
</tr>
<tr>
<td>RegionSize</td>
<td>Region size.</td>
</tr>
<tr>
<td>Restored</td>
<td>Number of restored transactions after recovery.</td>
</tr>
<tr>
<td>Snapshot</td>
<td>Number of snapshot transactions</td>
</tr>
<tr>
<td>Transactions</td>
<td>List of active transactions</td>
</tr>
</tbody>
</table>
See Also

TransactionStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
TransactionStats...:.Aborted Property

TransactionStats Class  See Also

Number of aborted transactions

Namespace:  BerkeleyDB
Syntax

C#

public ulong Aborted { get; }

Visual Basic (Declaration)

Public ReadOnly Property Aborted As ULong

Visual C++

public:
property unsigned long long Aborted {
    unsigned long long get ();
}


See Also

TransactionStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Number of active transactions

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public uint Active { get; }

Visual Basic (Declaration)

Public ReadOnly Property Active AsUInteger

Visual C++

public:
property unsigned int Active { unsigned int get (); }
See Also

TransactionStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
TransactionStats::Begun Property

Number of begun transactions

Namespace: BerkeleyDB
Syntax

C#

public ulong Begun { get; }

Visual Basic (Declaration)

Public ReadOnly Property Begun As ULong

Visual C++

public:
property unsigned long long Begun {
  unsigned long long get ();
}

See Also

TransactionStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
TransactionStats Class  See Also
Number of committed transactions

Namespace:  BerkeleyDB
Syntax

C#

public ulong Committed { get; }

Visual Basic (Declaration)

Public ReadOnly Property Committed As ULong

Visual C++

public:
property unsigned long long Committed {
    unsigned long long get ();
}

See Also

TransactionStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
LSN of the last checkpoint

Namespace: BerkeleyDB
Syntax

C#

public LSN LastCheckpoint { get; }

Visual Basic (Declaration)

Public ReadOnly Property LastCheckpoint As LSN

Visual C++

public:
property LSN^ LastCheckpoint {
    LSN^ get ();
}

See Also

TransactionStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
TransactionStats::LastCheckpointTime Property

TransactionStats Class  See Also

Time of last checkpoint

Namespace:  BerkeleyDB
Syntax

C#

public long LastCheckpointTime { get; }

Visual Basic (Declaration)

Public ReadOnly Property LastCheckpointTime As Long

Visual C++

public:
property long long LastCheckpointTime {
   long long get ();
}
See Also

TransactionStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
TransactionStats

LastID Property

Last transaction id given out

Namespace: BerkeleyDB
Syntax

C#

public uint LastID { get; }

Visual Basic (Declaration)

Public ReadOnly Property LastID As UInteger

Visual C++

public:
property unsigned int LastID {
    unsigned int get ();
}


See Also

TransactionStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Maximum active transactions

Namespace:  BerkeleyDB
Syntax

C#

public uint MaxActive { get; }

Visual Basic (Declaration)

Public ReadOnly Property MaxActive As UInteger

Visual C++

public:
property unsigned int MaxActive {
    unsigned int get ();
}
See Also

TransactionStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
TransactionStats::MaxSnapshot Property

Namespace: BerkeleyDB
Syntax

C#

public uint MaxSnapshot { get; }

Visual Basic (Declaration)

Public ReadOnly Property MaxSnapshot AsUInteger

Visual C++

public:
property unsigned int MaxSnapshot {
    unsigned int get ();
}

See Also

TransactionStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
TransactionStats Class  See Also

Maximum txns possible

Namespace:  BerkeleyDB
Syntax

C#

public uint MaxTransactions { get; }

Visual Basic (Declaration)

Public ReadOnly Property MaxTransactions As UInteger

Visual C++

public:
    property unsigned int MaxTransactions {
        unsigned int get ();
    }

See Also

TransactionStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Region lock granted without wait.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public ulong RegionLockNoWait { get; }

Visual Basic (Declaration)

Public ReadOnly Property RegionLockNoWait As ULong

Visual C++

public:
property unsigned long long RegionLockNoWait {
    unsigned long long get ();
}


See Also

TransactionStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Region lock granted after wait.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public ulong RegionLockWait { get; }

Visual Basic (Declaration)

Public ReadOnly Property RegionLockWait As ULong

Visual C++

public:
property unsigned long long RegionLockWait {
    unsigned long long get ();
}

See Also

TransactionStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Region size.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public ulong RegionSize { get; }

Visual Basic (Declaration)

Public ReadOnly Property RegionSize As ULong

Visual C++

public:
property unsigned long long RegionSize {
    unsigned long long get ();
}

See Also

TransactionStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Number of restored transactions after recovery.

