A Sandcastle Documented Class Library
Vajhoej.Record Namespace
Library for reading and writing binary native structs.
Purpose

It is a library to read and write native structs from and to byte arrays by .NET programs in a way that is natural for .NET.

It's main purpose is to be used by the NISAM library.
Java port

The library is a port of a similar Java library, so an occasional Java'ism may be found.
Getting started

To use the library you need to:

- create POCO classes that contains all the fields in the native structs
- put annotations on the class and the fields that tell the library how the data are mapped to the native structs
- write the code that uses the library and the new POCO classes

Note:

- the POCO classes should have a no args constructor
- the library does not care whether you use private fields and public properties or public fields (the first is .NET best practice)
- all fields except static fields must have attributes defining reading/writing

Examples:

POCO class

```csharp
[Struct]
public class Data
{
    [StructField(N=0, Type=FieldType.INT4)]
    private int iv;
    [StructField(N=1, Type=FieldType.FP8)]
    private double xv;
    [StructField(N=2, Type=FieldType.FIXSTR, Length=8, Encoding="ISO")
    private String sv;
    public int Iv
    {
        get { return iv; }
        set { iv = value; }
    }
    public double Xv
    {
        get { return xv; }
        set { xv = value; }
    }
    public string Sv
```
The class got a [Struct] attribute. Each field in the class got a [StructField] attribute with an element N that determines the order of the fields and an element type that describes the datatype in the native struct plus some optional elements that are needed for some field types.

**Code fragments**

```csharp
// read
StructReader sr = new StructReader(somebytearray);
Data obj = sr.Read<Data>(typeof(Data));

// write
Data obj = new Data();
...
StructWriter sw = new StructWriter();
sw.Write(obj);
byte[] ba = sw.GetBytes();
```

**Arrays**

To specify a field as an array put an @Array annotation on the field and make the .NET type an array.

**Sub structs**

To specify a field as a sub struct just specify Type=FieldType.STRUCT in the [StructField] attribute.

**Polymorphism**

The Record library supports polymorphism in records.

You must use the [Selector] attribute on the last field in the super class.

```csharp
[Struct]
public class SuperData
```
```csharp
{
[StructField(N=0, Type=FieldType.INT4)]
private int id;
[StructField(N=1, Type=FieldType.INT4)]
[Selector]
[SubType(Value=1, Type=typeof(DataX))]
[SubType(Value=2, Type=typeof(DataY))]
private int typ;
public int Id
{
    get { return id; }
    set { id = value; }
}
public int Typ
{
    get { return typ; }
    set { typ = value; }
}
}

Usage is as simple as:

Data obj = sr.Read<Data>(typeof(Data));

This will actually read DataX or DataY instances depending on the value of the typ field.

Note that the numbering of the struct fields continue in the sub class - it does not reset to zero.
System requirements

- .NET 2.0 or newer
License

All the C# code is licensed under the Apache License, Version 2.0.
Author, bug reports etc.

This library is written by Arne Vajhoej (arne@vajhoej.dk).

Bug reports, bug fixes and comments in general are welcome.

Since this is a hobby project, then I can not give any guarantees about timeline for new releases.
Version history:

1.0
   Initial release (February 2011).

1.1
   More advanced length providers and some new struct types (May 2013).
# Types

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alignment</td>
<td>Enum Alignment specifies alignment within a native struct for arrays.</td>
</tr>
<tr>
<td>ArrayFieldAttribute</td>
<td>Annotation for arrays.</td>
</tr>
<tr>
<td>BCDUtil</td>
<td>Class TimeUtil converts between bytes with Binary Coded Decimals and BigDecimal objects.</td>
</tr>
<tr>
<td>ConvertSelector</td>
<td>Converts a selector of any type to a usable integer selector.</td>
</tr>
<tr>
<td>ElementsProvider</td>
<td>Get number of elements in array. Note: can only be used with struct fields that are arrays.</td>
</tr>
<tr>
<td>Endian</td>
<td>Enum Endian specifies endianess within a native struct.</td>
</tr>
<tr>
<td>FieldInfo</td>
<td>Class FieldInfo contains information about a native struct field needed for conversions.</td>
</tr>
<tr>
<td></td>
<td>Enum FieldType specifies native struct types.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Enum</th>
<th>Description</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>INT1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>INT2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>INT4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>INT8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UINT1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UINT2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UINT4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Code</td>
<td>Description</td>
<td>Length</td>
</tr>
<tr>
<td>----------</td>
<td>-------------------------------------------------------</td>
<td>--------</td>
</tr>
<tr>
<td>FP4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FP8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>INTX</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FIXSTR</td>
<td>Fixed length string</td>
<td>length</td>
</tr>
<tr>
<td>FIXSTRNULTERM</td>
<td>Fixed length string null terminated</td>
<td>length</td>
</tr>
<tr>
<td>VARSTR</td>
<td>Variable length string with 2 byte length prefix</td>
<td>length</td>
</tr>
<tr>
<td>FieldType</td>
<td>Description</td>
<td>Length</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
<td>--------</td>
</tr>
<tr>
<td>VARFIXSTR</td>
<td>Variable length string with 2 byte length prefix and padded to max length</td>
<td>length of string</td>
</tr>
<tr>
<td>REMSTR</td>
<td>Remaining data string</td>
<td></td>
</tr>
<tr>
<td>BOOLEAN</td>
<td>Boolean (0=false, other=true)</td>
<td>length of byte</td>
</tr>
<tr>
<td>BIT</td>
<td>Bits</td>
<td>length of bits used</td>
</tr>
<tr>
<td>JAVATIME</td>
<td>Binary time in Java format (milliseconds since 1-Jan-1970)</td>
<td></td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>---------------</td>
<td>-----------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>UNIXTIME</td>
<td>Binary time in Unix format (seconds since 1-Jan-1970)</td>
<td></td>
</tr>
<tr>
<td>VMSTIME</td>
<td>Binary time in VMS format (100 nanoseconds since 17-Nov-1858)</td>
<td></td>
</tr>
<tr>
<td>PACKEDBCD</td>
<td>Packed BCD (1 byte = 2 decimal digit nibbles)</td>
<td></td>
</tr>
<tr>
<td>ZONEDBCD</td>
<td>Zoned BCD (1 byte = 1 zone nibble + 1 decimal digit nibble)</td>
<td></td>
</tr>
<tr>
<td>VAXFP4</td>
<td>VAX F floating point</td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>----------------------</td>
<td></td>
</tr>
<tr>
<td>VAXFP8</td>
<td>VAX G floating point</td>
<td></td>
</tr>
<tr>
<td>STRUCT</td>
<td>Sub struct</td>
<td></td>
</tr>
</tbody>
</table>

**LengthProvider**
Get length of field. Note: can only be used with FIXSTRNULTERM, PACKEDBCD and ZONEDPBCD.

**MaxLengthProvider**
Get max length of struct. Note: can only be used with FIXSTR, FIXSTRNULTERM, PACKEDBCD.

**Util::ObjectHandlerProcess<Of <('T')>>**
Process object.

**Util2::ObjectHandlerProcess<Of <('T')>>**
Process object.

**RecordException**
Class RecordException encapsulates exceptions related to record processing.

**SelectorAttribute**
Annotation for selection of sub types.

**StructAttribute**
Annotation for structs.

**StructFieldAttribute**
Annotation for fields.

**StructInfo**
Class StructInfo contains information about structs and/or writing.

**StructInfoCache**
Class StructInfoCache caches StructInfo objects in a singleton cache.

**StructReader**
Class StructReader reads a .NET object from a byte array containing a native struct.

**StructWriter**
Class StructWriter writes a .NET object to a byte array as a native struct.

**SubClassAndPad**
Class SubClassAndPad contains information about class and padding for select field.

**SubTypeAttribute**
Annotation for sub types.

**TimeUtil**
Class TimeUtil converts between integers in various time formats and DateTime objects.

**Util::TransformerConvert<Of <('T1, T2')>>**
Convert object.

**Util2::TransformerConvert<Of <('T1, T2')>>**
Convert object.

**Util**
Utility class to process lists and to work with...
<table>
<thead>
<tr>
<th><strong>Util2</strong></th>
<th>Utility class to process lists and to work even with variable length structs.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>VAXFloatUtil</strong></td>
<td>Class VAXFloatUtil converts between VA</td>
</tr>
</tbody>
</table>
See Also
A Sandcastle Documented Class Library
Alignment Enumeration

Enum Alignment specifies alignment within native struct.

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
Syntax

C# Visual Basic Visual C++

public enum Alignment
Public Enumeration Alignment
public enum class Alignment
### Members

<table>
<thead>
<tr>
<th>Member</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PACKED</td>
<td>No padding.</td>
</tr>
<tr>
<td>NATURAL</td>
<td>Padding to natural alignment.</td>
</tr>
<tr>
<td>ALIGN1</td>
<td>Padding to multiple of 1 alignment (same as PACKED).</td>
</tr>
<tr>
<td>ALIGN2</td>
<td>Padding to multiple of 2 alignment.</td>
</tr>
<tr>
<td>ALIGN4</td>
<td>Padding to multiple of 4 alignment.</td>
</tr>
<tr>
<td>ALIGN8</td>
<td>Padding to multiple of 8 alignment.</td>
</tr>
</tbody>
</table>
See Also

Vajhoej.Record Namespace
A Sandcastle Documented Class Library
ArrayFieldAttribute Class
Annotation for arrays.

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
Syntax

C#   Visual Basic   Visual C++

[AttributeUsageAttribute(AttributeTargets.Field)]
public class ArrayFieldAttribute : Attribute

[AttributeUsageAttribute(AttributeTargets.Field)]
public ref class ArrayFieldAttribute : public Attribute
Members

All Members

Constructors

Properties

Methods

Explicit Interface Implementations

Public

Protected

Instance

Static

Declared

Inherited

XNA Framework Only

.NET Compact Framework Only

Member

Description

ArrayFieldAttribute QQQQ
Default constructor.

Elements
Number of elements in array.

Equals(Object)
Returns a value that indicates whether this instance is equal to a specified object.
(Inherited from Attribute.)

Finalize QQQQ
Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection.
(Inherited from Object.)

GetHashCode QQQQ
Returns the hash code for this instance.
(Inherited from Attribute.)

_GetIDsOfNames(Guid%, IntPtr, UInt32, UInt32, IntPtr)
Maps a set of names to a corresponding set of dispatch identifiers.
(Inherited from Attribute.)

GetType QQQQ
Gets the type of the current instance.
(Inherited from Object.)

GetTypeInfo(UInt32, UInt32, IntPtr)
Retrieves the type information for an object, which can be used to get the type...
information for an interface. (Inherited from \texttt{Attribute}.)

Retrieves the number of type information interfaces that an object provides (either 0 or 1). (Inherited from \texttt{Attribute}.)

Provides access to properties and methods exposed by an object. (Inherited from \texttt{Attribute}.)

When overridden in a derived class, indicates whether the value of this instance is the default value for the derived class. (Inherited from \texttt{Attribute}.)

When overridden in a derived class, returns a value that indicates whether this instance equals a specified object. (Inherited from \texttt{Attribute}.)

Creates a shallow copy of the current \texttt{Object}. (Inherited from \texttt{Object}.)

Returns a string that represents the current object. (Inherited from \texttt{Object}.)

When implemented in a derived class, gets a unique identifier for this \texttt{Attribute}. (Inherited from \texttt{Attribute}.)
Inheritance Hierarchy

System..::..Object
System..::..Attribute
Vajhoej.Record..::..ArrayFieldAttribute
See Also

Vajhoej.Record Namespace
Default constructor.

**Namespace:** [Vajhoej.Record](https://example.com)

**Assembly:** Record (in Record.dll) Version: 0.0.0.0
Syntax

C#       Visual Basic       Visual C++

public ArrayFieldAttribute()

Public Sub New

public:
ArrayFieldAttribute()
See Also

ArrayFieldAttribute Class
Vajhoej.Record Namespace
See Also
A Sandcastle Documented Class Library
ArrayFieldAttribute Methods

The ArrayFieldAttribute type exposes the following methods.
# Methods

<table>
<thead>
<tr>
<th>Public</th>
<th>Instance</th>
<th>Declared</th>
<th>XNA Framework Only</th>
</tr>
</thead>
<tbody>
<tr>
<td>Protected</td>
<td>Static</td>
<td>Inherited</td>
<td>.NET Compact Framework Only</td>
</tr>
</tbody>
</table>

## Description

<table>
<thead>
<tr>
<th>Member</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals(Object)</strong></td>
<td>Returns a value that indicates whether this instance is equal to a specified object. (Inherited from <code>Attribute</code>.)</td>
</tr>
<tr>
<td><strong>Finalize()</strong></td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>GetHashCode()</strong></td>
<td>Returns the hash code for this instance. (Inherited from <code>Attribute</code>.)</td>
</tr>
<tr>
<td><strong>_Attribute,::,GetIDsOfNames(Guid%, IntPtr, UInt32, UInt32, IntPtr)</strong></td>
<td>Maps a set of names to a corresponding set of dispatch identifiers. (Inherited from <code>Attribute</code>.)</td>
</tr>
<tr>
<td><strong>_Attribute,::,GetType()</strong></td>
<td>Gets the type of the current instance. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>_Attribute,::,GetTypeInfo(UInt32, UInt32, IntPtr)</strong></td>
<td>Retrieves the type information for an object, which can be used to get the type information for an interface. (Inherited from <code>Attribute</code>.)</td>
</tr>
<tr>
<td><strong>_Attribute,::,GetTypeInfoCount(UInt32%)</strong></td>
<td>Retrieves the number of type information interfaces that an object provides (either 0 or 1). (Inherited from <code>Attribute</code>.)</td>
</tr>
</tbody>
</table>
Provides access to properties and methods exposed by an object. 
(Inherited from Attribute.)

When overridden in a derived class, indicates whether the value of this instance is the default value for the derived class. 
(Inherited from Attribute.)

When overridden in a derived class, returns a value that indicates whether this instance equals a specified object. 
(Inherited from Attribute.)

Creates a shallow copy of the current Object. 
(Inherited from Object.)

Returns a string that represents the current object. 
(Inherited from Object.)
See Also

ArrayFieldAttribute Class
Vajhoej.Record Namespace
See Also

A Sandcastle Documented Class Library

ArrayFieldAttribute Properties

The `ArrayFieldAttribute` type exposes the following properties.
## Properties

- **Public**  
- **Instance**  
- **Declared**  
- **XNA Framework Only**

- **Protected**  
- **Static**  
- **Inherited**  
- **.NET Compact Framework Only**

<table>
<thead>
<tr>
<th>Member</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Elements</td>
<td>Number of elements in array.</td>
</tr>
<tr>
<td></td>
<td>When implemented in a derived class, gets a unique identifier for this Attribute.</td>
</tr>
<tr>
<td>TypId</td>
<td>(Inherited from Attribute.)</td>
</tr>
</tbody>
</table>
See Also

ArrayFieldAttribute Class
Vajhoej.Record Namespace
See Also
A Sandcastle Documented Class Library
ArrayFieldAttribute::Elements Property

Number of elements in array.

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
**Syntax**

C#  Visual Basic  Visual C++