Namespace: BerkeleyDB
Syntax

C#

public uint Restored { get; }

Visual Basic (Declaration)

Public ReadOnly Property Restored As UInteger

Visual C++

public:
property unsigned int Restored {
    unsigned int get ();
}

See Also

TransactionStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
TransactionStats Class  See Also

Number of snapshot transactions

Namespace:  BerkeleyDB
Syntax

C#

public uint Snapshot { get; }

Visual Basic (Declaration)

Public ReadOnly Property Snapshot As UInteger

Visual C++

public:
property unsigned int Snapshot {
    unsigned int get ();
}

See Also

TransactionStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
List of active transactions

**Namespace:**  [BerkeleyDB](#)
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public List<ActiveTransaction> Transactions { get; }

Visual Basic (Declaration)

Public ReadOnly Property Transactions As List(Of ActiveTransaction)

Visual C++

public:
property List<ActiveTransaction>^ Transactions {
    List<ActiveTransaction>^ get ();
}
See Also

TransactionStats Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Enable specific additional informational and debugging messages.

**Namespace:**  BerkeleyDB  
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll)  
**Version:**  4.8.24.0
Syntax

**C#**

```
public class VerboseMessages
```

**Visual Basic (Declaration)**

```
Public Class VerboseMessages
```

**Visual C++**

```
public ref class VerboseMessages
```
Inheritance Hierarchy

System..::..Object
BerkeleyDB..::..VerboseMessages
See Also

VerboseMessages Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The **VerboseMessages** type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VerboseMessages</td>
<td>Initializes a new instance of the <code>VerboseMessages</code> class</td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <code>Object</code> is equal to the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <code>Type</code> of the current instance. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a <code>String</code> that represents the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>AllFileOps</td>
<td>Display additional information when performing all filesystem operations,</td>
</tr>
<tr>
<td></td>
<td>including read and write. May not be available on all platforms.</td>
</tr>
<tr>
<td>Deadlock</td>
<td>Display additional information when doing deadlock detection.</td>
</tr>
<tr>
<td>FileOps</td>
<td>Display additional information when performing filesystem operations such</td>
</tr>
<tr>
<td></td>
<td>as open, close or rename. May not be available on all platforms.</td>
</tr>
<tr>
<td>Recovery</td>
<td>Display additional information when performing recovery.</td>
</tr>
<tr>
<td>Register</td>
<td>Display additional information concerning support for Register.</td>
</tr>
<tr>
<td>Replication</td>
<td>Display all detailed information about replication. This includes the</td>
</tr>
<tr>
<td></td>
<td>information displayed by all of the other Replication* and RepMgr* values.</td>
</tr>
<tr>
<td>ReplicationElection</td>
<td>Display detailed information about replication elections.</td>
</tr>
<tr>
<td>ReplicationLease</td>
<td>Display detailed information about replication master leases.</td>
</tr>
<tr>
<td>ReplicationMessages</td>
<td>Display detailed information about replication message processing.</td>
</tr>
<tr>
<td>ReplicationMisc</td>
<td>Display detailed information about general replication processing not</td>
</tr>
<tr>
<td></td>
<td>covered by the other Replication* values.</td>
</tr>
<tr>
<td>ReplicationSync</td>
<td>Display detailed information about replication client synchronization.</td>
</tr>
<tr>
<td>RepMgrConnectionFailure</td>
<td>Display detailed information about</td>
</tr>
<tr>
<td>Feature</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>------------------------------------------------------------------</td>
</tr>
<tr>
<td>Replication Manager connection failures.</td>
<td>Display detailed information about general Replication Manager processing.</td>
</tr>
</tbody>
</table>
See Also

VerboseMessages Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Berkeley DB .NET API Documentation