```csharp
public int Elements { get; set; }
```

```vbnet
Public Property Elements As Integer
    Get
    Set
```

```cpp
public:
    property int Elements { 
        int get ();
        void set (int value);
    }
```

See Also

ArrayFieldAttribute Class
Vajhoej.Record Namespace
A Sandcastle Documented Class Library

**BCDUtil Class**

Class `TimeUtil` converts between bytes with Binary Coded Decimals and `BigDecimal` objects.

**Namespace:** `Vajhoej.Record`

**Assembly:** `Record (in Record.dll)` Version: 0.0.0.0
C#  Visual Basic  Visual C++

public class BCDUtil
Public Class BCDUtil
public ref class BCDUtil
# Members

<table>
<thead>
<tr>
<th>Member</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BCDUtil()</strong></td>
<td>Initializes a new instance of the BCDUtil class.</td>
</tr>
<tr>
<td><strong>ASCII</strong></td>
<td>ASCII zone nibble.</td>
</tr>
<tr>
<td><strong>DecodePackedBCD(array&lt;Byte&gt;[], Int32)</strong></td>
<td>Convert from packed BCD to decimal.</td>
</tr>
<tr>
<td><strong>DecodeZonedBCD(array&lt;Byte&gt;[], Byte, Int32)</strong></td>
<td>Convert from zoned BCD to decimal.</td>
</tr>
<tr>
<td><strong>EBCDIC</strong></td>
<td>EBCDIC zone nibble.</td>
</tr>
<tr>
<td><strong>EncodePackedBCD(Decimal, Int32)</strong></td>
<td>Convert from decimal to packed BCD.</td>
</tr>
<tr>
<td><strong>EncodeZonedBCD(Decimal, Byte, Int32)</strong></td>
<td>Convert from BigDecimal to zoned BCD.</td>
</tr>
<tr>
<td><strong>Equals(Object)</strong></td>
<td>Determines whether the specified Object is equal to the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>Finalize()</strong></td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (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>GetHashCode()</strong></td>
<td>Gets the type of the current</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>---------------------------------------------------------------</td>
</tr>
<tr>
<td><code>GetType()</code></td>
<td>Gets the type of the current instance.</td>
</tr>
<tr>
<td><code>MemberwiseClone()</code></td>
<td>Creates a shallow copy of the current <code>Object</code>.</td>
</tr>
<tr>
<td><code>ToString()</code></td>
<td>Returns a string that represents the current object.</td>
</tr>
<tr>
<td><code>ZERO</code></td>
<td>Zero zone nibble.</td>
</tr>
</tbody>
</table>
Inheritance Hierarchy

System...Object
Vajhoej.Record...BCDUtil
See Also

Vajhoej.Record Namespace
See Also
A Sandcastle Documented Class Library
BCDUtil Constructor
Initializes a new instance of the BCDUtil class

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
Syntax

C#   Visual Basic   Visual C++

public BCDUtil()

Public Sub New

public:
BCDUtil()
See Also

BCDUtil Class
Vajhoej.Record Namespace
See Also
A Sandcastle Documented Class Library

BCDUtil Fields

The **BCDUtil** type exposes the following fields.
**Fields**

- **Public**  
- **Instance**  
- **Declared**  
- **XNA Framework Only**

- **Protected**  
- **Static**  
- **Inherited**  
- **.NET Compact Framework Only**

<table>
<thead>
<tr>
<th>Member</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>s ASCII</td>
<td>ASCII zone nibble.</td>
</tr>
<tr>
<td>s EBCDIC</td>
<td>EBCDIC zone nibble.</td>
</tr>
<tr>
<td>s ZERO</td>
<td>Zero zone nibble.</td>
</tr>
</tbody>
</table>
See Also

BCDUil Class
Vajhoej.Record Namespace
See Also
A Sandcastle Documented Class Library
BCDUtil::ASCII Field
ASCII zone nibble.

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
Syntax

C#   Visual Basic   Visual C++

public const byte ASCII

Public Const ASCII As Byte

public:
literal unsigned char ASCII
See Also

BCDUtil Class
Vajhoej.Record Namespace
See Also
A Sandcastle Documented Class Library
BCDUtil::EBCDIC Field

EBCDIC zone nibble.

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
Syntax

C#    Visual Basic    Visual C++

public const byte EBCDIC

Public Const EBCDIC As Byte

public:
literal unsigned char EBCDIC
See Also

BCDUtil Class
Vajhoej.Record Namespace
C# | Visual Basic
| Visual C++

See Also
A Sandcastle Documented Class Library
BCDUtil::<ZERO Field
Zero zone nibble.

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
Syntax

C#   Visual Basic   Visual C++

public const byte ZERO

Public Const ZERO As Byte

public:
    literal unsigned char ZERO
See Also

BCDUtil Class
Vajhoej.Record Namespace
The **BCDUtil** type exposes the following methods.
# Methods

<table>
<thead>
<tr>
<th>Public</th>
<th>Instance</th>
<th>Declared</th>
<th>XNA Framework Only</th>
</tr>
</thead>
<tbody>
<tr>
<td>Protected</td>
<td>Static</td>
<td>Inherited</td>
<td>.NET Compact Framework Only</td>
</tr>
</tbody>
</table>

## Methods

<table>
<thead>
<tr>
<th>Member</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>DecodePackedBCD(array&lt;Byte&gt;[], Int32)</code></td>
<td>Convert from packed BCD to decimal.</td>
</tr>
<tr>
<td><code>DecodeZonedBCD(array&lt;Byte&gt;[], Byte, Int32)</code></td>
<td>Convert from zoned BCD to decimal.</td>
</tr>
<tr>
<td><code>EncodePackedBCD(Decimal, Int32, Int32)</code></td>
<td>Convert from decimal to packed BCD.</td>
</tr>
<tr>
<td><code>EncodeZonedBCD(Decimal, Byte, Int32, Int32)</code></td>
<td>Convert from BigDecimal to zoned BCD.</td>
</tr>
<tr>
<td><code>Equals(Object)</code></td>
<td>Determines whether the specified Object is equal to the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td><code>Finalize()</code></td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td><code>GetHashCode()</code></td>
<td>Serves as a hash function for a particular type. (Inherited from Object.)</td>
</tr>
<tr>
<td><code>GetType()</code></td>
<td>Gets the type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td><code>MemberwiseClone()</code></td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td><code>ToString()</code></td>
<td>Returns a string that represents the current object.</td>
</tr>
</tbody>
</table>
(Inherited from `Object`.)
See Also

BCDUtil Class
Vajhoej.Record Namespace
See Also
A Sandcastle Documented Class Library
BCDUtil::DecodePackedBCD Method
Convert from packed BCD to decimal.

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
### Syntax

C#  Visual Basic  Visual C++

```csharp
public static decimal DecodePackedBCD(
    byte[] b,
    int decimals
)
```

```vbnet
Public Shared Function DecodePackedBCD ( _
    b As Byte(), _
    decimals As Integer _
) As Decimal
```

```cpp
public:
static Decimal DecodePackedBCD(
    array<unsigned char>^ b,
    int decimals
)
```

**Parameters**

- `b`  
  array<Byte>[]()[[]][]
  Bytes with packed BCD.

- `decimals`  
  Int32
  Implied decimals.

**Return Value**

Decimal.
See Also

BCDUtil Class
Vajhoej.Record Namespace
See Also
A Sandcastle Documented Class Library
BCDUtil...::DecodeZonedBCD Method
Convert from zoned BCD to decimal.

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
Syntax

C#  Visual Basic  Visual C++

public static decimal DecodeZonedBCD(
    byte[] b,
    byte zone,
    int decimals
)

Public Shared Function DecodeZonedBCD ( _
    b As Byte(), _
    zone As Byte, _
    decimals As Integer _
) As Decimal

public:
static Decimal DecodeZonedBCD(
    array<unsigned char>^ b,
    unsigned char zone,
    int decimals
)

Parameters

b
array<Byte>[]][][
Bytes with zoned BCD.

zone
Byte
Zone nibble value.

decimals
Int32
Implied decimals.

Return Value
Decimal.
See Also

BCDUtil Class
Vajhoej.Record Namespace
See Also
A Sandcastle Documented Class Library
BCDUtil::EncodePackedBCD Method
Convert from decimal to packed BCD.

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
Syntax

C#  Visual Basic  Visual C++

public static byte[] EncodePackedBCD(
    decimal v,
    int decimals,
    int length
)

Public Shared Function EncodePackedBCD ( _
    v As Decimal, _
    decimals As Integer, _
    length As Integer _
) As Byte()

public:
static array<unsigned char>^ EncodePackedBCD(
    Decimal v,
    int decimals,
    int length
)

Parameters

v

    Decimal
    Decimal.

decimals

    Int32
    Implied decimals.

length

    Int32
    Length.

Return Value
Byte array with packed BCD.
See Also

BCDUtil Class
Vajhoej.Record Namespace
| See Also |
A Sandcastle Documented Class Library
BCDUtil...::EncodeZonedBCD Method
Convert from BigDecimal to zoned BCD.

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
Syntax

C#  Visual Basic  Visual C++

public static byte[] EncodeZonedBCD(
    decimal v,
    byte zone,
    int decimals,
    int length
)

Public Shared Function EncodeZonedBCD ( _
    v As Decimal, _
    zone As Byte, _
    decimals As Integer, _
    length As Integer _
) As Byte()

public:
static array<unsigned char>^ EncodeZonedBCD(
    Decimal v,
    unsigned char zone,
    int decimals,
    int length
)

Parameters

v

    Decimal
    Decimal.

zone

    Byte
    Zone nibble value.

decimals

    Int32
    Implied decimals.
length
  \texttt{Int32}
  \texttt{Length}.

\textbf{ReturnValue}

Byte array with zoned BCD.
See Also

BCDUtil Class
Vajhoej.Record Namespace
ConvertSelector Delegate

Converts a selector of any type to a usable integer selector.

**Namespace:** Vajhoej.Record

**Assembly:** Record (in Record.dll) Version: 0.0.0.0
**Syntax**

C#  Visual Basic  Visual C++

```csharp
public delegate Nullable<int> ConvertSelector(
    Object o
)
```

```vbnet
Public Delegate Function ConvertSelector ( _
    o As Object _
) As Nullable(Of Integer)
```

```cpp
public delegate Nullable<int> ConvertSelector(
    Object^ o
)
```

**Parameters**

- `o`  
  `Object`  
  Object (not completely initialized for read).

**Return Value**

Real selector (null indicates that it is to be ignored).
See Also

Vajhoej.Record Namespace
See Also

A Sandcastle Documented Class Library

ElementsProvider Delegate

Get number of elements in array. Note: can only be used with struct fields that are arrays.

Namespace: Vajhoej.Record

Assembly: Record (in Record.dll) Version: 0.0.0.0
Syntax

**C#  Visual Basic  Visual C++**

```csharp
public delegate int ElementsProvider(
    Object o,
    int n
)
```

```vbnet
Public Delegate Function ElementsProvider ( _
    o As Object, _
    n As Integer _
) As Integer
```

```cpp
public delegate int ElementsProvider(
    Object^ o,
    int n
)
```

**Parameters**

- **o**
  - `Object`
  - Object (not completely initialized for read).

- **n**
  - `Int32`
  - Field number.

**Return Value**

Elements (values < 0 indicates that value is to be ignored).
See Also

Vajhoej.Record Namespace
A Sandcastle Documented Class Library

Endian Enumeration

Enum Endian specifies endianess within native struct.

**Namespace:** Vajhoej.Record

**Assembly:** Record (in Record.dll) Version: 0.0.0.0
Syntax

C#  Visual Basic  Visual C++

public enum Endian
Public Enumeration Endian
public enum class Endian
## Members

<table>
<thead>
<tr>
<th>Member</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LITTLE</td>
<td>Little endian.</td>
</tr>
<tr>
<td>BIG</td>
<td>Big endian (alias network order).</td>
</tr>
</tbody>
</table>
See Also

Vajhoej.Record Namespace
A Sandcastle Documented Class Library

FieldInfo Class

Class FieldInfo contains information about a native struct field needed for conversions.

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
Syntax

C#  Visual Basic  Visual C++

public class FieldInfo
Public Class FieldInfo
public ref class FieldInfo
### Members

<table>
<thead>
<tr>
<th>FieldInfo(FieldType, Int32, Int32, String, Byte, Int32, Type, FieldInfo, IDictionary&lt;Of &lt;&lt;'(Int32, SubClassAndPad)&gt;&gt;, Boolean, Int32)</th>
<th>Create instance of FieldInfo with all necessary properties.</th>
</tr>
</thead>
<tbody>
<tr>
<td>ClassType</td>
<td>.NET class type.</td>
</tr>
<tr>
<td>Decimals</td>
<td>Decimals of BCD.</td>
</tr>
<tr>
<td>Elements</td>
<td>Number of elements.</td>
</tr>
<tr>
<td>Encoding</td>
<td>Encoding of string.</td>
</tr>
<tr>
<td>Equals(Object)</td>
<td>Determines whether the specified Object is equal to the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Field</td>
<td>Corresponding reflection object.</td>
</tr>
<tr>
<td>Finalize()</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (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>Length</td>
<td>Length of fixed length string.</td>
</tr>
</tbody>
</table>
**MemberwiseClone()**

Creates a shallow copy of the current `Object`. (Inherited from `Object`.)

**PrefixLength**

Prefix length.

**SelectPad**

Sub class padding to fixed length.

**Selects**

Sub class selections.

**StructType**

Struct type.

**ToString()**

Returns a string that represents the current object. (Inherited from `Object`.)

**Zone**

Zone of zoned BCD.
Inheritance Hierarchy

System..Object
Vajhoj.Record..FieldInfo
See Also

Vajhoej.Record Namespace
See Also
A Sandcastle Documented Class Library
FieldInfo Constructor

Create instance of FieldInfo with all necessary properties.

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
# Syntax

**C#**  **Visual Basic**  **Visual C++**

```csharp
public FieldInfo(
    FieldType structType,
    int length,
    int decimals,
    string encoding,
    byte zone,
    int prefixLength,
    Type classType,
    FieldInfo field,
    IDictionary<int, SubClassAndPad> selects,
    bool selectPad,
    int elements)
}
```

```vbnet
Public Sub New (_
    structType As FieldType, _
    length As Integer, _
    decimals As Integer, _
    encoding As String, _
    zone As Byte, _
    prefixLength As Integer, _
    classType As Type, _
    field As FieldInfo, _
    selects As IDictionary(Of Integer, SubClassAndPad), _
    selectPad As Boolean, _
    elements As Integer _
)
}
```

```cpp
public:
    FieldInfo(
        FieldType structType,
        int length,
        int decimals,
        String^ encoding,
        unsigned char zone,
        int prefixLength,
        Type^ classType,
        FieldInfo^ field,
```
IDictionary<int, SubClassAndPad> selects,
bool selectPad,
int elements
}

**Parameters**

**structType**

Field

native struct type.

**length**

Int32

length of fixed length string.

**decimals**

Int32

Number of decimals.

**encoding**

String

Encoding of string.

**zone**

Byte

Zone of zoned BCD.

**prefixLength**

Int32

Prefix length of variable length string.

**classType**

Type

.NET class type.

**field**

FieldInfo

Corresponding reflection object.

selects
IDictionary<Of (<'Int32, SubClassAndPad'>)> >
Sub class selections.

selectPad
  Boolean
Pad sub classes to fixed length.

elements
  Int32
Number of elements.
See Also

FieldInfo Class
Vajhoej.Record Namespace
The `FieldInfo` type exposes the following methods.
## Methods

<table>
<thead>
<tr>
<th>Public</th>
<th>Instance</th>
<th>Declared</th>
<th>XNA Framework Only</th>
</tr>
</thead>
<tbody>
<tr>
<td>Protected</td>
<td>Static</td>
<td>Inherited</td>
<td>.NET Compact Framework Only</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Member</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Equals(Object)</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>Finalize()</code></td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (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 type of the current instance. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>MemberwiseClone()</code></td>
<td>Creates a shallow copy of the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>ToString()</code></td>
<td>Returns a string that represents the current object. (Inherited from <code>Object</code>.)</td>
</tr>
</tbody>
</table>
See Also

FieldInfo Class
Vajhoej.Record Namespace
The `FieldInfo` type exposes the following properties.
## Properties

- **Public**  
  - Instance  
  - Declared  
  - XNA Framework Only

- **Protected**  
  - Static  
  - Inherited  
  - .NET Compact Framework Only

<table>
<thead>
<tr>
<th>Member</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ClassType</td>
<td>.NET class type.</td>
</tr>
<tr>
<td>Decimals</td>
<td>Decimals of BCD.</td>
</tr>
<tr>
<td>Elements</td>
<td>Number of elements.</td>
</tr>
<tr>
<td>Encoding</td>
<td>Encoding of string.</td>
</tr>
<tr>
<td>Field</td>
<td>Corresponding reflection object.</td>
</tr>
<tr>
<td>Length</td>
<td>Length of fixed length string.</td>
</tr>
<tr>
<td>PrefixLength</td>
<td>Prefix length.</td>
</tr>
<tr>
<td>SelectPad</td>
<td>Sub class padding to fixed length.</td>
</tr>
<tr>
<td>Selects</td>
<td>Sub class selections.</td>
</tr>
<tr>
<td>StructType</td>
<td>Struct type.</td>
</tr>
<tr>
<td>Zone</td>
<td>Zone of zoned BCD.</td>
</tr>
</tbody>
</table>
See Also

FieldInfo Class
Vajhoej.Record Namespace
See Also
A Sandcastle Documented Class Library
FieldInfo...::ClassType Property
.NET class type.

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
Syntax

C#  Visual Basic  Visual C++

public Type ClassType { get; }

Public ReadOnly Property ClassType As Type
Get

public:
property Type^ ClassType {
      Type^ get ();
}


See Also

FieldInfo Class
Vajhoej.Record Namespace
Decimals of BCD.

**Namespace:** [Vajhoej.Record](#)

**Assembly:** Record (in Record.dll) Version: 0.0.0.0
Syntax

C#   Visual Basic   Visual C++

public int Decimals { get; }

Public ReadOnly Property Decimals As Integer
    Get

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

See Also

FieldInfo Class
Vajhoej.Record Namespace
See Also
A Sandcastle Documented Class Library
FieldInfo...::Elements Property

Number of elements.

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
Syntax

C#  Visual Basic  Visual C++

public int Elements { get; }

Public ReadOnly Property Elements As Integer
    Get

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

See Also

FieldInfo Class
Vajhoej.Record Namespace
See Also
A Sandcastle Documented Class Library
FieldInfo::Encoding Property

Encoding of string.

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
Syntax

C#   Visual Basic   Visual C++

public string Encoding { get; }

Public ReadOnly Property Encoding As String
    Get

public:
property String^ Encoding {
    String^ get ();
}
See Also

FieldInfo Class
Vajhoej.Record Namespace
See Also
A Sandcastle Documented Class Library
FieldInfo::Field Property
Corresponding reflection object.

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
## Syntax

### C#  Visual Basic  Visual C++

```
public FieldInfo Field { get; }
Public ReadOnly Property Field As FieldInfo
    Get

public:
    property FieldInfo^ Field { 
        FieldInfo^ get();
    }
```
See Also

FieldInfo Class
Vajhoej.Record Namespace
See Also

A Sandcastle Documented Class Library
FieldInfo::<FieldInfo::Length Property

Length of fixed length string.

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
## Syntax

**C#**  Visual Basic  Visual C++

```csharp
public int Length { get; }
Public ReadOnly Property Length As Integer
  Get
  
public:
property int Length {
    int get ();
  }
```
See Also

FieldInfo Class
Vajhoej.Record Namespace
See Also
A Sandcastle Documented Class Library
FieldInfo...::PrefixLength Property

Prefix length.

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
C#  Visual Basic  Visual C++

public int PrefixLength { get; }

Public ReadOnly Property PrefixLength As Integer
    Get

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

See Also

FieldInfo Class
Vajhoej.Record Namespace
C# | Visual Basic
Visual C++

See Also
A Sandcastle Documented Class Library
FieldInfo::<SelectPad Property
Sub class padding to fixed length.

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
Syntax

C#    Visual Basic    Visual C++

public bool SelectPad { get; }

Public ReadOnly Property SelectPad As Boolean
Get

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


See Also

FieldInfo Class
Vajhoej.Record Namespace
See Also

A Sandcastle Documented Class Library

FieldInfo::Selects Property

Sub class selections.

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
Syntax

C#    Visual Basic    Visual C++

public IDictionary<int, SubClassAndPad> Selects { get; }

Public ReadOnly Property Selects As IDictionary(Of Integer, SubClass)
Get

public:
property IDictionary<int, SubClassAndPad> Selects {
            IDictionary<int, SubClassAndPad> get ();
}
See Also

FieldInfo Class
Vajhoej.Record Namespace
See Also
A Sandcastle Documented Class Library
FieldInfo::StructType Property

Struct type.

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
**Syntax**

C#   Visual Basic   Visual C++

```c#
public FieldType StructType { get; }
```

```vbnet
Public ReadOnly Property StructType As FieldType
    Get
```

```cpp
public:
    property FieldType StructType {
        FieldType get ();
    }
```

See Also

FieldInfo Class
Vajhoej.Record Namespace
See Also

A Sandcastle Documented Class Library

FieldInfo:::Zone Property

Zone of zoned BCD.

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
**Syntax**

C#   Visual Basic   Visual C++

```csharp
public byte Zone { get; }
```

```vbscript
Public ReadOnly Property Zone As Byte
Get
```

```cpp
public:
property unsigned char Zone {
    unsigned char get ();
}
```
See Also

FieldInfo Class
Vajhoej.Record Namespace
**FieldType Enumeration**

Enum FieldType specifies native struct types. Semantics:

<table>
<thead>
<tr>
<th>enum value</th>
<th>description</th>
<th>attributes</th>
<th>native implementation</th>
<th>.NET implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>INT1</td>
<td></td>
<td>8 bit signed integer</td>
<td>sbyte</td>
<td></td>
</tr>
<tr>
<td>INT2</td>
<td></td>
<td>16 bit signed integer</td>
<td>short</td>
<td></td>
</tr>
<tr>
<td>INT4</td>
<td></td>
<td>32 bit signed integer</td>
<td>int</td>
<td></td>
</tr>
<tr>
<td>INT8</td>
<td></td>
<td>64 bit signed integer</td>
<td>long</td>
<td></td>
</tr>
<tr>
<td>UINT1</td>
<td></td>
<td>8 bit unsigned integer</td>
<td>byte</td>
<td></td>
</tr>
<tr>
<td>UINT2</td>
<td></td>
<td>16 bit unsigned integer</td>
<td>ushort</td>
<td></td>
</tr>
<tr>
<td>UINT4</td>
<td></td>
<td>32 bit unsigned integer</td>
<td>uint</td>
<td></td>
</tr>
<tr>
<td>FP4</td>
<td></td>
<td>32 bit IEEE floating point</td>
<td>float</td>
<td></td>
</tr>
<tr>
<td>FP8</td>
<td></td>
<td>64 bit IEEE floating point</td>
<td>double</td>
<td></td>
</tr>
<tr>
<td>INTX</td>
<td>length= &lt;bytes used&gt;</td>
<td>bytes</td>
<td>ulong</td>
<td></td>
</tr>
<tr>
<td>Code</td>
<td>Description</td>
<td>Length</td>
<td>Encoding</td>
<td>Notes</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------------------------------------------</td>
<td>-----------------</td>
<td>------------------</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>FIXSTR</td>
<td>Fixed length string</td>
<td>&lt;length of string&gt;</td>
<td>&lt;encoding used&gt;</td>
<td>(default encoding is ISO-8859-1)</td>
</tr>
<tr>
<td>FIXSTRNULTERM</td>
<td>Fixed length string nul terminated</td>
<td>length=</td>
<td>&lt;encoding used&gt;</td>
<td>sequence of bytes with nul bytes added for write and stripped for read</td>
</tr>
<tr>
<td>VARSTR</td>
<td>Variable length string with 2 byte length prefix</td>
<td></td>
<td>&lt;encoding used&gt;</td>
<td>2 byte length + sequence of bytes</td>
</tr>
<tr>
<td>VARFIXSTR</td>
<td>Variable length string with 2 byte length prefix and padded to max</td>
<td></td>
<td>&lt;encoding used&gt;</td>
<td>2 byte length + sequence of bytes</td>
</tr>
<tr>
<td>Type</td>
<td>Description</td>
<td>Length</td>
<td>Encoding</td>
<td>Format</td>
</tr>
<tr>
<td>------------</td>
<td>----------------------------------------------------------</td>
<td>-----------------</td>
<td>----------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>REMSTR</td>
<td>Remaining data string</td>
<td>length</td>
<td>encoding=</td>
<td>sequence of bytes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>8859-1, max.</td>
<td>&lt;encoding used&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>length is 32767</td>
<td>(default encoding is ISO-8859-1, max. length is 32767)</td>
<td></td>
</tr>
<tr>
<td>BOOLEAN</td>
<td>Boolean (0=false, other=true)</td>
<td>length=</td>
<td>&lt;bytes used&gt;</td>
<td>bytes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>8859-1, max.</td>
<td>max. length is 32767</td>
<td>bool</td>
</tr>
<tr>
<td>BIT</td>
<td>Bits</td>
<td>length=</td>
<td>&lt;bits used&gt;</td>
<td>bytes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>8859-1, max.</td>
<td>max. bits is 32</td>
<td>int</td>
</tr>
<tr>
<td>JAVATIME</td>
<td>Binary time in Java format (milliseconds since 1-Jan-1970)</td>
<td>length=</td>
<td>64 bit integer</td>
<td>System.DateTime</td>
</tr>
<tr>
<td></td>
<td></td>
<td>8859-1, max.</td>
<td>max. length is 32767</td>
<td></td>
</tr>
<tr>
<td>UNIXTIME</td>
<td>Binary time in Unix format (seconds since 1-Jan-1970)</td>
<td>length=</td>
<td>32 bit integer</td>
<td>System.DateTime</td>
</tr>
<tr>
<td>VMSTIME</td>
<td>Binary time in VMS format (100 bytes)</td>
<td>length=</td>
<td>64 bit integer</td>
<td>System.DateTime</td>
</tr>
<tr>
<td>Namespace</td>
<td>Description</td>
<td>Type</td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>nanoseconds</td>
<td>nanoseconds since 17-Nov-1858</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PACKEDBCD</td>
<td>Packed BCD (1 byte = 2 decimal digit nibbles)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ZONEDBCD</td>
<td>Zoned BCD (1 byte = 1 zone nibble + 1 decimal digit nibble)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VAXFP4</td>
<td>VAX F floating point</td>
<td>float</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VAXFP8</td>
<td>VAX G floating point</td>
<td>double</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STRUCT</td>
<td>Sub struct</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Namespace: [Vajhoej.Record](#)
Assembly: Record (in Record.dll) Version: 0.0.0.0
Syntax

C#   Visual Basic   Visual C++

public enum FieldType
Public Enumeration FieldType
public enum class FieldType
## Members

<table>
<thead>
<tr>
<th>Member</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INT1</td>
<td>8 bit signed integer.</td>
</tr>
<tr>
<td>INT2</td>
<td>16 bit signed integer.</td>
</tr>
<tr>
<td>INT4</td>
<td>32 bit signed integer.</td>
</tr>
<tr>
<td>INT8</td>
<td>64 bit signed integer.</td>
</tr>
<tr>
<td>UINT1</td>
<td>8 bit unsigned integer.</td>
</tr>
<tr>
<td>UINT2</td>
<td>16 bit unsigned integer.</td>
</tr>
<tr>
<td>UINT4</td>
<td>32 bit unsigned integer.</td>
</tr>
<tr>
<td>FP4</td>
<td>32 bit IEEE floating point.</td>
</tr>
<tr>
<td>FP8</td>
<td>64 bit IEEE floating point.</td>
</tr>
<tr>
<td>INTX</td>
<td>8-56 bit integer (intended for 24, 40, 48 and 56 bits).</td>
</tr>
<tr>
<td>FIXSTR</td>
<td>Fixed length string.</td>
</tr>
<tr>
<td>FIXSTRNULTERM</td>
<td>Fixed length string nul terminated.</td>
</tr>
<tr>
<td>VARSTR</td>
<td>Variable length string with 2 byte length prefix.</td>
</tr>
<tr>
<td>VARFIXSTR</td>
<td>Variable length string with 2 byte length prefix and padded to max length.</td>
</tr>
<tr>
<td>REMSTR</td>
<td>Remaining data string.</td>
</tr>
<tr>
<td>BOOLEAN</td>
<td>Boolean.</td>
</tr>
<tr>
<td>BIT</td>
<td>Bits.</td>
</tr>
<tr>
<td>JAVATIME</td>
<td>Binary time in Java format.</td>
</tr>
<tr>
<td>UNIXTIME</td>
<td>Binary time in Unix format.</td>
</tr>
<tr>
<td>VMSTIME</td>
<td>Binary time in VMS format.</td>
</tr>
<tr>
<td>PACKEDBCD</td>
<td>Packed BCD.</td>
</tr>
<tr>
<td>ZONEDBCD</td>
<td>Zoned BCD.</td>
</tr>
<tr>
<td>VAXFP4</td>
<td>VAX F floating point.</td>
</tr>
<tr>
<td>VAXFP8</td>
<td>VAX G floating point.</td>
</tr>
<tr>
<td>STRUCT</td>
<td>Sub struct.</td>
</tr>
</tbody>
</table>
See Also

Vajhoej.Record Namespace
<table>
<thead>
<tr>
<th>See Also</th>
</tr>
</thead>
</table>
A Sandcastle Documented Class Library

**LengthProvider Delegate**

Get length of field. Note: can only be used with struct fields of field types FIXSTR, FIXSTRNULTERM, PACKEDBCD and ZONEDPBCD.

**Namespace:** Vajhoej.Record

**Assembly:** Record (in Record.dll) Version: 0.0.0.0
Syntax

C#   Visual Basic   Visual C++

public delegate int LengthProvider(
   Object o,
   int n
)

Public Delegate Function LengthProvider ( _
   o As Object, _
   n As Integer _
) As Integer

public delegate int LengthProvider(
   Object^ o,
   int n
)

Parameters

  o
   Object
   Object (not completely initialized for read).

  n
   Int32
   Field number.

Return Value

Length (values < 0 indicates that value is to be ignored).
See Also

Vajhoej.Record Namespace
C#  Visual Basic

Visual C++

See Also
A Sandcastle Documented Class Library
MaxLengthProvider Delegate
Get max length of struct. Note: can only be used with struct fields of field types FIXSTR, FIXSTRNULTERM, PACKEDBCD and ZONEDBCD.

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
Syntax

C#  Visual Basic  Visual C++

public delegate int MaxLengthProvider()
Public Delegate Function MaxLengthProvider As Integer
public delegate int MaxLengthProvider()

Return Value

MaxLength (values < 0 indicates that value is to be ignored).
See Also

Vajhoej.Record Namespace
RecordException Class

Class RecordException encapsulates exceptions related to record processing.

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
Syntax

C#  Visual Basic  Visual C++

public class RecordException : Exception

Public Class RecordException _
   Inherits Exception

public ref class RecordException : public Exception
# Members

All Members  Constructors  Properties  Methods  
Public  Instance  Declared  XNA Framework Only  
Protected  Static  Inherited  .NET Compact Framework Only  

<table>
<thead>
<tr>
<th>Member</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>RecordException()</code></td>
<td>Create instance of <code>RecordException</code>.</td>
</tr>
<tr>
<td><code>RecordException(Exception)</code></td>
<td>Create instance of <code>RecordException</code>.</td>
</tr>
<tr>
<td><code>RecordException(String)</code></td>
<td>Create instance of <code>RecordException</code>.</td>
</tr>
<tr>
<td><code>RecordException(String, Exception)</code></td>
<td>Create instance of <code>RecordException</code>.</td>
</tr>
</tbody>
</table>

**Data**

<table>
<thead>
<tr>
<th>Member</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Data</code></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>
</tbody>
</table>

**Equals(Object)**

<table>
<thead>
<tr>
<th>Member</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Equals(Object)</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>
</tbody>
</table>

**Finalize()**

<table>
<thead>
<tr>
<th>Member</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Finalize()</code></td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <code>Object</code>.)</td>
</tr>
</tbody>
</table>

**GetBaseException()**

<table>
<thead>
<tr>
<th>Member</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>GetBaseException()</code></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>
</tbody>
</table>

**GetHashCode()**

<table>
<thead>
<tr>
<th>Member</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>GetHashCode()</code></td>
<td>Serves as a hash function for a particular type. (Inherited from <code>Object</code>.)</td>
</tr>
</tbody>
</table>

**GetObjectData()**

<table>
<thead>
<tr>
<th>Member</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>GetObjectData()</code></td>
<td>When overridden in a derived class, sets the <code>SerializationInfo, SerializationInfo, StreamingContext</code> with information about the exception.</td>
</tr>
</tbody>
</table>
**Inherited from** Exception.

**GetType()**

Gets the runtime type of the current instance.

**HelpLink**

Gets or sets a link to the help file associated with this exception.

**HResult**

Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception.

**InnerException**

Gets the Exception instance that caused the current exception.

**MemberwiseClone()**

Creates a shallow copy of the current Object.

**Message**

Gets a message that describes the current exception.

**Source**

Gets or sets the name of the application or the object that causes the error.

**StackTrace**

Gets a string representation of the frames on the call stack at the time the current exception was thrown.

**TargetSite**

Gets the method that throws the current exception.

**ToString()**

Creates and returns a string representation of the current exception.
Inheritance Hierarchy

System..::..Object
System..::..Exception
Vajhoej.Record..::..RecordException
See Also

Vajhoej.Record Namespace
| See Also |
| A Sandcastle Documented Class Library
| RecordException Constructor |
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>RecordException()</code></td>
<td>Create instance of <code>RecordException</code>.</td>
</tr>
<tr>
<td><code>RecordException(Except)</code></td>
<td>Create instance of <code>RecordException</code>.</td>
</tr>
<tr>
<td><code>RecordException(String)</code></td>
<td>Create instance of <code>RecordException</code>.</td>
</tr>
<tr>
<td><code>RecordException(String, </code>Exception)`</td>
<td>Create instance of <code>RecordException</code>.</td>
</tr>
</tbody>
</table>
See Also

RecordException Class
Vajhoej.Record Namespace
| See Also
A Sandcastle Documented Class Library
RecordException Constructor
Create instance of RecordException.

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
C#    Visual Basic    Visual C++

public RecordException()

Public Sub New

public:
RecordException()
See Also

RecordException Class
RecordException Overload
Vajhoej.Record Namespace
See Also
A Sandcastle Documented Class Library
RecordException Constructor (Exception)
Create instance of RecordException.

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
Syntax

C#   Visual Basic   Visual C++

```csharp
public RecordException(
    Exception cause
)
```

```vbnet
Public Sub New (_
    cause As Exception _
)
```

```cpp
public:
RecordException(
    Exception^ cause
)
```

**Parameters**

cause

`Exception`

Underlying exception.
See Also

RecordException Class
RecordException Overload
Vajhoej.Record Namespace
See Also

A Sandcastle Documented Class Library

RecordException Constructor (String)

Create instance of RecordException.

**Namespace:** Vajhoej.Record

**Assembly:** Record (in Record.dll) Version: 0.0.0.0
Syntax

C#   Visual Basic   Visual C++

public RecordException(
    string message
)

Public Sub New (_
    message As String _
)

public:
RecordException(
    String^ message
)

Parameters

message
    String
    Message describing exception.
See Also

RecordException Class
RecordException Overload
Vajhoej.Record Namespace
See Also
A Sandcastle Documented Class Library
RecordException Constructor (String, Exception)
Create instance of RecordException.

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
## Syntax

C#  Visual Basic  Visual C++

```csharp
public RecordException(
    string message,
    Exception cause
)

Public Sub New (
    message As String,
    cause As Exception
)