VerboseMessages Constructor

VerboseMessages Class See Also

Initializes a new instance of the VerboseMessages class

Namespace: BerkeleyDB
Syntax

**C#**

`public VerboseMessages()`

**Visual Basic (Declaration)**

`Public Sub New`  

**Visual C++**

`public:  
VerboseMessages()`
See Also

VerboseMessages Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The **VerboseMessages** type exposes the following members.
# Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AllFileOps</td>
<td>Display additional information when performing all filesystem operations,</td>
</tr>
<tr>
<td></td>
<td>including read and write. May not be available on all platforms.</td>
</tr>
<tr>
<td>Deadlock</td>
<td>Display additional information when doing deadlock detection.</td>
</tr>
<tr>
<td>FileOps</td>
<td>Display additional information when performing filesystem operations</td>
</tr>
<tr>
<td></td>
<td>such as open, close or rename. May not be available on all platforms.</td>
</tr>
<tr>
<td>Recovery</td>
<td>Display additional information when performing recovery.</td>
</tr>
<tr>
<td>Register</td>
<td>Display additional information concerning support for Register.</td>
</tr>
<tr>
<td>Replication</td>
<td>Display all detailed information about replication. This includes the</td>
</tr>
<tr>
<td></td>
<td>information displayed by all of the other Replication* and RepMgr* values.</td>
</tr>
<tr>
<td>ReplicationElection</td>
<td>Display detailed information about replication elections.</td>
</tr>
<tr>
<td>ReplicationLease</td>
<td>Display detailed information about replication master leases.</td>
</tr>
<tr>
<td>ReplicationMessages</td>
<td>Display detailed information about replication message processing.</td>
</tr>
<tr>
<td>ReplicationMisc</td>
<td>Display detailed information about general replication processing not</td>
</tr>
<tr>
<td></td>
<td>covered by the other Replication* values.</td>
</tr>
<tr>
<td>ReplicationSync</td>
<td>Display detailed information about replication client synchronization.</td>
</tr>
<tr>
<td>RepMgrConnectionFailure</td>
<td>Display detailed information about</td>
</tr>
</tbody>
</table>
Replication Manager connection failures.

**RepMgrMisc**
Display detailed information about general Replication Manager processing.

**WaitsForTable**
Display the waits-for table when doing deadlock detection.
See Also

VerboseMessages Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Display additional information when performing all filesystem operations, including read and write. May not be available on all platforms.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool AllFileOps

Visual Basic (Declaration)

Public AllFileOps As Boolean

Visual C++

public:
bool AllFileOps
See Also

VerboseMessages Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Display additional information when doing deadlock detection.

**Namespace:**  [BerkeleyDB](https://example.com/berkleydb)

**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool Deadlock

Visual Basic (Declaration)

Public Deadlock As Boolean

Visual C++

public:
bool Deadlock
See Also

VerboseMessages Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
VerboseMessages...::FileOps Field

Display additional information when performing filesystem operations such as open, close or rename. May not be available on all platforms.

**Namespace:** BerkeleyDB
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool FileOps

Visual Basic (Declaration)

Public FileOps As Boolean

Visual C++

public:
bool FileOps
See Also

VerboseMessages Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Display additional information when performing recovery.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
**Syntax**

**C#**

public bool Recovery

**Visual Basic (Declaration)**

Public Recovery As Boolean

**Visual C++**

public:

bool Recovery
See Also

VerboseMessages Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Display additional information concerning support for Register.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool Register

Visual Basic (Declaration)

Public Register As Boolean

Visual C++

public:
bool Register
See Also

VerboseMessages Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Display all detailed information about replication. This includes the information displayed by all of the other Replication* and RepMgr* values.

Namespace: BerkeleyDB
Syntax

C#

public bool Replication

Visual Basic (Declaration)

Public Replication As Boolean

Visual C++

public:

bool Replication
See Also

VerboseMessages Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Display detailed information about replication elections.

**Namespace:** [BerkeleyDB](https://example.com/berkeleydb)

**Assembly:** `libdb_dotnet48` (in `libdb_dotnet48.dll`) Version: 4.8.24.0
Syntax

**C#**

```csharp
public bool ReplicationElection
```

**Visual Basic (Declaration)**

```vbnet
Public ReplicationElection As Boolean
```

**Visual C++**

```cpp
public:
bool ReplicationElection
```
See Also

VerboseMessages Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Display detailed information about replication master leases.

**Namespace:**  BerkeleyDB  
**Assembly:**  libdb_dotnet48 (in libdb_dotnet48.dll)  Version: 4.8.24.0
Syntax

C#

public bool ReplicationLease

Visual Basic (Declaration)

Public ReplicationLease As Boolean

Visual C++

public:
bool ReplicationLease
See Also

VerboseMessages Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Display detailed information about replication message processing.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool ReplicationMessages

Visual Basic (Declaration)

Public ReplicationMessages As Boolean

Visual C++

public:
bool ReplicationMessages
See Also

VerboseMessages Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Display detailed information about general replication processing not covered by the other Replication* values.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool ReplicationMisc

Visual Basic (Declaration)

Public ReplicationMisc As Boolean

Visual C++

public:
bool ReplicationMisc
See Also

VerboseMessages Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Display detailed information about replication client synchronization.

**Namespace:** [BerkeleyDB](https://docs.oracle.com/database/121/bdbook/index.html)

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool ReplicationSync

Visual Basic (Declaration)

Public ReplicationSync As Boolean

Visual C++

public:
bool ReplicationSync
See Also

VerboseMessages Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
VerboseMessages....::ReplicationTest Field

Namespace: BerkeleyDB
Syntax

C#

public bool ReplicationTest

Visual Basic (Declaration)

Public ReplicationTest As Boolean

Visual C++

public:
 bool ReplicationTest
See Also

VerboseMessages Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
VerboseMessages.RepMgrConnectionFailure Field

Display detailed information about Replication Manager connection failures.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool RepMgrConnectionFailure

Visual Basic (Declaration)

Public RepMgrConnectionFailure As Boolean

Visual C++

public:
bool RepMgrConnectionFailure
See Also

VerboseMessages Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Display detailed information about general Replication Manager processing.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll)  
**Version:** 4.8.24.0
Syntax

C#

public bool RepMgrMisc

Visual Basic (Declaration)

Public RepMgrMisc As Boolean

Visual C++

public:
bool RepMgrMisc
See Also

VerboseMessages Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Display the waits-for table when doing deadlock detection.

**Namespace:** BerkeleyDB  
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public bool WaitsForTable

Visual Basic (Declaration)

Public WaitsForTable As Boolean

Visual C++

public:

bool WaitsForTable
See Also

VerboseMessages Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The **VerboseMessages** type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a>.</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the <a href="#">Type</a> of the current instance. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a <a href="#">String</a> that represents the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
</tbody>
</table>
See Also

VerboseMessages Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
Thrown by `Verify(String, DatabaseConfig)` if a database is corrupted, and by `Salvage(String, DatabaseConfig)` if all key/data pairs in the file may not have been successfully output.

**Namespace:** BerkeleyDB

**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public class VerificationException : DatabaseException

Visual Basic (Declaration)

Public Class VerificationException _
    Inherits DatabaseException

Visual C++

public ref class VerificationException : public DatabaseException
Inheritance Hierarchy

System:::Object
System:::Exception
BerkeleyDB:::DatabaseException
BerkeleyDB:::VerificationException
See Also

VerificationException Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The **VerificationException** type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>VerificationException</code></td>
<td>Initialize a new instance of the VerificationException</td>
</tr>
</tbody>
</table>
# Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="/Object">Object</a> is equal to the current <a href="/Object">Object</a>. (Inherited from <a href="/Object">Object</a>.) When overridden in a derived class, returns the <a href="/Exception">Exception</a> that is the root cause of one or more subsequent exceptions. (Inherited from <a href="/Exception">Exception</a>.)</td>
</tr>
<tr>
<td><strong>GetBaseException</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="/Object">Object</a>.) When overridden in a derived class, sets the <a href="/SerializationInfo">SerializationInfo</a> with information about the exception. (Inherited from <a href="/Exception">Exception</a>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Gets the runtime type of the current instance. (Inherited from <a href="/Exception">Exception</a>.) Creates and returns a string representation of the current exception. (Inherited from <a href="/Exception">Exception</a>.)</td>
</tr>
<tr>
<td><strong>GetObjectData</strong></td>
<td></td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td></td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ErrorCode</td>
<td>The underlying error code from the Berkeley DB C library. (Inherited from DatabaseException.)</td>
</tr>
</tbody>
</table>
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data</strong></td>
<td>Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from <em>Exception.</em>)</td>
</tr>
<tr>
<td><strong>HelpLink</strong></td>
<td>Gets or sets a link to the help file associated with this exception. (Inherited from <em>Exception.</em>)</td>
</tr>
<tr>
<td><strong>InnerException</strong></td>
<td>Gets the <em>Exception</em> instance that caused the current exception. (Inherited from <em>Exception.</em>)</td>
</tr>
<tr>
<td><strong>Message</strong></td>
<td>Gets a message that describes the current exception. (Inherited from <em>Exception.</em>)</td>
</tr>
<tr>
<td><strong>Source</strong></td>
<td>Gets or sets the name of the application or the object that causes the error. (Inherited from <em>Exception.</em>)</td>
</tr>
<tr>
<td><strong>StackTrace</strong></td>
<td>Gets a string representation of the frames on the call stack at the time the current exception was thrown. (Inherited from <em>Exception.</em>)</td>
</tr>
<tr>
<td><strong>TargetSite</strong></td>
<td>Gets the method that throws the current exception. (Inherited from <em>Exception.</em>)</td>
</tr>
</tbody>
</table>
See Also