public: RecordException(
    String^ message,
    Exception^ cause
)
```

### Parameters

message  

- **String**  
  Message describing exception.

cause  

- **Exception**  
  Underlying exception.
See Also

RecordException Class
RecordException Overload
Vajhoej.Record Namespace
A Sandcastle Documented Class Library

RecordException Methods

The `RecordException` type exposes the following methods.
## Methods

<table>
<thead>
<tr>
<th>Public</th>
<th>Instance</th>
<th>Declared</th>
<th>XNA Framework Only</th>
</tr>
</thead>
<tbody>
<tr>
<td>Protected</td>
<td>Static</td>
<td>Inherited</td>
<td>.NET Compact Framework Only</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Member</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Equals(Object)</code></td>
<td>Determines whether the specified Object is equal to the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td><code>Finalize()</code></td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td><code>GetBaseException()</code></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><code>GetHashCode()</code></td>
<td>Serves as a hash function for a particular type. (Inherited from Object.)</td>
</tr>
<tr>
<td><code>GetObjectData(SerializationInfo, StreamingContext)</code></td>
<td>When overridden in a derived class, sets the SerializationInfo with information about the exception. (Inherited from Exception.)</td>
</tr>
<tr>
<td><code>GetType()</code></td>
<td>Gets the runtime type of the current instance. (Inherited from Exception.)</td>
</tr>
<tr>
<td><code>MemberwiseClone()</code></td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td><code>ToString()</code></td>
<td>Creates and returns a string representation of the current exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>
See Also

RecordException Class
Vajhoej.Record Namespace
The `RecordException` type exposes the following properties.
**Properties**

<table>
<thead>
<tr>
<th>Public</th>
<th>Instance</th>
<th>Declared</th>
<th>XNA Framework Only</th>
</tr>
</thead>
<tbody>
<tr>
<td>Protected</td>
<td>Static</td>
<td>Inherited</td>
<td>.NET Compact Framework Only</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Member</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>HResult</strong></td>
<td>Gets or sets HRESULT, a coded numerical value that is assigned to a specific 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

RecordException Class
Vajhoej.Record Namespace
SelectorAttribute Class

Annotation for selection of sub types.

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
Syntax

C#   Visual Basic   Visual C++

[AttributeUsageAttribute(AttributeTargets.Field)]
public class SelectorAttribute : Attribute

<AttributeUsageAttribute(AttributeTargets.Field)> _
Public Class SelectorAttribute _
    Inherits Attribute

[AttributeUsageAttribute(AttributeTargets::Field)]
public ref class SelectorAttribute : public Attribute
### Members

- **All Members**
- **Constructors**
- **Properties**
- **Methods**
- **Explicit Interface Implementations**

- **Public**
- **Instance**
- **Declared**
- **XNA Framework Only**

- **Protected**
- **Static**
- **Inherited**
- **.NET Compact Framework Only**

<table>
<thead>
<tr>
<th>Member</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SelectorAttribute()</strong></td>
<td>Default constructor. Returns a value that indicates whether this instance is equal to a specified object. (Inherited from <strong>Attribute</strong>.)</td>
</tr>
<tr>
<td><strong>Equals(Object)</strong></td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>Finalize()</strong></td>
<td>Returns the hash code for this instance. (Inherited from <strong>Attribute</strong>.)</td>
</tr>
<tr>
<td><strong>GetHashCode()</strong></td>
<td>Maps a set of names to a corresponding set of dispatch identifiers. (Inherited from <strong>Attribute</strong>.)</td>
</tr>
<tr>
<td><strong>.GetType()</strong></td>
<td>Gets the type of the current instance. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>_Attribute, , , GetIDsOfNames(Guid%, IntPtr, UInt32, UInt32, IntPtr)</strong></td>
<td>Retrieves the type information for an object, which can be used to get the type information for an interface. (Inherited from <strong>Attribute</strong>.)</td>
</tr>
</tbody>
</table>
_Attribute...::..GetTypeInfoCount(UInt32%)

Retrieves the number of type information interfaces that an object provides (either 0 or 1).
(Inherited from Attribute.)

_Attribute...::..Invoke(UInt32, Guid%, UInt32, Int16, IntPtr, IntPtr, IntPtr, IntPtr)

Provides access to properties and methods exposed by an object.
(Inherited from Attribute.)

IsDefaultAttribute()()()()

When overridden in a derived class, indicates whether the value of this instance is the default value for the derived class.
(Inherited from Attribute.)

Match(Object)

When overridden in a derived class, returns a value that indicates whether this instance equals a specified object.
(Inherited from Attribute.)

MemberwiseClone()()()

Creates a shallow copy of the current Object.
(Inherited from Object.)

Pad

Pad all sub types to same length.

Subtypes

Available sub types.

ToString()()()()

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

When implemented in a derived class, gets a unique identifier for this Attribute.
(Inherited from Attribute.)

TypeId

When overridden from Object, gets a unique identifier for this object.
(Inherited from Object.)
Inheritance Hierarchy

System..::..Object
System..::..Attribute
Vajhoej.Record..::..SelectorAttribute
See Also

Vajhoej.Record Namespace
See Also
A Sandcastle Documented Class Library
SelectorAttribute Constructor

Default constructor.

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
Syntax

C#   Visual Basic   Visual C++

public SelectorAttribute()

Public Sub New

public:
SelectorAttribute()
See Also

SelectorAttribute Class
Vajhoej.Record Namespace
See Also

A Sandcastle Documented Class Library

SelectorAttribute Methods

The SelectorAttribute type exposes the following methods.
# Methods

<table>
<thead>
<tr>
<th>Public</th>
<th>Instance</th>
<th>Declared</th>
<th>XNA Framework Only</th>
</tr>
</thead>
<tbody>
<tr>
<td>Protected</td>
<td>Static</td>
<td>Inherited</td>
<td>.NET Compact Framework Only</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Member</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals(Object)</strong></td>
<td>Returns a value that indicates whether this instance is equal to a specified object. (Inherited from Attribute.)</td>
</tr>
<tr>
<td><strong>Finalize()</strong></td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>GetHashCode()</strong></td>
<td>Returns the hash code for this instance. (Inherited from Attribute.)</td>
</tr>
<tr>
<td><strong>Attribute...:GetIDsOfNames(Guid%, IntPtr, UInt32, UInt32, IntPtr)</strong></td>
<td>Maps a set of names to a corresponding set of dispatch identifiers. (Inherited from Attribute.)</td>
</tr>
<tr>
<td><strong>GetType()</strong></td>
<td>Gets the type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>Attribute...:GetTypeInfo(UInt32, UInt32, IntPtr)</strong></td>
<td>Retrieves the type information for an object, which can be used to get the type information for an interface. (Inherited from Attribute.)</td>
</tr>
<tr>
<td><strong>Attribute...:GetTypeInfoCount(UInt32)</strong></td>
<td>Retrieves the number of type information interfaces that an object provides (either 0 or 1). (Inherited from Attribute.)</td>
</tr>
</tbody>
</table>
**Invoke(UInt32, Guid%, UInt32, Int16, IntPtr, IntPtr, IntPtr, IntPtr)**

Provides access to properties and methods exposed by an object.
(Inherited from [Attribute](#).)

When overridden in a derived class, indicates whether the value of this instance is the default value for the derived class.
(Inherited from [Attribute](#).)

**IsDefaultAttribute()**

When overridden in a derived class, indicates whether this instance equals a specified object.
(Inherited from [Attribute](#).)

**Match(Object)**

**MemberwiseClone()**

Creates a shallow copy of the current [Object](#).
(Inherited from [Object](#).)

**ToString()**

Returns a string that represents the current object.
(Inherited from [Object](#).)
See Also

SelectorAttribute Class
Vajhoej.Record Namespace
See Also
A Sandcastle Documented Class Library SelectorAttribute Properties

The SelectorAttribute type exposes the following properties.
**Properties**

- Public  ✅ Instance  ✅ Declared  ✅ XNA Framework Only
- Protected  ✅ Static  ✅ Inherited  ✅ .NET Compact Framework Only

<table>
<thead>
<tr>
<th>Member</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>✅ Pad</td>
<td>Pad all sub types to same length.</td>
</tr>
<tr>
<td>✅ Subtypes</td>
<td>Available sub types.</td>
</tr>
<tr>
<td>✅ Typel</td>
<td>When implemented in a derived class, gets a unique identifier for this <a href="#">Attribute</a>.</td>
</tr>
</tbody>
</table>

(Inherited from [Attribute](#).)
See Also

SelectorAttribute Class
Vajhoej.Record Namespace
See Also
A Sandcastle Documented Class Library
SelectorAttribute...::Pad Property
Pad all sub types to same length.

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
Syntax

C#   Visual Basic   Visual C++

public bool Pad { get; set; }

Public Property Pad As Boolean
    Get
    Set

public:
property bool Pad {
    bool get ();
    void set (bool value);
}
See Also

SelectorAttribute Class
Vajhoej.Record Namespace
See Also
A Sandcastle Documented Class Library SelectorAttribute...::Subtypes Property
Available sub types.

**Namespace:** Vajhoej.Record
**Assembly:** Record (in Record.dll) Version: 0.0.0.0
Syntax

C#    Visual Basic    Visual C++

public  Object[]  Subtypes {  get;  set;  }

Public  Property  Subtypes  As  Object()
  Get
  Set

public:
property array<Object^>^  Subtypes {
  array<Object^>^  get ();
  void  set (array<Object^>^  value);
}
See Also

SelectorAttribute Class
Vajhoej.Record Namespace
A Sandcastle Documented Class Library
StructAttribute Class

Annotation for structs.

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
Syntax

C#    Visual Basic    Visual C++

[AttributeUsageAttribute(AttributeTargets.Class)]
public class StructAttribute : Attribute

<AttributeUsageAttribute(AttributeTargets.Class)> _
Public Class StructAttribute _
    Inherits Attribute

[AttributeUsageAttribute(AttributeTargets::Class)]
public ref class StructAttribute : public Attribute
## Members

### StructAttribute

**Description:**
Default constructor.

### Alignment

**Description:**
Alignment. Default is packed.

### Endianess

**Description:**
Byte order. Default is little endian.

### Endpad

**Description:**
End padding. Default is false.

### Equals(Object)

**Description:**
Returns a value that indicates whether this instance is equal to a specified object.

(Inherited from [Attribute](#).)

### Finalize

**Description:**
Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection.

(Inherited from [Object](#).)

### GetHashCode

**Description:**
Returns the hash code for this instance.

(Inherited from [Attribute](#).)

### GetIDsOfNames(Guid%, IntPtr, UInt32, UInt32, IntPtr)

**Description:**
Maps a set of names to a corresponding set of dispatch identifiers.

(Inherited from [Attribute](#).)

### GetType

**Description:**
Gets the type of the current instance.

(Inherited from [Object](#).)
Retrieve the type information for an object, which can be used to get the type information for an interface. (Inherited from `Attribute`.)

Retrieve the number of type information interfaces that an object provides (either 0 or 1). (Inherited from `Attribute`.)

Provides access to properties and methods exposed by an object. (Inherited from `Attribute`.)

When overridden in a derived class, indicates whether the value of this instance is the default value for the derived class. (Inherited from `Attribute`.)

When overridden in a derived class, returns a value that indicates whether this instance equals a specified object. (Inherited from `Attribute`.)

Creates a shallow copy of the current `Object`. (Inherited from `Object`.)

Returns a string that represents the current object. (Inherited from `Object`.)

When implemented in a derived class, gets a unique identifier for this `Attribute`. (Inherited from `Attribute`.)
Inheritance Hierarchy

System..::..Object
System..::..Attribute
Vajhoej.Record..::..StructAttribute
See Also

Vajhoej.Record Namespace
See Also
A Sandcastle Documented Class Library
StructAttribute Constructor
Default constructor.

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
Syntax

C#    Visual Basic    Visual C++

public StructAttribute()

Public Sub New

public:
StructAttribute()
See Also

StructAttribute Class
Vajhoej.Record Namespace
The `StructAttribute` type exposes the following methods.
## Methods

<table>
<thead>
<tr>
<th>Public</th>
<th>Instance</th>
<th>Declared</th>
<th>XNA Framework Only</th>
</tr>
</thead>
<tbody>
<tr>
<td>Protected</td>
<td>Static</td>
<td>Inherited</td>
<td>.NET Compact Framework Only</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Member</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals(Object)</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Finalize()</strong></td>
<td></td>
</tr>
<tr>
<td><strong>GetHashCode()</strong></td>
<td></td>
</tr>
<tr>
<td><strong>_Attribute::..:GetIDsOfNames(Guid%, IntPtr, UInt32, UInt32, IntPtr)</strong></td>
<td></td>
</tr>
<tr>
<td><strong>GetTypeInfo(UInt32, UInt32, IntPtr)</strong></td>
<td></td>
</tr>
<tr>
<td><strong>GetTypeInfo(UInt32 %)</strong></td>
<td></td>
</tr>
<tr>
<td><strong>_Attribute::..:GetTypeInfoCount(UInt32 %)</strong></td>
<td></td>
</tr>
</tbody>
</table>

- Returns a value that indicates whether this instance is equal to a specified object. (Inherited from Attribute.)
- Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
- Returns the hash code for this instance. (Inherited from Attribute.)
- Maps a set of names to a corresponding set of dispatch identifiers. (Inherited from Attribute.)
- Gets the type of the current instance. (Inherited from Object.)
- Retrieves the type information for an object, which can be used to get the type information for an interface. (Inherited from Attribute.)
- Retrieves the number of type information interfaces that an object provides (either 0 or 1). (Inherited from Attribute.)
Invoke(UInt32, Guid%, UInt32, Int16, IntPtr, IntPtr, IntPtr, IntPtr) Provides access to properties and methods exposed by an object. (Inherited from Attribute.)

IsDefaultAttribute() When overridden in a derived class, indicates whether the value of this instance is the default value for the derived class. (Inherited from Attribute.)

Match(Object) When overridden in a derived class, returns a value that indicates whether this instance equals a specified object. (Inherited from Attribute.)

MemberwiseClone() Creates a shallow copy of the current Object. (Inherited from Object.)

ToString() Returns a string that represents the current object. (Inherited from Object.)
See Also

StructAttribute Class
Vajhoej.Record Namespace
The `StructAttribute` type exposes the following properties.
### Properties

<table>
<thead>
<tr>
<th>Public</th>
<th>Instance</th>
<th>Declared</th>
<th>XNA Framework Only</th>
</tr>
</thead>
<tbody>
<tr>
<td>Protected</td>
<td>Static</td>
<td>Inherited</td>
<td>.NET Compact Framework Only</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Member</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Alignment" /> <strong>Alignment</strong></td>
<td>Alignment. Default is packed.</td>
</tr>
<tr>
<td><img src="image" alt="Endianess" /> <strong>Endianess</strong></td>
<td>Byte order. Default is little endian.</td>
</tr>
<tr>
<td><img src="image" alt="Endpad" /> <strong>Endpad</strong></td>
<td>End padding. Default is false.</td>
</tr>
<tr>
<td><img src="image" alt="TypeId" /> <strong>TypeId</strong></td>
<td>When implemented in a derived class, gets a unique identifier for this <strong>Attribute</strong>.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from <strong>Attribute</strong>.)</td>
</tr>
</tbody>
</table>
See Also

StructAttribute Class
Vajhoej.Record Namespace
See Also
A Sandcastle Documented Class Library
StructAttribute...::Alignment Property
Alignment. Default is packed.

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
- Syntax

C#   Visual Basic   Visual C++

public Alignment Alignment { get; set; }

Public Property Alignment As Alignment
Get
Set

public:
property Alignment Alignment {
    Alignment get ();
    void set (Alignment value);
}
See Also

StructAttribute Class
Vajhoej.Record Namespace
See Also

A Sandcastle Documented Class Library

StructAttribute...::Endianess Property

Byte order. Default is little endian.

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
Syntax

C#   Visual Basic   Visual C++

public Endian Endianess { get; set; }

Public Property Endianess As Endian
    Get
        Endian
    Set
        Endian value;

public:
    property Endian Endianess {
        Endian get ();
        void set (Endian value);
    }
See Also

StructAttribute Class
Vajhoej.Record Namespace
End padding. Default is false.

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
**Syntax**

C#     Visual Basic     Visual C++

```csharp
public bool Endpad { get; set; }

Public Property Endpad As Boolean
    Get
    Set

public:
property bool Endpad {
    bool get ();
    void set (bool value);
}
```
See Also

StructAttribute Class
Vajhoej.Record Namespace
A Sandcastle Documented Class Library

StructFieldAttribute Class

Annotation for fields.

**Namespace:** [Vajhoej.Record](#)

**Assembly:** Record (in Record.dll) Version: 0.0.0.0
**Syntax**

C#     Visual Basic     Visual C++

```csharp
[AttributeUsageAttribute(AttributeTargets.Field)]
public class StructFieldAttribute : Attribute
```

```vbnet
<AttributeUsageAttribute(AttributeTargets.Field)> _
Public Class StructFieldAttribute _
   Inherits Attribute
```

```cpp
[AttributeUsageAttribute(AttributeTargets::Field)]
public ref class StructFieldAttribute : public Attribute
```
# Members

<table>
<thead>
<tr>
<th>All Members</th>
<th>Constructors</th>
<th>Properties</th>
<th>Methods</th>
<th>Explicit Interface Implementations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Public</td>
<td>Instance</td>
<td>Declared</td>
<td>XNA Framework Only</td>
<td></td>
</tr>
</tbody>
</table>

| Protected   | Static       | Inherited | .NET Compact Framework Only |

## Member

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default constructor.</td>
</tr>
<tr>
<td>Field decimals (for BCD's).</td>
</tr>
<tr>
<td>Field encoding (for strings).</td>
</tr>
<tr>
<td>Returns a value that indicates whether this instance is equal to a specified object. (Inherited from <strong>Attribute</strong>.)</td>
</tr>
<tr>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td>Returns the hash code for this instance. (Inherited from <strong>Attribute</strong>.)</td>
</tr>
<tr>
<td>Maps a set of names to a corresponding set of dispatch identifiers. (Inherited from <strong>Attribute</strong>.)</td>
</tr>
<tr>
<td>Gets the type of the current instance. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td>Retrieves the type information for an object, which can be</td>
</tr>
</tbody>
</table>
**Attribute**::

_GetTypeInfo(UInt32, UInt32, IntPtr)_

Used to get the type information for an interface. (Inherited from **Attribute**.)

_Returns the number of type information interfaces that an object provides (either 0 or 1). (Inherited from **Attribute**.)

_PrefixLength_ (for variable length strings).

_Returns a string that represents the current object. (Inherited from **Object**.)

_Field type. When implemented in a derived class, gets a unique_
| **TypeId** | identifier for this [Attribute](#). (Inherited from [Attribute](#).) |
| **Zone** | Field zone value (for zoned BCD's). |
Inheritance Hierarchy

System..::..Object
System..::..Attribute
Vajhoj.Record..::..StructFieldAttribute
See Also

Vajhoej.Record Namespace
See Also
A Sandcastle Documented Class Library
StructFieldAttribute Constructor
Default constructor.

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
Syntax

C#    Visual Basic    Visual C++

public StructFieldAttribute()

Public Sub New

public:
StructFieldAttribute()
See Also

StructFieldAttribute Class
Vajhoej.Record Namespace
The `StructFieldAttribute` type exposes the following methods.
## Methods

<table>
<thead>
<tr>
<th>Member</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="https://img.shields.io/badge/Member-Public-0077B6" alt="Public" /> <img src="https://img.shields.io/badge/Public-0077B6" alt="Public" /> <img src="https://img.shields.io/badge/Instance-00B578" alt="Instance" /> <img src="https://img.shields.io/badge/Declared-00B578" alt="Declared" /> <img src="https://img.shields.io/badge/XNA%20Framework%20Only-008000" alt="XNA Framework Only" /></td>
<td>Returns a value that indicates whether this instance is equal to a specified object. (Inherited from Attribute.)</td>
</tr>
<tr>
<td><img src="https://img.shields.io/badge/Member-Public-0077B6" alt="Public" /> <img src="https://img.shields.io/badge/Public-0077B6" alt="Public" /> <img src="https://img.shields.io/badge/Instance-00B578" alt="Instance" /> <img src="https://img.shields.io/badge/Declared-00B578" alt="Declared" /> <img src="https://img.shields.io/badge/XNA%20Framework%20Only-008000" alt="XNA Framework Only" /></td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td><img src="https://img.shields.io/badge/Member-Public-0077B6" alt="Public" /> <img src="https://img.shields.io/badge/Public-0077B6" alt="Public" /> <img src="https://img.shields.io/badge/Instance-00B578" alt="Instance" /> <img src="https://img.shields.io/badge/Declared-00B578" alt="Declared" /> <img src="https://img.shields.io/badge/XNA%20Framework%20Only-008000" alt="XNA Framework Only" /></td>
<td>Returns the hash code for this instance. (Inherited from Object.)</td>
</tr>
<tr>
<td><img src="https://img.shields.io/badge/Member-Public-0077B6" alt="Public" /> <img src="https://img.shields.io/badge/Protected-0077B6" alt="Protected" /> <img src="https://img.shields.io/badge/Static-0077B6" alt="Static" /> <img src="https://img.shields.io/badge/Inherited-0077B6" alt="Inherited" /> <img src="https://img.shields.io/badge/.NET%20Compact%20Framework%20Only-0077B6" alt=".NET Compact Framework Only" /></td>
<td>Maps a set of names to a corresponding set of dispatch identifiers. (Inherited from Attribute.)</td>
</tr>
<tr>
<td><img src="https://img.shields.io/badge/Member-Public-0077B6" alt="Public" /> <img src="https://img.shields.io/badge/Public-0077B6" alt="Public" /> <img src="https://img.shields.io/badge/Instance-00B578" alt="Instance" /> <img src="https://img.shields.io/badge/Declared-00B578" alt="Declared" /> <img src="https://img.shields.io/badge/XNA%20Framework%20Only-008000" alt="XNA Framework Only" /></td>
<td>Gets the type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td><img src="https://img.shields.io/badge/Member-Public-0077B6" alt="Public" /> <img src="https://img.shields.io/badge/Public-0077B6" alt="Public" /> <img src="https://img.shields.io/badge/Instance-00B578" alt="Instance" /> <img src="https://img.shields.io/badge/Declared-00B578" alt="Declared" /> <img src="https://img.shields.io/badge/XNA%20Framework%20Only-008000" alt="XNA Framework Only" /></td>
<td>Retrieves the type information for an object, which can be used to get the type information for an interface. (Inherited from Attribute.)</td>
</tr>
<tr>
<td><img src="https://img.shields.io/badge/Member-Public-0077B6" alt="Public" /> <img src="https://img.shields.io/badge/Public-0077B6" alt="Public" /> <img src="https://img.shields.io/badge/Instance-00B578" alt="Instance" /> <img src="https://img.shields.io/badge/Declared-00B578" alt="Declared" /> <img src="https://img.shields.io/badge/XNA%20Framework%20Only-008000" alt="XNA Framework Only" /></td>
<td>Retrieves the number of type information interfaces that an object provides (either 0 or 1). (Inherited from Attribute.)</td>
</tr>
</tbody>
</table>
**Attribute::Invoke(UInt32, Guid%, UInt32, Int16, IntPtr, IntPtr, IntPtr, IntPtr, IntPtr)**

Provides access to properties and methods exposed by an object.
(Inherited from Attribute.)

When overridden in a derived class, indicates whether the value of this instance is the default value for the derived class.
(Inherited from Attribute.)

**IsDefaultAttribute()**

When overridden in a derived class, indicates whether this instance equals a specified object.
(Inherited from Attribute.)

**Match(Object)**

**MemberwiseClone()**

Creates a shallow copy of the current Object.
(Inherited from Object.)

**ToString()**

Returns a string that represents the current object.
(Inherited from Object.)
See Also

StructFieldAttribute Class
Vajhoej.Record Namespace
The `StructFieldAttribute` type exposes the following properties.
## Properties

- **Public**
- **Instance**
- **Declared**
- **XNA Framework Only**

- **Protected**
- **Static**
- **Inherited**
- **.NET Compact Framework Only**

<table>
<thead>
<tr>
<th>Member</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decimals</td>
<td>Field decimals (for BCD's).</td>
</tr>
<tr>
<td>Encoding</td>
<td>Field encoding (for strings).</td>
</tr>
<tr>
<td>Length</td>
<td>Field length (for fixed length strings and BCD's).</td>
</tr>
<tr>
<td>N</td>
<td>Field number.</td>
</tr>
<tr>
<td>PrefixLength</td>
<td>Prefix length (for variable length strings).</td>
</tr>
<tr>
<td>Type</td>
<td>Field type.</td>
</tr>
</tbody>
</table>

- When implemented in a derived class, gets a unique identifier for this **Attribute**.
- (Inherited from **Attribute**.)

| Zone           | Field zone value (for zoned BCD's).                                          |
See Also

StructFieldAttribute Class
Vajhoej.Record Namespace
| See Also
A Sandcastle Documented Class Library
StructFieldAttribute....::Decimals Property
Field decimals (for BCD's).

Namespace: **Vajhoej.Record**
Assembly: Record (in Record.dll) Version: 0.0.0.0
Syntax

C#  Visual Basic  Visual C++

public int Decimals { get; set; }

Public Property Decimals As Integer
    Get
    Set

public:
    property int Decimals {
        int get ();
        void set (int value);
    }

See Also

StructFieldAttribute Class
Vajhoej.Record Namespace
Field encoding (for strings).

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
Syntax

C#   Visual Basic   Visual C++

public string Encoding { get; set; }

Public Property Encoding As String
    Get
    Set

public:
property String^ Encoding {
    String^ get ();
    void set (String^ value);
}
See Also

StructFieldAttribute Class
Vajhoej.Record Namespace
| See Also
A Sandcastle Documented Class Library
StructFieldAttribute::Length Property
Field length (for fixed length strings and BCD's).

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
Syntax

C#   Visual Basic   Visual C++

public int Length { get; set; }

Public Property Length As Integer
    Get
    Set

public:
    property int Length {
        int get ();
        void set (int value);
    }
See Also

StructFieldAttribute Class
Vajhoej.Record Namespace
A Sandcastle Documented Class Library

**StructFieldAttribute::N Property**

Field number.

**Namespace:** [Vajhoej.Record](#)

**Assembly:** Record (in Record.dll) Version: 0.0.0.0
- Syntax

C#   Visual Basic   Visual C++

public int N { get; set; }

Public Property N As Integer
Get
Set

public:
property int N {
    int get();
    void set(int value);
}


See Also

StructFieldAttribute Class
Vajhoej.Record Namespace
| See Also

A Sandcastle Documented Class Library

StructFieldAttribute::PrefixLength Property

Prefix length (for variable length strings).

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
Syntax

C#       Visual Basic       Visual C++

public int PrefixLength { get; set; }

Public Property PrefixLength As Integer
    Get
    Set

public:
property int PrefixLength {
    int get ();
    void set (int value);
}
See Also

StructFieldAttribute Class
Vajhoej.Record Namespace
See Also
A Sandcastle Documented Class Library
StructFieldAttribute::Type Property
Field type.

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
## Syntax

C#  Visual Basic  Visual C++

public FieldType Type { get; set; }

Public Property Type As FieldType
    Get
    Set

public:
    property FieldType Type {
        FieldType get ();
        void set (FieldType value);
    }

}
See Also

StructFieldAttribute Class
Vajhoej.Record Namespace
See Also
A Sandcastle Documented Class Library
StructFieldAttribute...::Zone Property
Field zone value (for zoned BCD's).

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
Syntax

C#    Visual Basic    Visual C++

public byte Zone { get; set; }

Public Property Zone As Byte
    Get
    Set

public:
    property unsigned_char Zone {
        unsigned_char get ();
        void set (unsigned_char value);
    }

See Also

StructFieldAttribute Class
Vajhoej.Record Namespace
A Sandcastle Documented Class Library

StructInfo Class

Class StructInfo contains information about a native struct needed for reading and/or writing.

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
Syntax

C#   Visual Basic   Visual C++

public class StructInfo
Public Class StructInfo
public ref class StructInfo
## Members

All Members  Constructors  Properties  Methods  
Public  Instance  Declared  XNA Framework Only  
Protected  Static  Inherited  .NET Compact Framework Only  

<table>
<thead>
<tr>
<th>Member</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>StructInfo(Endian, Alignment, Boolean, IList&lt;Of &lt;='(FieldInfo))&gt;&gt;, Type)</td>
<td>Create instance of StructInfo.</td>
</tr>
<tr>
<td>Alignment</td>
<td>Alignment.</td>
</tr>
<tr>
<td>Analyze(Type)</td>
<td>Analyze class.</td>
</tr>
<tr>
<td>Endianess</td>
<td>Endianess.</td>
</tr>
<tr>
<td>Endpad</td>
<td>Pad at end. Determines whether the specified Object is equal to the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Equals(Object)</td>
<td>Fields.</td>
</tr>
<tr>
<td>Fields</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize()</td>
<td>Fixed length struct.</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>Length</td>
<td>Length.</td>
</tr>
<tr>
<td>MemberwiseClone()</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.) Returns a string that represents the current</td>
</tr>
</tbody>
</table>
ToString(object)
(Inherited from Object.)
Inheritance Hierarchy

System..::..Object
Vajhoej.Record..::..StructInfo
See Also

Vajhoej.Record Namespace
See Also
A Sandcastle Documented Class Library
StructInfo Constructor
Create instance of StructInfo.

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
Syntax

C# Visual Basic Visual C++

public StructInfo(
    Endian endianess,
    Alignment alignment,
    bool endpad,
    IList<FieldInfo> fields,
    Type clz
)

Public Sub New ( _
    endianess As Endian, _
    alignment As Alignment, _
    endpad As Boolean, _
    fields As IList(Of FieldInfo), _
    clz As Type _
)

public:
StructInfo(
    Endian endianess,
    Alignment alignment,
    bool endpad,
    IList<FieldInfo>^ fields,
    Type^ clz
)

Parameters

endianess
    Endian
    Byte order for all fields.

alignment
    Alignment
    Alignment for all fields.
endpad

**Boolean**
Pad at end.

fields

**IList<(Of (Of ('FieldInfo')))>**
Array of FieldInfo describing all fields.

clz

**Type**
Class implementing struct.
## Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vajhoej.Record:::RecordException</td>
<td>If error calculation length information.</td>
</tr>
</tbody>
</table>
See Also

StructInfo Class
Vajhoej.Record Namespace
See Also
A Sandcastle Documented Class Library
StructInfo Methods

The `StructInfo` type exposes the following methods.
## Methods

<table>
<thead>
<tr>
<th>Public</th>
<th>Instance</th>
<th>Declared</th>
<th>XNA Framework Only</th>
</tr>
</thead>
<tbody>
<tr>
<td>Protected</td>
<td>Static</td>
<td>Inherited</td>
<td>.NET Compact Framework Only</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Member</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>❧ <strong>Analyze(Type)</strong></td>
<td>Analyze class.</td>
</tr>
<tr>
<td>❧ <strong>Equals(Object)</strong></td>
<td>Determines whether the specified Object is equal to the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>❧ <strong>Finalize()</strong></td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (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>MemberwiseClone()</strong></td>
<td>Creates a shallow copy of the current Object. (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

StructInfo Class
Vajhoej.Record Namespace
A Sandcastle Documented Class Library
StructInfo..:::Analyze Method

Analyze class.

**Namespace:** [Vajhoej.Record](#)
**Assembly:** Record (in Record.dll) Version: 0.0.0.0
**Syntax**

C#    Visual Basic    Visual C++

```csharp
public static StructInfo Analyze(
    Type clz
)
```

```vbnet
Public Shared Function Analyze ( _
    clz As Type _
) As StructInfo
```

```cpp
public:
static StructInfo^ Analyze(
    Type^ clz
)
```

**Parameters**

clz  

_Type_

Class to analyze.

**Return Value**

StructInfo for class.
### Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vajhoej.Record:::RecordException</td>
<td>If error calculation length information.</td>
</tr>
</tbody>
</table>
See Also

StructInfo Class
Vajhoej.Record Namespace
The `StructInfo` type exposes the following properties.
# Properties

- **Public**  - **Instance**  - **Declared**  - **XNA Framework Only**

- **Protected**  - **Static**  - **Inherited**  - **.NET Compact Framework Only**

<table>
<thead>
<tr>
<th>Member</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alignment</td>
<td>Alignment.</td>
</tr>
<tr>
<td>Endianess</td>
<td>Endianess.</td>
</tr>
<tr>
<td>Endpad</td>
<td>Pad at end.</td>
</tr>
<tr>
<td>Fields</td>
<td>Fields.</td>
</tr>
<tr>
<td>FixedLength</td>
<td>Fixed length struct.</td>
</tr>
<tr>
<td>Length</td>
<td>Length.</td>
</tr>
</tbody>
</table>
See Also

StructInfo Class
Vajhoej.Record Namespace
See Also
A Sandcastle Documented Class Library
StructInfo..::Alignment Property
Alignment.

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
Syntax

C#  Visual Basic  Visual C++

public Alignment Alignment { get; }

Public ReadOnly Property Alignment As Alignment
    Get

public:
property Alignment Alignment {
    Alignment get ();
}
}
See Also

StructInfo Class
Vajhoej.Record Namespace
See Also
A Sandcastle Documented Class Library
StructInfo..:::Endianess Property
Endianess.

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
C#    Visual Basic    Visual C++

public  **Endian** Endianess { get; }

Public ReadOnly Property Endianess As **Endian**
    Get

public:
property **Endian** Endianess {
    **Endian** get ();
}


See Also

- StructInfo Class
- Vajhoej.Record Namespace
See Also
A Sandcastle Documented Class Library
StructInfo..::Endpad Property
Pad at end.

Namespace:  Vajhoej.Record
Assembly:  Record (in Record.dll) Version: 0.0.0.0
Syntax

C#      Visual Basic      Visual C++

public bool Endpad { get; }

Public ReadOnly Property Endpad As Boolean
Get

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

See Also

StructInfo Class
Vajhoej.Record Namespace
See Also
A Sandcastle Documented Class Library
StructInfo..::Fields Property

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
Syntax

C#  Visual Basic  Visual C++

public IList<FieldInfo> Fields { get; }

Public ReadOnly Property Fields As IList(Of FieldInfo)
     Get

public:
property IList<FieldInfo>^ Fields {
     IList<FieldInfo>^ get ();
}
See Also

StructInfo Class
Vajhoej.Record Namespace
See Also
A Sandcastle Documented Class Library
StructInfo..::FixedLength Property

Fixed length struct.

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
Syntax

C#   Visual Basic   Visual C++

public   bool   FixedLength   {   get;   }

Public    ReadOnly    Property    FixedLength    As    Boolean

Get

c::public:    
property    bool    FixedLength    {
   bool    get ();
}

}
See Also

StructInfo Class
Vajhoej.Record Namespace
See Also
A Sandcastle Documented Class Library
StructInfo..::Length Property
Length.

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
Syntax

C#   Visual Basic   Visual C++

public int Length { get; }

Public ReadOnly Property Length As Integer
    Get

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

See Also

StructInfo Class
Vajhoej.Record Namespace
Class StructInfoCache caches StructInfo objects in a singleton cache.

**Namespace:** Vajhoej.Record  
**Assembly:** Record (in Record.dll) Version: 0.0.0.0
Syntax

C#  Visual Basic  Visual C++

public class StructInfoCache
Public Class StructInfoCache
public ref class StructInfoCache
## Members

All Members  Properties ➤  Methods ➤
- Public  - Instance  - Declared  - XNA Framework Only ✗
- Protected  - Static  - Inherited  - .NET Compact Framework Only ✓

<table>
<thead>
<tr>
<th>Member</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>➤ Analyze(Type)</td>
<td>Convenience method to get StructInfo from cache and analyze class if not in cache.</td>
</tr>
<tr>
<td>➤ Equals(Object)</td>
<td>Determines whether the specified Object is equal to the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>➤ Finalize()</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>➤ Get(Type)</td>
<td>Get StructInfo from cache. (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>➤ HitRate</td>
<td>Cache hit rate.</td>
</tr>
<tr>
<td>➤ Instance</td>
<td>Singleton instance.</td>
</tr>
<tr>
<td>➤ MemberwiseClone()</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>➤ Put(Type, StructInfo)</td>
<td>Put StructInfo into cache.</td>
</tr>
<tr>
<td>➤ Reset()</td>
<td>Reset cache.</td>
</tr>
<tr>
<td>➤ ToString()</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>
Inheritance Hierarchy

System. Object
Vajhoj. Record. StructInfoCache
See Also

Vajhoej.Record Namespace
<table>
<thead>
<tr>
<th>See Also</th>
</tr>
</thead>
<tbody>
<tr>
<td>A Sandcastle Documented Class Library</td>
</tr>
<tr>
<td>StructInfoCache Methods</td>
</tr>
</tbody>
</table>

The `StructInfoCache` type exposes the following methods.
<table>
<thead>
<tr>
<th>Member</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Analyze(Type)</code></td>
<td>Convenience method to get StructInfo from cache and analyze class if not in cache.</td>
</tr>
<tr>
<td><code>Equals(Object)</code></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><code>Finalize()</code></td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td><code>Get(Type)</code></td>
<td>Get StructInfo from cache.</td>
</tr>
<tr>
<td><code>GetHashCode()</code></td>
<td>Serves as a hash function for a particular type.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td><code>GetType()</code></td>
<td>Gets the type of the current instance.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td><code>MemberwiseClone()</code></td>
<td>Creates a shallow copy of the current Object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td><code>Put(Type, StructInfo)</code></td>
<td>Put StructInfo into cache.</td>
</tr>
<tr>
<td><code>Reset()</code></td>
<td>Reset cache.</td>
</tr>
<tr>
<td><code>ToString()</code></td>
<td>Returns a string that represents the current object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
</tbody>
</table>
See Also

StructInfoCache Class
Vajhoej.Record Namespace
See Also
A Sandcastle Documented Class Library
StructInfoCache::Analyze Method

Convenience method to get StructInfo from cache and analyze class if not in cache.

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
**Syntax**

C#  Visual Basic  Visual C++

public static StructInfo Analyze(
    Type t
)

Public Shared Function Analyze ( _
    t As Type _
) As StructInfo

public:
static StructInfo^ Analyze(
    Type^ t
)

**Parameters**

t
    Type
Class.

**Return Value**

StructInfo for class.
<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vajhoej.Record.....RecordException</td>
<td>If error analyzing cache.</td>
</tr>
</tbody>
</table>
See Also

StructInfoCache Class
Vajhoej.Record Namespace
See Also
A Sandcastle Documented Class Library
StructInfoCache...::Get Method
Get StructInfo from cache.

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
Syntax

C#    Visual Basic    Visual C++

public StructInfo Get(
    Type clz
)

Public Function Get (_
    clz As Type _
) As StructInfo

public:
    StructInfo^ Get(
        Type^ clz
    )

Parameters

clz
    Type
        Class we want StructInfo for.

Return Value

StructInfo.
See Also

StructInfoCache Class
Vajhoej.Record Namespace
See Also
A Sandcastle Documented Class Library
StructInfoCache:::Put Method

Put StructInfo into cache.

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
**Syntax**

**C#**

```csharp
public void Put(
    Type clz,
    StructInfo si
)
```

**Visual Basic**

```vb
Public Sub Put (_
    clz As Type, _
    si As StructInfo _
)
```

**Visual C++**