VerificationException Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
VerificationException Constructor

Initialize a new instance of the VerificationException

**Namespace:** BerkeleyDB
**Assembly:** libdb_dotnet48 (in libdb_dotnet48.dll) Version: 4.8.24.0
Syntax

C#

public VerificationException()

Visual Basic (Declaration)

Public Sub New

Visual C++

public:
VerificationException()
See Also

VerificationException Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The **VerificationException** type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ErrorCode</td>
<td>The underlying error code from the Berkeley DB C library. (Inherited from DatabaseException.)</td>
</tr>
</tbody>
</table>
**See Also**

[VerificationException Class](#)
[BerkeleyDB Namespace](#)

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `VerificationException` type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a> is equal to the current <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.) When overridden in a derived class, returns the <a href="https://docs.microsoft.com/en-us/dotnet/api/system.exception">Exception</a> that is the root cause of one or more subsequent exceptions. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.exception">Exception</a>.)</td>
</tr>
<tr>
<td><strong>GetBaseException</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.) When overridden in a derived class, sets the <a href="https://docs.microsoft.com/en-us/dotnet/api/system.serialization.serializationinfo">SerializationInfo</a> with information about the exception. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.exception">Exception</a>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Gets the runtime type of the current instance. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.exception">Exception</a>.) Creates and returns a string representation of the current exception. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.exception">Exception</a>.)</td>
</tr>
<tr>
<td><strong>GetObjectData</strong></td>
<td></td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td></td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td></td>
</tr>
</tbody>
</table>
See Also

VerificationException Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `VerificationException` type exposes the following members.
### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data</strong></td>
<td>Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from <strong>Exception</strong>.)</td>
</tr>
<tr>
<td><strong>HelpLink</strong></td>
<td>Gets or sets a link to the help file associated with this exception.</td>
</tr>
<tr>
<td><strong>InnerException</strong></td>
<td>Gets the <strong>Exception</strong> instance that caused the current exception.</td>
</tr>
<tr>
<td><strong>Message</strong></td>
<td>Gets a message that describes the current exception.</td>
</tr>
<tr>
<td><strong>Source</strong></td>
<td>Gets or sets the name of the application or the object that causes the error. (Inherited from <strong>Exception</strong>.)</td>
</tr>
<tr>
<td><strong>StackTrace</strong></td>
<td>Gets a string representation of the frames on the call stack at the time the current exception was thrown. (Inherited from <strong>Exception</strong>.)</td>
</tr>
<tr>
<td><strong>TargetSite</strong></td>
<td>Gets the method that throws the current exception.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <strong>Exception</strong>.)</td>
</tr>
</tbody>
</table>
See Also

VerificationException Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The version of the Berkeley DB library doesn't match the version that created the database environment.