```cpp
public:
    void Put(
        Type^ clz,
        StructInfo^ si
    )
```

### Parameters

**clz**

`Type`

Class we have StructInfo for.

**si**

`StructInfo`

StructInfo.
See Also

- StructInfoCache Class
- Vajhoej.Record Namespace
C#  
Visual Basic  
Visual C++

See Also
A Sandcastle Documented Class Library
StructInfoCache...::Reset Method
Reset cache.

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
Syntax

C# Visual Basic Visual C++

public void Reset()
Public Sub Reset

public:
void Reset()
See Also

StructInfoCache Class
Vajhoej.Record Namespace
The **StructInfoCache** type exposes the following properties.
Properties

- Public
- Instance
- Declared
- XNA Framework Only

- Protected
- Static
- Inherited
- .NET Compact Framework Only

<table>
<thead>
<tr>
<th>Member</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HitRate</td>
<td>Cache hit rate.</td>
</tr>
<tr>
<td>Instance</td>
<td>Singleton instance.</td>
</tr>
</tbody>
</table>
See Also

StructInfoCache Class
Vajhoej.Record Namespace
C# Visual Basic Visual C++

See Also
A Sandcastle Documented Class Library
StructInfoCache::HitRate Property
Cache hit rate.

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
Syntax

C#  Visual Basic  Visual C++

public double HitRate { get; }

Public ReadOnly Property HitRate As Double
Get

public:
property double HitRate {
    double get ();
}


See Also

StructInfoCache Class
Vajhoej.Record Namespace
See Also
A Sandcastle Documented Class Library
StructInfoCache...::Instance Property
Singleton instance.

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
**Syntax**

C#    Visual Basic    Visual C++

```csharp
public static StructInfoCache Instance { get; }
```

```vbnet
Public Shared ReadOnly Property Instance As StructInfoCache
Get
```

```cpp
public:
static property StructInfoCache^ Instance {
StructInfoCache^ get ();
}
```
See Also

StructInfoCache Class
Vajhoej.Record Namespace
Class StructReader reads a .NET object from a byte array containing a native struct.

**Namespace:** Vajhoej.Record
**Assembly:** Record (in Record.dll) Version: 0.0.0.0
Syntax

C#     Visual Basic     Visual C++

```csharp
public class StructReader
Public Class StructReader
public ref class StructReader
```
## Members

**All Members**  |  **Constructors**  |  **Properties**  |  **Methods**
---|---|---|---
**Public**  |  **Instance**  |  **Declared**  |  **XNA Framework Only**

### Protected  Static  Inherited  .NET Compact Framework Only

<table>
<thead>
<tr>
<th>Member</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>.StructReader(array&lt;Byte&gt;[][][])</td>
<td>Create instance of StructReader. Determines whether the specified Object is equal to the current Object. (Inherited from Object.) Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.) Serves as a hash function for a particular type. (Inherited from Object.) Gets the type of the current instance. (Inherited from Object.) Creates a shallow copy of the current Object. (Inherited from Object.) More records available. Read.</td>
</tr>
</tbody>
</table>

### More

Read(Of <<(T)>>)(Type)

Read(Of <<(T)>>)(Type, LengthProvider)

Read(Of <<(T)>>)(Type, LengthProvider, MaxLengthProvider)

Read(Of <<(T)>>)(Type,
LengthProvider, Read.
MaxLengthProvider, Read
ElementsProvider)
Read<(Of <<(T)>>)(Type, LengthProvider.
MaxLengthProvider, Read
ElementsProvider, ConvertSelector)
ToString()()

Returns a string that represents the current object.
(Inherited from Object.)
Inheritance Hierarchy

System...Object
Vajhoej.Record......StructReader
See Also

Vajhoej.Record Namespace
Create instance of StructReader.

**Namespace:** Vajhoej.Record  
**Assembly:** Record (in Record.dll) Version: 0.0.0.0
**Syntax**

C#   Visual Basic   Visual C++

public StructReader(
    byte[] ba
)

Public Sub New ( _
    ba As Byte() _
)

public:
StructReader(
    array<unsigned char>^ ba
)

**Parameters**

ba

array<Byte>[]()[[]]

Byte array to read from.
See Also

StructReader Class
Vajhoej.Record Namespace
See Also
A Sandcastle Documented Class Library
StructReader Methods

The **StructReader** type exposes the following methods.
## Methods

- **Public**
- **Instance**
- **Declared**
- **XNA Framework Only**

- **Protected**
- **Static**
- **Inherited**
- **.NET Compact Framework Only**

<table>
<thead>
<tr>
<th>Member</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals(Object)</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>Finalize()</strong></td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (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 type of the current instance. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>MemberwiseClone()</strong></td>
<td>Creates a shallow copy of the current <strong>Object</strong>. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>Read(Of &lt;&lt;'(T)&gt;&gt;&gt;())(Type)</strong></td>
<td>Read.</td>
</tr>
<tr>
<td><strong>Read(Of &lt;&lt;'(T)&gt;&gt;&gt;())(Type, LengthProvider)</strong></td>
<td>Read.</td>
</tr>
<tr>
<td><strong>Read(Of &lt;&lt;'(T)&gt;&gt;&gt;())(Type, LengthProvider, MaxLengthProvider)</strong></td>
<td>Read.</td>
</tr>
<tr>
<td><strong>Read(Of &lt;&lt;'(T)&gt;&gt;&gt;())(Type, LengthProvider, MaxLengthProvider, ElementsProvider)</strong></td>
<td>Read.</td>
</tr>
</tbody>
</table>
MaxLengthProvider, ElementsProvider, ConvertSelector)

ToString

Read.

Returns a string that represents the current object.
(Inherited from Object.)
See Also

StructReader Class
Vajhoej.Record Namespace
See Also
A Sandcastle Documented Class Library
StructReader:::Read Method
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Read&lt;(Of &lt;&lt;(T)&gt;&gt;)(Type)</code></td>
<td>Read.</td>
</tr>
<tr>
<td><code>Read&lt;(Of &lt;&lt;'(T)&gt;&gt;)(Type, LengthProvider)</code></td>
<td>Read.</td>
</tr>
<tr>
<td><code>Read&lt;(Of &lt;&lt;'(T)&gt;&gt;)(Type, LengthProvider, MaxLengthProvider)</code></td>
<td>Read.</td>
</tr>
<tr>
<td><code>Read&lt;(Of &lt;&lt;'(T)&gt;&gt;)(Type, LengthProvider, MaxLengthProvider, ElementsProvider)</code></td>
<td>Read.</td>
</tr>
<tr>
<td><code>Read&lt;(Of &lt;&lt;'(T)&gt;&gt;)(Type, LengthProvider, MaxLengthProvider, ElementsProvider, ConvertSelector)</code></td>
<td>Read.</td>
</tr>
</tbody>
</table>
See Also

StructReader Class
Vajhoej.Record Namespace
See Also
A Sandcastle Documented Class Library
StructReader::Read(Of <(<'T'>)>) Method (Type)
Read.

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
Syntax

C#    Visual Basic    Visual C++

public T Read<T>(
    Type t
)
where T : class, new()

Public Function Read(Of T As {Class, New}) ( _
    t As Type _
) As T

public:
generic<typename T>
where T : ref class, gcnew()
T Read(
    Type^ t
)

Parameters

t
    Type
    Type of what to read.
**Type Parameters**

T  
Type of what to read.

**Return Value**

Object read.
## Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vajhoej.Record...:::RecordException</td>
<td>If impossible to convert between types in class and struct.</td>
</tr>
</tbody>
</table>
See Also

StructReader Class
Read Overload
Vajhoej.Record Namespace
See Also
A Sandcastle Documented Class Library
StructReader...::Read<(Of <*>>) Method (Type, LengthProvider)
Read.

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
Syntax

C#   Visual Basic   Visual C++

```csharp
public T Read<T>(
    Type t,
    LengthProvider lenpvd
) where T : class, new()
```

```vbnet
Public Function Read(Of T As {Class, New}) ( _
    t As Type, _
    lenpvd As LengthProvider _
) As T
```

```cpp
public:

generic< typename T>
where T : ref class, gcnew()
T Read(
    Type^ t,
    LengthProvider^ lenpvd
)
```

**Parameters**

- `t`
  - `Type`
    - Type of what to read.

- `lenpvd`
  - `LengthProvider`
    - Supplies length for fields where it is not given.
Type Parameters

T
  Type of what to read.

Return Value

Object read.
## Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vajhoej.Record...:::RecordException</td>
<td>If impossible to convert between types in class and struct.</td>
</tr>
</tbody>
</table>
See Also

StructReader Class  
Read Overload  
Vajhoej.Record Namespace
| See Also |
A Sandcastle Documented Class Library
StructReader...::Read<Of <(<'T')>)> Method (Type, LengthProvider, MaxLengthProvider)
Read.

**Namespace:** Vajhoej.Record
**Assembly:** Record (in Record.dll) Version: 0.0.0.0
Syntax

C#   Visual Basic   Visual C++

```csharp
public T Read<T>(
    Type t,
    LengthProvider lenpvd,
    MaxLengthProvider maxlenpvd
)

where T : class, new()

Public Function Read(Of T As {Class, New}) ( _
    t As Type, _
    lenpvd As LengthProvider, _
    maxlenpvd As MaxLengthProvider _
) As T

public:
    generic<typename T>
    where T : ref class, gcnew()
    T Read(
        Type^ t,
        LengthProvider^ lenpvd,
        MaxLengthProvider^ maxlenpvd
    )
```

**Parameters**

t

*Type*
Type of what to read.

lenpvd

*LengthProvider*
Supplies length for fields where it is not given.

maxlenpvd

*MaxLengthProvider*
Supplies max length for fields where it is not given.
- **Type Parameters**

  **T**  
  Type of what to read.

- **Return Value**

  Object read.
<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vajhoej.Record:::RecordException</td>
<td>If impossible to convert between types in class and struct.</td>
</tr>
</tbody>
</table>
See Also

StructReader Class
Read Overload
Vajhoej.Record Namespace
See Also
A Sandcastle Documented Class Library
StructReader...:::Read<(Of <(<'T'>)>)> Method (Type, LengthProvider, MaxLengthProvider, ElementsProvider)
Read.

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
Syntax

C#   Visual Basic   Visual C++

public T Read<T>(
    Type t,
    LengthProvider lenpvd,
    MaxLengthProvider maxlenpvd,
    ElementsProvider elmpvd
)
where T : class, new()

Public Function Read(Of T As {Class, New}) ( _
    t As Type, _
    lenpvd As LengthProvider, _
    maxlenpvd As MaxLengthProvider, _
    elmpvd As ElementsProvider _
) As T

public:
    generic<typename T>
    where T : ref class, gcnew()
    T Read(
        Type^ t,
        LengthProvider^ lenpvd,
        MaxLengthProvider^ maxlenpvd,
        ElementsProvider^ elmpvd
    )

Parameters

    t
        Type
        Type of what to read.

    lenpvd
        LengthProvider
        Supplies length for fields where it is not given.
maxlenpvd
    MaxLengthProvider
    Supplies max length for fields where it is not given.

elmpvd
    ElementsProvider
    Supplies elements for fields where it is not given.
Type Parameters

T
   Type of what to read.

Return Value

Object read.
<table>
<thead>
<tr>
<th><strong>Exception</strong></th>
<th><strong>Condition</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Vajhoej.Record::RecordException</td>
<td>If impossible to convert between types in class and struct.</td>
</tr>
</tbody>
</table>
See Also

StructReader Class
Read Overload
Vajhoej.Record Namespace
See Also

A Sandcastle Documented Class Library

StructReader...::Read(Of <(<<>?)>) Method (Type, LengthProvider, MaxLengthProvider, ElementsProvider, ConvertSelector)

Read.

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
**Syntax**

C#    Visual Basic    Visual C++

```csharp
public T Read<T>(
    Type t,
    LengthProvider lenpvd,
    MaxLengthProvider maxlenpvd,
    ElementsProvider elmpvd,
    ConvertSelector infpvd
)
where T : class, new()
```

```vbnet
Public Function Read(Of T As {Class, New}) ( _
    t As Type, _
    lenpvd As LengthProvider, _
    maxlenpvd As MaxLengthProvider, _
    elmpvd As ElementsProvider, _
    infpvd As ConvertSelector _
) As T
```

```csharp
public:
    generic<typename T>
    where T : ref class, gcnew()
    T Read(  
        Type^ t,  
        LengthProvider^ lenpvd,  
        MaxLengthProvider^ maxlenpvd,  
        ElementsProvider^ elmpvd,  
        ConvertSelector^ infpvd  
    )
```

### Parameters

- **t**
  - `Type`
  - Type of what to read.

- **lenpvd**
  - `LengthProvider`
Supplies length for fields where it is not given.

maxlenpvd
   MaxLengthProvider
   Supplies max length for fields where it is not given.

elmpvd
   ElementsProvider
   Supplies elements for fields where it is not given.

infpvd
   ConvertSelector
   Supplies selector converter for fields where it is needed.
**Type Parameters**

T

Type of what to read.

**Return Value**

Object read.
# Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vajhoej.Record...:::RecordException</td>
<td>If impossible to convert between types in class and struct.</td>
</tr>
</tbody>
</table>
See Also

StructReader Class
Read Overload
Vajhoej.Record Namespace
The `StructReader` type exposes the following properties.
## Properties

- **Public**
- **Instance**
- **Declared**
- **XNA Framework Only**

- **Protected**
- **Static**
- **Inherited**
- **.NET Compact Framework Only**

<table>
<thead>
<tr>
<th>Member</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>More</td>
<td>More records available.</td>
</tr>
</tbody>
</table>
See Also

StructReader Class
Vajhoej.Record Namespace
See Also

A Sandcastle Documented Class Library
StructReader...::More Property

More records available.

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
Syntax

C#   Visual Basic   Visual C++

public bool More { get; }

Public ReadOnly Property More As Boolean
    Get

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

See Also

StructReader Class
Vajhoej.Record Namespace
Class StructWriter writes a .NET object to a byte array as a native struct.

**Namespace:** Vajhoej.Record  
**Assembly:** Record (in Record.dll) Version: 0.0.0.0
Syntax

C#    Visual Basic    Visual C++

public class StructWriter
Public Class StructWriter
public ref class StructWriter
## Members

<table>
<thead>
<tr>
<th>Member</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>StructWriter()()</code></td>
<td>Construct instance of <code>StructWriter</code> with default buffer size.</td>
</tr>
<tr>
<td><code>Equals(Object)</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>Extend(Int32)</code></td>
<td>Extend capacity.</td>
</tr>
<tr>
<td><code>Finalize()()</code></td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>GetBytes()</code></td>
<td>Get bytes.</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 type of the current instance.                                     (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>Length</code></td>
<td>The length.</td>
</tr>
<tr>
<td><code>MemberwiseClone()</code></td>
<td>Creates a shallow copy of the current <code>Object</code>.                            (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>ToString()</code></td>
<td>Returns a string that represents the current object.                      (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>Write(Object)</code></td>
<td>Write.</td>
</tr>
</tbody>
</table>
Write(Object, LengthProvider)  Write.
Write(Object, LengthProvider, MaxLengthProvider)  Write.
Write(Object, LengthProvider, MaxLengthProvider, ElementsProvider)  Write.
Write(Object, LengthProvider, MaxLengthProvider, ElementsProvider, ConvertSelector)  Write.
Inheritance Hierarchy

System::Object
Vajhoej.Record::StructWriter
See Also

Vajhoej.Record Namespace
See Also
A Sandcastle Documented Class Library
StructWriter Constructor
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>StructWriter()</code></td>
<td>Construct instance of StructWriter with default buffer size.</td>
</tr>
</tbody>
</table>
See Also

StructWriter Class
Vajhoej.Record Namespace
See Also

A Sandcastle Documented Class Library

StructWriter Constructor

Construct instance of StructWriter with default buffer size.

**Namespace:** Vajhoej.Record

**Assembly:** Record (in Record.dll) Version: 0.0.0.0
Syntax

C#   Visual Basic   Visual C++

public StructWriter()
Public Sub New

public:
StructWriter()
See Also

StructWriter Class
StructWriter Overload
Vajhoej.Record Namespace
A Sandcastle Documented Class Library
StructWriter Constructor (Int32)
Construct instance of StructWriter.

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
Syntax

C#   Visual Basic   Visual C++

public StructWriter(
                  int bufsiz
)       

Public Sub New (_
                  bufsiz As Integer _
)

public:
StructWriter(
                  int bufsiz
)

Parameters

bufsiz
                  Int32
Size of byte array to write to.
See Also

StructWriter Class
StructWriter Overload
Vajhoej.Record Namespace
The `StructWriter` type exposes the following methods.
# Methods

<table>
<thead>
<tr>
<th>Public</th>
<th>Instance</th>
<th>Declared</th>
<th>XNA Framework Only</th>
<th>Protected</th>
<th>Static</th>
<th>Inherited</th>
<th>.NET Compact Framework Only</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Member</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Equals&lt;Object&gt;]</td>
<td>Determines whether the specified <a href="#">Object</a> is equal to the current <a href="#">Object</a>. <em>(Inherited from <a href="#">Object</a>).</em></td>
</tr>
<tr>
<td>![Extend(Int32)]</td>
<td>Extend capacity.</td>
</tr>
<tr>
<td><img src="#" alt="Finalize()" /></td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. <em>(Inherited from <a href="#">Object</a>).</em></td>
</tr>
<tr>
<td><img src="#" alt="getBytes()" /></td>
<td>Get bytes.</td>
</tr>
<tr>
<td><img src="#" alt="GetHashCode()" /></td>
<td>Serves as a hash function for a particular type. <em>(Inherited from <a href="#">Object</a>).</em></td>
</tr>
<tr>
<td><img src="#" alt="GetType()" /></td>
<td>Gets the type of the current instance. <em>(Inherited from <a href="#">Object</a>).</em></td>
</tr>
<tr>
<td><img src="#" alt="MemberwiseClone()" /></td>
<td>Creates a shallow copy of the current <a href="#">Object</a>. <em>(Inherited from <a href="#">Object</a>).</em></td>
</tr>
<tr>
<td><img src="#" alt="ToString()" /></td>
<td>Returns a string that represents the current object. <em>(Inherited from <a href="#">Object</a>).</em></td>
</tr>
<tr>
<td>![Write(Object)]</td>
<td>Write.</td>
</tr>
<tr>
<td>![Write(Object, LengthProvider)]</td>
<td>Write.</td>
</tr>
<tr>
<td>![Write(Object, LengthProvider, MaxLengthProvider)]</td>
<td>Write.</td>
</tr>
<tr>
<td>![Write(Object, MaxLengthProvider, ElementsProvider)]</td>
<td>Write.</td>
</tr>
<tr>
<td>![Write(Object, LengthProvider)]</td>
<td>Write.</td>
</tr>
</tbody>
</table>
MaxLengthProvider, ElementsProvider, ConvertSelector)

Write.
See Also

StructWriter Class
Vajhoej.Record Namespace
A Sandcastle Documented Class Library
StructWriter...::Extend Method

Extend capacity.

**Namespace:** [Vajhoej.Record](#)

**Assembly:** Record (in Record.dll) Version: 0.0.0.0
Syntax

C#   Visual Basic   Visual C++

public void Extend(
    int newbufsiz
)

Public Sub Extend (_
    newbufsiz As Integer _
)

public:
void Extend(
    int newbufsiz
)

Parameters

newbufsiz
    Int32
    New size of byte array to write to.
See Also

StructWriter Class
Vajhoej.Record Namespace
See Also
A Sandcastle Documented Class Library
StructWriter...::GetBytes Method
Get bytes.

Namespace:  Vajhoej.Record
Assembly:  Record (in Record.dll) Version: 0.0.0.0
Syntax

C#   Visual Basic   Visual C++

public byte[] GetBytes()

Public Function GetBytes As Byte()

public:
array<unsigned char>^ GetBytes()

Return Value

The resulting byte array.
See Also
A Sandcastle Documented Class Library
StructWriter::Write Method
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Write(Object)</td>
<td>Write.</td>
</tr>
<tr>
<td>Write(Object, LengthProvider)</td>
<td>Write.</td>
</tr>
<tr>
<td>Write(Object, LengthProvider, MaxLengthProvider)</td>
<td>Write.</td>
</tr>
<tr>
<td>Write(Object, LengthProvider, MaxLengthProvider, ElementsProvider)</td>
<td>Write.</td>
</tr>
<tr>
<td>Write(Object, LengthProvider, MaxLengthProvider, ElementsProvider, ConvertSelector)</td>
<td>Write.</td>
</tr>
</tbody>
</table>
See Also

StructWriter Class
Vajhoej.Record Namespace
See Also
A Sandcastle Documented Class Library
StructWriter...::Write Method (Object)
Write.

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
# Syntax

<table>
<thead>
<tr>
<th></th>
<th>C#</th>
<th>Visual Basic</th>
<th>Visual C++</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>public void Write(</td>
<td>Public Sub Write(</td>
<td>public: void Write(</td>
</tr>
<tr>
<td></td>
<td>Object o</td>
<td>o As Object _</td>
<td>Object^ o</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

## Parameters

- **Object**
  - Object to write.
<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vajhoej.Record::RecordException</td>
<td>If impossible to convert between types in class and struct.</td>
</tr>
</tbody>
</table>
See Also

StructWriter Class
Write Overload
Vajhoej.Record Namespace
See Also

A Sandcastle Documented Class Library
StructWriter...::Write Method (Object, LengthProvider)
Write.

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
Syntax

C#    Visual Basic    Visual C++

public void Write(  
    Object o,  
    LengthProvider lenpvd  
  )

Public Sub Write ( _  
    o As Object, _  
    lenpvd As LengthProvider _  
  )

public: 
void Write(  
    Object^ o,  
    LengthProvider^ lenpvd  
  )

Parameters

    o

        Object
        Object to write.

    lenpvd

        LengthProvider
        Supplies length for fields where it is not given -
# Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vajhoej.Record::RecordException</td>
<td>If impossible to convert between types in class and struct.</td>
</tr>
</tbody>
</table>
See Also

StructWriter Class
Write Overload
Vajhoej.Record Namespace
See Also
A Sandcastle Documented Class Library
StructWriter...::Write Method (Object, LengthProvider, MaxLengthProvider)
Write.

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
<Syntax>

C#   Visual Basic   Visual C++

public void Write(
    Object o,
    LengthProvider lenpvd,
    MaxLengthProvider maxlenpvd
)

Public Sub Write ( _
    o As Object, _
    lenpvd As LengthProvider, _
    maxlenpvd As MaxLengthProvider _
)

public:
void Write(
    Object^ o,
    LengthProvider^ lenpvd,
    MaxLengthProvider^ maxlenpvd
)

Parameters

o
Object
Object to write.

lenpvd
LengthProvider
Supplies length for fields where it is not given -

maxlenpvd
MaxLengthProvider
Supplies max length for fields where it is not given.
<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vajhoej.Record:::RecordException</td>
<td>If impossible to convert between types in class and struct.</td>
</tr>
</tbody>
</table>
See Also

StructWriter Class
Write Overload
Vajhoj.Record Namespace
See Also

A Sandcastle Documented Class Library
StructWriter::Write Method (Object, LengthProvider, MaxLengthProvider, ElementsProvider)

Write.

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
## Syntax

**C#**

```csharp
public void Write(
    Object o,
    LengthProvider lenpvd,
    MaxLengthProvider maxlenpvd,
    ElementsProvider elmpvd
)
```

**Visual Basic**

```vbnet
Public Sub Write (_,
    o As Object, _
    lenpvd As LengthProvider, _
    maxlenpvd As MaxLengthProvider, _
    elmpvd As ElementsProvider _
)
```

**Visual C++**

```cpp
public:
void Write(
    Object^ o,
    LengthProvider^ lenpvd,
    MaxLengthProvider^ maxlenpvd,
    ElementsProvider^ elmpvd
)
```

### Parameters

- **o**
  - `Object`
  - Object to write.

- **lenpvd**
  - `LengthProvider`
  - Supplies length for fields where it is not given.

- **maxlenpvd**
  - `MaxLengthProvider`
  - Supplies max length for fields where it is not given.
ElementsProvider
Supplies elements for fields where it is not given.
## Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vajhoej.Record::&lt;RecordException&gt;</td>
<td>If impossible to convert between types in class and struct.</td>
</tr>
</tbody>
</table>
See Also

StructWriter Class
Write Overload
Vajhoej.Record Namespace
See Also
A Sandcastle Documented Class Library
StructWriter::Write Method (Object, LengthProvider, MaxLengthProvider, ElementsProvider, ConvertSelector)
Write.

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
**Syntax**

C#     Visual Basic     Visual C++

public void Write(
    Object o,
    LengthProvider lenpvd,
    MaxLengthProvider maxlenpvd,
    ElementsProvider elmpvd,
    ConvertSelector infpvd
)

Public Sub Write ( _
    o As Object, _
    lenpvd As LengthProvider, _
    maxlenpvd As MaxLengthProvider, _
    elmpvd As ElementsProvider, _
    infpvd As ConvertSelector _
)

public:
    void Write(
    Object^ o,
    LengthProvider^ lenpvd,
    MaxLengthProvider^ maxlenpvd,
    ElementsProvider^ elmpvd,
    ConvertSelector^ infpvd
)

**Parameters**

o
    Object
    Object to write.

lenpvd
    LengthProvider
    Supplies length for fields where it is not given -
maxlenpvd
   MaxLengthProvider
   Supplies max length for fields where it is not given.

ejmpvd
   ElementsProvider
   Supplies elements for fields where it is not given.

infpvd
   ConvertSelector
   Supplies selector converter for fields where it is needed.
<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vajhoej.Record::RecordException</td>
<td>If impossible to convert between types in class and struct.</td>
</tr>
</tbody>
</table>
See Also

**StructWriter Class**
**Write Overload**
**Vajhoj.Record Namespace**
The *StructWriter* type exposes the following properties.
Properties

- Public
- Instance
- Declared
- XNA Framework Only

- Protected
- Static
- Inherited
- .NET Compact Framework Only

**Member Description**

| Length | The length. |
See Also

StructWriter Class
Vajhoej.Record Namespace
See Also
A Sandcastle Documented Class Library
StructWriter:::Length Property
The length.

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
Syntax

C#    Visual Basic    Visual C++

public int Length { get; }
Public ReadOnly Property Length As Integer
    Get

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

StructWriter Class
Vajhoej.Record Namespace
Class SubClassAndPad contains information about class and padding for select field.

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
Syntax

C#    Visual Basic    Visual C++

public class SubClassAndPad
Public Class SubClassAndPad
public ref class SubClassAndPad
## Members

<table>
<thead>
<tr>
<th>Member</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>SubClassAndPad(Type, Int32)</code></td>
<td>Create instance of ClassAndPad with all necessary properties.</td>
</tr>
<tr>
<td><code>Equals(Object)</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>Finalize()</code></td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (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 type of the current instance. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>MemberwiseClone()</code></td>
<td>Creates a shallow copy of the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>Pad</code></td>
<td>Padding.</td>
</tr>
<tr>
<td><code>SubClass</code></td>
<td>Class.</td>
</tr>
<tr>
<td><code>ToString()</code></td>
<td>Returns a string that represents the current object. (Inherited from <code>Object</code>.)</td>
</tr>
</tbody>
</table>
Inheritance Hierarchy

System..::.Object
Vajhoej.Record..::.SubClassAndPad
See Also

Vajhoej.Record Namespace
SubClassAndPad Constructor
Create instance of ClassAndPad with all necessary properties.

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
## Syntax

### C#  Visual Basic  Visual C++

```csharp
public SubClassAndPad(
    Type subclass,
    int pad
)

Public Sub New (
    _
    subclass As Type, _
    pad As Integer _
)

public:
SubClassAndPad(
    Type^ subclass,
    int pad
)
```

### Parameters

- **subClass**
  - Type
  - Class.

- **pad**
  - Int32
  - Padding.
See Also

SubClassAndPad Class
Vajhoej.Record Namespace
The `SubClassAndPad` type exposes the following methods.
# Methods

- **Public**
- **Instance**
- **Declared**
- **XNA Framework Only**

- **Protected**
- **Static**
- **Inherited**
- **.NET Compact Framework Only**

<table>
<thead>
<tr>
<th>Member</th>
<th>Description</th>
</tr>
</thead>
</table>
| ![Equals](image) | Determines whether the specified `Object` is equal to the current `Object`.  
(Inherited from `Object`.) |
| ![Finalize](image) | Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection.  
(Inherited from `Object`.) |
| ![GetHashCode](image) | Serves as a hash function for a particular type.  
(Inherited from `Object`.) |
| ![GetType](image) | Gets the type of the current instance.  
(Inherited from `Object`.) |
| ![MemberwiseClone](image) | Creates a shallow copy of the current `Object`.  
(Inherited from `Object`.) |
| ![ToString](image) | Returns a string that represents the current object.  
(Inherited from `Object`.) |
See Also

SubClassAndPad Class
Vajhoej.Record Namespace
The SubClassAndPad type exposes the following properties.
Properties

Public  Instance  Declared  XNA Framework Only

Protected  Static  Inherited  .NET Compact Framework Only

**Member Description**

- **Pad**: Padding.
- **SubClass**: Class.
See Also

SubClassAndPad Class
Vajhoej.Record Namespace
See Also
A Sandcastle Documented Class Library
SubClassAndPad..::Pad Property
Padding.

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
Syntax

C#   Visual Basic   Visual C++

public int Pad { get; }

Public ReadOnly Property Pad As Integer
    Get

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

See Also

SubClassAndPad Class
Vajhoej.Record Namespace
| See Also |
A Sandcastle Documented Class Library
SubClassAndPad..::SubClass Property
Class.

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
Syntax

C#       Visual Basic       Visual C++

public  Type  SubClass { get; }

Public ReadOnly Property SubClass As Type
    Get

public:
    property Type^ SubClass {
        Type^ get();
    }

See Also

SubClassAndPad Class
Vajhoej.Record Namespace
A Sandcastle Documented Class Library
SubTypeAttribute Class
Annotation for sub types.

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
Syntax

C#  Visual Basic  Visual C++

[AttributeUsageAttribute(AttributeTargets.Field, AllowMultiple = true]
public class SubTypeAttribute : Attribute

<AttributeUsageAttribute(AttributeTargets.Field, AllowMultiple := True
Public Class SubTypeAttribute _
Inherits Attribute

[AttributeUsageAttribute(AttributeTargets::Field, AllowMultiple = true]
public ref class SubTypeAttribute : public Attribute
# Members

<table>
<thead>
<tr>
<th>All Members</th>
<th>Constructors</th>
<th>Properties</th>
<th>Methods</th>
<th>Explicit Interface Implementations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Public</td>
<td>Instance</td>
<td>Declared</td>
<td>XNA Framework Only</td>
<td></td>
</tr>
</tbody>
</table>

### Member

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initialized a new instance of the <code>SubTypeAttribute</code> class</td>
</tr>
<tr>
<td>Returns a value that indicates whether this instance is equal to a specified object. (Inherited from <code>Attribute</code>.)</td>
</tr>
<tr>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>Returns the hash code for this instance. (Inherited from <code>Attribute</code>.)</td>
</tr>
<tr>
<td>Maps a set of names to a corresponding set of dispatch identifiers. (Inherited from <code>Attribute</code>.)</td>
</tr>
<tr>
<td>Gets the type of the current instance. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>Retrieves the type information for an object, which can be used to get the type</td>
</tr>
</tbody>
</table>

### SubTypes

- `SubTypeAttribute()` |
- `Equals(Object)` |
- `Finalize()` |
- `GetHashCode()` |
- `GetIDsOfNames(Guid %, IntPtr, UInt32, UInt32, IntPtr)` |
- `GetType()` |
- `GetTypeInfo(UInt32, UInt32, IntPtr)` |
information for an interface. (Inherited from \texttt{Attribute}.)
Retrieves the number of type information interfaces that an object provides (either 0 or 1).
(Inherited from \texttt{Attribute}.)

\texttt{Invoke(UInt32, Guid\%, UInt32, Int16, IntPtr, IntPtr, IntPtr, IntPtr)}
Provides access to properties and methods exposed by an object.
(Inherited from \texttt{Attribute}.)

\texttt{IsDefaultAttribute()}
When overridden in a derived class, indicates whether the value of this instance is the default value for the derived class.
(Inherited from \texttt{Attribute}.)

\texttt{Match(Object)}
When overridden in a derived class, returns a value that indicates whether this instance equals a specified object.
(Inherited from \texttt{Attribute}.)

\texttt{MemberwiseClone()}\texttt{()}
Creates a shallow copy of the current \texttt{Object}.
(Inherited from \texttt{Object}.)

\texttt{ToString()}\texttt{()}
Returns a string that represents the current object.
(Inherited from \texttt{Object}.)

\texttt{Type}
Sub type class.
When implemented in a derived class, gets a unique identifier for this \texttt{Attribute}.
(Inherited from \texttt{Attribute}.)

\texttt{TypeId}
Value of selector.
Inheritance Hierarchy

System..::..Object
System..::..Attribute
Vajhoej.Record..::..SubTypeAttribute
See Also

Vajhoej.Record Namespace
| See Also
A Sandcastle Documented Class Library
SubTypeAttribute Constructor
Initializes a new instance of the SubTypeAttribute class

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
Syntax

C#   Visual Basic   Visual C++

public SubTypeAttribute()
Public Sub New

public:
SubTypeAttribute()
See Also

SubTypeAttribute Class
Vajhoej.Record Namespace
The SubTypeAttribute type exposes the following methods.
# Methods

<table>
<thead>
<tr>
<th>Public</th>
<th>Instance</th>
<th>Declared</th>
<th>XNA Framework Only</th>
</tr>
</thead>
<tbody>
<tr>
<td>Protected</td>
<td>Static</td>
<td>Inherited</td>
<td>.NET Compact Framework Only</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Member</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals(Object)</strong></td>
<td>Returns a value that indicates whether this instance is equal to a specified object. (Inherited from <a href="#">Attribute</a>.)</td>
</tr>
<tr>
<td><strong>Finalize()</strong></td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>GetHashCode()</strong></td>
<td>Returns the hash code for this instance. (Inherited from <a href="#">Attribute</a>.)</td>
</tr>
<tr>
<td><strong>_Attribute..::..GetIDsOfNames(Guid%, IntPtr, UInt32, UInt32, IntPtr)</strong></td>
<td>Maps a set of names to a corresponding set of dispatch identifiers. (Inherited from <a href="#">Attribute</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>_Attribute..::..GetTypeInfo(UInt32, UInt32, IntPtr)</strong></td>
<td>Retrieves the type information for an object, which can be used to get the type information for an interface. (Inherited from <a href="#">Attribute</a>.)</td>
</tr>
<tr>
<td><strong>_Attribute..::..GetTypeInfoCount(UInt32%)</strong></td>
<td>Retrieves the number of type information interfaces that an object provides (either 0 or 1). (Inherited from <a href="#">Attribute</a>.)</td>
</tr>
</tbody>
</table>
Attribute::Invoke(UInt32, Guid%,(UInt32, Int16, IntPtr, IntPtr, IntPtr, IntPtr))

Provides access to properties and methods exposed by an object.
(Inherited from Attribute.)
When overridden in a derived class, indicates whether the value of this instance is the default value for the derived class.
(Inherited from Attribute.)

IsDefaultAttribute()()

When overridden in a derived class, indicates whether the value of this instance is the default value for the derived class.
(Inherited from Attribute.)

Match(Object)

When overridden in a derived class, returns a value that indicates whether this instance equals a specified object.
(Inherited from Attribute.)

MemberwiseClone()()

Creates a shallow copy of the current Object.
(Inherited from Object.)

ToString()()

Returns a string that represents the current object.
(Inherited from Object.)
See Also

SubTypeAttribute Class
Vajhoej.Record Namespace
The `SubTypeAttribute` type exposes the following properties.
**Properties**

- Public
- Instance
- Protected
- Static
- Declared
- Inherited
- XNA Framework Only
- .NET Compact Framework Only

<table>
<thead>
<tr>
<th>Member</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Sub type class. When implemented in a derived class, gets a unique identifier for this <code>Attribute</code>. (Inherited from <code>Attribute</code>.)</td>
</tr>
<tr>
<td>TypId</td>
<td>Value of selector.</td>
</tr>
</tbody>
</table>
See Also

SubTypeAttribute Class
Vajhoej.Record Namespace
See Also
A Sandcastle Documented Class Library
SubTypeAttribute...::Type Property
Sub type class.

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
Syntax

C#    Visual Basic    Visual C++

public Type Type { get; set; }

Public Property Type As Type
    Get
    Set

public:
    property Type^ Type {
        Type^ get ();
        void set (Type^ value);
    }
}
See Also

SubTypeAttribute Class
Vajhoej.Record Namespace
C#  
Visual Basic  
Visual C++  

| See Also  
A Sandcastle Documented Class Library  
SubTypeAttribute...::Value Property  
Value of selector.  

Namespace: [Vajhoej.Record](#)  
Assembly: Record (in Record.dll) Version: 0.0.0.0
Syntax

C#    Visual Basic    Visual C++