Namespace: BerkeleyDB
Syntax

C#

public class VersionMismatchException : DatabaseException

Visual Basic (Declaration)

Public Class VersionMismatchException _
    Inherits DatabaseException

Visual C++

public ref class VersionMismatchException : public DatabaseException
Inheritance Hierarchy

System:::Object
System:::Exception
BerkeleyDB:::DatabaseException
BerkeleyDB:::VersionMismatchException
See Also

VersionMismatchException Members
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `VersionMismatchException` type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>VersionMismatchException</code></td>
<td>Initialize a new instance of the <code>VersionMismatchException</code></td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a> is equal to the current <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.) When overridden in a derived class, returns the <a href="https://docs.microsoft.com/en-us/dotnet/api/system.exception">Exception</a> that is the root cause of one or more subsequent exceptions. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.exception">Exception</a>.)</td>
</tr>
<tr>
<td><strong>GetBaseException</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.object">Object</a>.) When overridden in a derived class, sets the <a href="https://docs.microsoft.com/en-us/dotnet/api/system.serialization.serializationinfo">SerializationInfo</a> with information about the exception. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.exception">Exception</a>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Gets the runtime type of the current instance. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.exception">Exception</a>.)</td>
</tr>
<tr>
<td><strong>GetObjectData</strong></td>
<td>Creates and returns a string representation of the current exception. (Inherited from <a href="https://docs.microsoft.com/en-us/dotnet/api/system.exception">Exception</a>.)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ErrorCode</td>
<td>The underlying error code from the Berkeley DB C library.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <a href="#">DatabaseException</a>.)</td>
</tr>
</tbody>
</table>
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data</strong></td>
<td>Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td><strong>HelpLink</strong></td>
<td>Gets or sets a link to the help file associated with this exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td><strong>InnerException</strong></td>
<td>Gets the Exception instance that caused the current exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td><strong>Message</strong></td>
<td>Gets a message that describes the current exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td><strong>Source</strong></td>
<td>Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)</td>
</tr>
<tr>
<td><strong>StackTrace</strong></td>
<td>Gets a string representation of the frames on the call stack at the time the current exception was thrown. (Inherited from Exception.)</td>
</tr>
<tr>
<td><strong>TargetSite</strong></td>
<td>Gets the method that throws the current exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>
See Also

VersionMismatchException Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
VersionMismatchException Constructor

Initialize a new instance of the VersionMismatchException

Namespace: BerkeleyDB
**Syntax**

**C#**

```csharp
public VersionMismatchException()
```

**Visual Basic (Declaration)**

```vbnet
Public Sub New
```

**Visual C++**

```cpp
public:
VersionMismatchException()
```
See Also

VersionMismatchException Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `VersionMismatchException` type exposes the following members.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ErrorCode</strong></td>
<td>The underlying error code from the Berkeley DB C library. (Inherited from <strong>DatabaseException</strong>.)</td>
</tr>
</tbody>
</table>
See Also

VersionMismatchException Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `VersionMismatchException` type exposes the following members.
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Determines whether the specified <code>Object</code> is equal to the current <code>Object</code>. (Inherited from <code>Object</code>.) When overridden in a derived class, returns the <code>Exception</code> that is the root cause of one or more subsequent exceptions. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td><strong>GetBaseException</strong></td>
<td>Serves as a hash function for a particular type. (Inherited from <code>Object</code>.) When overridden in a derived class, sets the <code>SerializationInfo</code> with information about the exception. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Gets the runtime type of the current instance. (Inherited from <code>Exception</code>.) Creates and returns a string representation of the current exception. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td><strong>GetObjectData</strong></td>
<td></td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td></td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td></td>
</tr>
</tbody>
</table>
See Also

VersionMismatchException Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.
The `VersionMismatchException` type exposes the following members.
## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data</strong></td>
<td>Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td><strong>HelpLink</strong></td>
<td>Gets or sets a link to the help file associated with this exception.        (Inherited from Exception.)</td>
</tr>
<tr>
<td><strong>InnerException</strong></td>
<td>Gets the Exception instance that caused the current exception.              (Inherited from Exception.)</td>
</tr>
<tr>
<td><strong>Message</strong></td>
<td>Gets a message that describes the current exception.                        (Inherited from Exception.)</td>
</tr>
<tr>
<td><strong>Source</strong></td>
<td>Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)</td>
</tr>
<tr>
<td><strong>StackTrace</strong></td>
<td>Gets a string representation of the frames on the call stack at the time the current exception was thrown. (Inherited from Exception.)</td>
</tr>
<tr>
<td><strong>TargetSite</strong></td>
<td>Gets the method that throws the current exception.                          (Inherited from Exception.)</td>
</tr>
</tbody>
</table>
See Also

VersionMismatchException Class
BerkeleyDB Namespace

Report Feedback on this item in the Oracle Technology Network Forum

Copyright (c) 1996-2009 Oracle. All rights reserved.