```c#
public int Value { get; set; }

Public Property Value As Integer
    Get
    Set

public:
    property int Value {
        int get();
        void set (int value);
    }
```

```vbnet
Public Property Value As Integer
    Get
    Set

Public Property Value As Integer
    Get
    Set

Public Property Value As Integer
    Get
    Set
```

```cpp
public:
    property int Value {
        int get();
        void set (int value);
    }
```
See Also

SubTypeAttribute Class
Vajhoej.Record Namespace
A Sandcastle Documented Class Library

TimeUtil Class

Class TimeUtil converts between integers in various time formats and DateTime objects.

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
Syntax

C#   Visual Basic   Visual C++

public class TimeUtil
Public Class TimeUtil
public ref class TimeUtil
### Members

<table>
<thead>
<tr>
<th>Member</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>TimeUtil()</strong></td>
<td>Initializes a new instance of the TimeUtil class.</td>
</tr>
<tr>
<td><strong>Equals(Object)</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>Finalize()</strong></td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>FromJavaTime(Int64)</strong></td>
<td>Convert from long with Java time (milliseconds since 1-Jan-1970) to Date object.</td>
</tr>
<tr>
<td><strong>FromUnixTime(Int32)</strong></td>
<td>Convert from int with Unix time (seconds since 1-Jan-1970) to Date object.</td>
</tr>
<tr>
<td><strong>FromVMSTime(Int64)</strong></td>
<td>Convert from long with VMS time (100 nanoseconds since 17-Nov-1858) to Date object.</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 type of the current instance. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>MemberwiseClone()</strong></td>
<td>Creates a shallow copy of the current <strong>Object</strong>. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>ToJavaTime(DateTime)</strong></td>
<td>Convert from DateTime object to long with Java time (milliseconds since 1-Jan-1970).</td>
</tr>
<tr>
<td><strong>ToString()</strong></td>
<td>Returns a string that represents the current object. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>ToUnixTime(DateTime)</strong></td>
<td>Convert from Date object to int with Unix time.</td>
</tr>
</tbody>
</table>
ToVMSTime

Convert from Date object to long with VMS time (100 nanoseconds since 17-Nov-1858).

(seconds since 1-Jan-1970).
Inheritance Hierarchy

System...Object
Vajhoej.Record.....TimeUtil
See Also

Vajhoej.Record Namespace
C#  Visual Basic  Visual C++

See Also
A Sandcastle Documented Class Library
TimeUtil Constructor
Initializes a new instance of the TimeUtil class

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
Syntax

C#     Visual Basic     Visual C++

public TimeUtil()
Public Sub New

public:
TimeUtil()
See Also

TimeUtil Class
Vajhoej.Record Namespace
The `TimeUtil` type exposes the following methods.
Methods

Public  Instance  Declared  XNA Framework Only

Protected  Static  Inherited  .NET Compact Framework Only

<table>
<thead>
<tr>
<th>Member</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Equals" /></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><img src="image" alt="Finalize" /></td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection.</td>
</tr>
<tr>
<td>(Inherited from <a href="#">Object</a>)</td>
<td></td>
</tr>
<tr>
<td><img src="image" alt="FromJavaTime" /></td>
<td>Convert from long with Java time (milliseconds since 1-Jan-1970) to Date object.</td>
</tr>
<tr>
<td><img src="image" alt="FromUnixTime" /></td>
<td>Convert from int with Unix time (seconds since 1-Jan-1970) to Date object.</td>
</tr>
<tr>
<td><img src="image" alt="FromVMSTime" /></td>
<td>Convert from long with VMS time (100 nanoseconds since 17-Nov-1858) to Date object.</td>
</tr>
<tr>
<td><img src="image" alt="GetHashCode" /></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><img src="image" alt="GetType" /></td>
<td>Gets the type of the current instance.</td>
</tr>
<tr>
<td>(Inherited from <a href="#">Object</a>)</td>
<td></td>
</tr>
<tr>
<td><img src="image" alt="MemberwiseClone" /></td>
<td>Creates a shallow copy of the current <a href="#">Object</a>.</td>
</tr>
<tr>
<td>(Inherited from <a href="#">Object</a>)</td>
<td></td>
</tr>
<tr>
<td><img src="image" alt="ToJavaTime" /></td>
<td>Convert from DateTime object to long with Java time (milliseconds since 1-Jan-1970).</td>
</tr>
<tr>
<td><img src="image" alt="ToUnixTime" /></td>
<td>Convert from Date object to int with Unix time (seconds since 1-Jan-1970).</td>
</tr>
<tr>
<td><img src="image" alt="ToVMSTime" /></td>
<td>Convert from Date object to long with VMS time (100 nanoseconds since 17-Nov-1858).</td>
</tr>
</tbody>
</table>
See Also

TimeUtil Class
Vajhoej.Record Namespace
See Also
A Sandcastle Documented Class Library
TimeUtil::FromJavaTime Method
Convert from long with Java time (milliseconds since 1-Jan-1970) to Date object.

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
**Syntax**

C#   Visual Basic   Visual C++

```csharp
public static DateTime FromJavaTime(
     long v
)
```

```vbnet
Public Shared Function FromJavaTime ( _
  v As Long _
) As DateTime
```

```cpp
public:
static DateTime FromJavaTime(
     long long v
)
```

**Parameters**

v

`Int64`

Java time.

**Return Value**

DateTime object
See Also

TimeUtil Class
Vajhoej.Record Namespace
See Also
A Sandcastle Documented Class Library
TimeUtil...::FromUnixTime Method
Convert from int with Unix time (seconds since 1-Jan-1970) to Date object.

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
## Syntax

**C#**  Visual Basic  Visual C++

```csharp
public static DateTime FromUnixTime(
    int v
)
```

```vbnet
Public Shared Function FromUnixTime ( _
    v As Integer _
) As DateTime
```

```cpp
public:
static DateTime FromUnixTime(
    int v
)
```

### Parameters

- **v**
  - `Int32`
  - Unix time.

### Return Value

DateTime object.
See Also

TimeUtil Class
Vajhoj.Record Namespace
See Also
A Sandcastle Documented Class Library
TimeUtil::FromVMSTime Method
Convert from long with VMS time (100 nanoseconds since 17-Nov-1858) to Date object.

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
Syntax

C#  Visual Basic  Visual C++

```csharp
public static DateTime FromVMSTime(
    long v
)
```

```vbnet
Public Shared Function FromVMSTime ( _
    v As Long _
) As DateTime
```

```cpp
public:
static DateTime FromVMSTime(
    long long v
)
```

**Parameters**

`v`

 Int64 VMS time.

**Return Value**

DateTime object.
See Also

TimeUtil Class
Vajhoej.Record Namespace
See Also
A Sandcastle Documented Class Library
TimeUtil...::ToJavaTime Method
Convert from DateTime object to long with Java time (milliseconds since 1-Jan-1970).

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
**Syntax**

C#   Visual Basic   Visual C++

```csharp
public static long ToJavaTime(
    DateTime dt
)
Public Shared Function ToJavaTime ( dt As DateTime _
) As Long

public:
static long long ToJavaTime(
    DateTime dt
)
```

**Parameters**

dt

  `DateTime`

  DateTime object.

**Return Value**

Java time.
See Also

TimeUtil Class
Vajhoej.Record Namespace
See Also
A Sandcastle Documented Class Library
TimeUtil...::ToUnixTime Method
Convert from Date object to int with Unix time (seconds since 1-Jan-1970).

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
Syntax

C#   Visual Basic   Visual C++

public static int ToUnixTime(
    DateTime dt
)

Public Shared Function ToUnixTime ( _
    dt As DateTime _
) As Integer

public:
static int ToUnixTime(
    DateTime dt
)

Parameters

dt
    DateTime
    DateTime object

Return Value

Unix time.
See Also

TimeUtil Class
Vajhoej.Record Namespace
See Also
A Sandcastle Documented Class Library
TimeUtil...::ToVMSTime Method
Convert from Date object to long with VMS time (100 nanoseconds since 17-Nov-1858).

Namespace:  Vajhoej.Record
Assembly:  Record (in Record.dll) Version: 0.0.0.0
Syntax

C#  Visual Basic  Visual C++

```csharp
public static long ToVMSTime(
    DateTime dt
)
```

```vbnet
Public Shared Function ToVMSTime ( _
    dt As DateTime _
) As Long
```

```cpp
public:
static long long ToVMSTime(
    DateTime dt
)
```

**Parameters**

`dt`  
`DateTime`  
DateTime object.

**Return Value**

VMS time.
See Also

TimeUtil Class
Vajhoej.Record Namespace
A Sandcastle Documented Class Library
Util Class
Utility class to process lists and to work with files instead of byte arrays.

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
Syntax

C#    Visual Basic    Visual C++

public class Util
Public Class Util
public ref class Util
### Members

#### All Members  Constructors  Methods

- [ ] Public  [ ] Instance  [ ] Declared  [ ] XNA Framework Only  [x]

- [ ] Protected  [s] Static  [s] Inherited  [.NET Compact Framework Only  [s]

<table>
<thead>
<tr>
<th>Member</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Util0000</strong></td>
<td>Initializes a new instance of the Util class</td>
</tr>
<tr>
<td><strong>CopyAll&lt;(Of &lt;&lt;'(T1, T2)&gt;)&gt;&gt; (Type, array&lt;Byte&gt;[][], Type, Util,;,;, TransformerConvert&lt;(Of &lt;&lt;'(T1, T2)&gt;&gt;))</strong></td>
<td>Convert array of struct in bytes into array of struct in bytes.</td>
</tr>
<tr>
<td><strong>CopyAll&lt;(Of &lt;&lt;'(T1, T2)&gt;&gt;) (Type, Stream, Type, Stream, Util,;,;, TransformerConvert&lt;(Of &lt;&lt;'(T1, T2)&gt;&gt;))</strong></td>
<td>Convert array of struct in stream into array of struct in stream.</td>
</tr>
<tr>
<td><strong>Equals(Object)</strong></td>
<td>Determines whether the specified Object is equal to the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>Finalize()</strong></td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (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>MemberwiseClone()</strong></td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>
ReadAll(Of <<(T)>>)(Type, array[Byte][][][]) Read array of struct in byte array into List of objects.

ReadAll(Of <<(T)>>)(Type, Stream, IList(Of <<(T)>>)) Read array of struct in stream into List of objects. Note: does not work with structs containing VARSTR fields and STRUCT fields.

ReadAll(Of <<(T)>>)(Type, Stream, Util:::ObjectHandlerProcess(Of <<(T)>>)) Read array of struct in stream and processes them by handler. Note: does not work with structs containing VARSTR fields and STRUCT fields.

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

WriteAll(Of <<(T)>>)(Type, IList(Of <<(T)>>)) Write List of objects into array of struct in byte array. Note: does not work with structs containing VARSTR fields and STRUCT fields.

WriteAll(Of <<(T)>>)(Type, IList(Of <<(T)>>), Stream) Write List of objects into array of struct in stream. Note: does not work with structs containing VARSTR fields and STRUCT fields.
Inheritance Hierarchy

System...Object
Vajhoej.Record...Util
See Also

Vajhoej.Record Namespace
Util Constructor

Initializes a new instance of the Util class

**Namespace:** Vajhoej.Record

**Assembly:** Record (in Record.dll) Version: 0.0.0.0
Syntax

C#   Visual Basic   Visual C++

public Util()
Public Sub New
public:
Util()
See Also

Util Class
Vajhoej.Record Namespace
The `Util` type exposes the following methods.
Methods

Public  Instance  Declared  XNA Framework Only

Protected  Static  Inherited  .NET Compact Framework Only

<table>
<thead>
<tr>
<th>Member</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CopyAll&lt;(Of &lt;&lt;(T1, T2)&gt;&gt;) (Type, array&lt;Byte&gt;[][], Type, Util::TransformerConvert&lt;(Of &lt;&lt;(T1, T2)&gt;&gt;)</td>
<td>Convert array of struct in bytes into array of struct in bytes.</td>
</tr>
<tr>
<td>CopyAll&lt;(Of &lt;&lt;(T1, T2)&gt;&gt;) (Type, Stream, Type, Stream, Util::TransformerConvert&lt;(Of &lt;&lt;(T1, T2)&gt;&gt;)</td>
<td>Convert array of struct in stream into array of struct in stream.</td>
</tr>
<tr>
<td>Equals(Object)</td>
<td>Determines whether the specified Object is equal to the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize()</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (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>MemberwiseClone()</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ReadAll&lt;(Of &lt;&lt;(T)&gt;&gt;) (Type, array&lt;Byte&gt;[][])</td>
<td>Read array of struct in byte array into List of objects.</td>
</tr>
<tr>
<td>ReadAll&lt;(Of &lt;&lt;(T)&gt;&gt;) (Type, Stream, Type, Stream, Util::TransformerConvert&lt;(Of &lt;&lt;(T)&gt;&gt;)</td>
<td>Read array of struct in stream into List of objects. Note: does not work with</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>Stream, IList&lt;Of &lt;&lt;'(T)&gt;&gt;)</td>
<td></td>
</tr>
<tr>
<td>ReadAll&lt;Of &lt;&lt;'(T)&gt;&gt;)(Type, Stream, Util::&lt;:;:::ObjectHandlerProcess&lt;Of &lt;&lt;'(T)&gt;&gt;)&gt;</td>
<td>Read array of struct in stream and processes them by handler. Note: does not work with structs containing VARSTR fields and STRUCT fields.</td>
</tr>
<tr>
<td>ToString()</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>WriteAll&lt;Of &lt;&lt;'(T)&gt;&gt;)(Type, IList&lt;Of &lt;&lt;'(T)&gt;&gt;)</td>
<td>Write List of objects into array of struct in byte array. Note: does not work with structs containing VARSTR fields and STRUCT fields.</td>
</tr>
<tr>
<td>WriteAll&lt;Of &lt;&lt;'(T)&gt;&gt;)(Type, IList&lt;Of &lt;&lt;'(T)&gt;&gt;), Stream)</td>
<td>Write List of objects into array of struct in stream. Note: does not work with structs containing VARSTR fields and STRUCT fields.</td>
</tr>
</tbody>
</table>
See Also

Util Class
Vajhoej.Record Namespace
See Also
A Sandcastle Documented Class Library
Util::CopyAll Method
### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>CopyAll&lt;(&lt;T1, T2&gt;&gt;)(Type, array&lt;Byte&gt;[][])</code>, <code>TransformerConvert&lt;(&lt;T1, T2&gt;&gt;)</code></td>
<td>Convert array of struct in bytes into array of struct in bytes.</td>
</tr>
<tr>
<td><code>CopyAll&lt;(&lt;T1, T2&gt;&gt;)(Type, Stream, Type, Stream, Util, Util,</code>, <code>TransformerConvert&lt;(&lt;T1, T2&gt;&gt;)</code></td>
<td>Convert array of struct in stream into array of struct in stream.</td>
</tr>
</tbody>
</table>
See Also

Util Class
Vajhoej.Record Namespace
See Also
A Sandcastle Documented Class Library
Util::CopyAll<(Of<T1, T2>)> Method (Type, array<Byte>[][],[], Type,
Util::...TransformerConvert<(Of<T1, T2>)>)
Convert array of struct in bytes into array of struct in bytes.

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
**Syntax**

C#  Visual Basic  Visual C++

```csharp
public static byte[] CopyAll<T1, T2>(
    Type t1,
    byte[] b,
    Type t2,
    Util::<T1, T2> cvt
)
where T1 : class, new()

Public Shared Function CopyAll(Of T1 As {Class, New}, T2) ( _
    t1 As Type, _
    b As Byte(), _
    t2 As Type, _
    cvt As Util::<T1, T2> _
) As Byte()

public:
    generic<typename T1, typename T2>
    where T1 : ref class, gcnew()
    static array<unsigned char>^ CopyAll(
        Type^ t1,
        array<unsigned char>^ b,
        Type^ t2,
        Util::<T1, T2>^ cvt
    )
```

**Parameters**

**t1**

*Type*
From type.

**b**

*array<Byte>[][][]*
From byte array.
Type
To type.

cvt

Util:::TransformerConvert<Of<('T1, T2)>>
Converter of objects.
Type Parameters

T1
   From type.
T2
   To type.

Return Value

To byte array.
### Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Vajhoej.Record::RecordException</code></td>
<td>If problem with record definition.</td>
</tr>
</tbody>
</table>
See Also

Util Class
CopyAll Overload
Vajhoej.Record Namespace
See Also
A Sandcastle Documented Class Library
Util..::CopyAll<(Of <"T1, T2">)> Method (Type, Stream, Type, Stream, Util..::..TransformerConvert<(Of <"T1, T2">)>)
Convert array of struct in stream into array of struct in stream.

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
### Syntax

**C#**  Visual Basic  Visual C++

```csharp
public static void CopyAll<T1, T2>(
    Type t1,
    Stream instm,
    Type t2,
    Stream outstm,
    Util...::...TransformerConvert<T1, T2> cvt
)
where T1 : class, new()

Public Shared Sub CopyAll(Of T1 As {Class, New}, T2) ( _
    t1 As Type, _
    instm As Stream, _
    t2 As Type, _
    outstm As Stream, _
    cvt As Util...::...TransformerConvert(Of T1, T2) _
)

class:

generic<typename T1, typename T2>
where T1 : ref class, gcnew()
static void CopyAll(
    Type^ t1,
    Stream^ instm,
    Type^ t2,
    Stream^ outstm,
    Util...::...TransformerConvert<T1, T2>^ cvt
)
```

### Parameters

**t1**

*Type*

- From type.

**instm**

*Stream*
From stream.

\[ t_2 \]

Type
To type.

\[ \text{outstm} \]

Stream
To stream.

\[ \text{cvt} \]

\text{Util}::\text{TransformerConvert}<\text{Of}<\langle T_1, T_2 \rangle\rangle>
Converter of objects.
Type Parameters

T1
  From type.
T2
  To type.
## Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vajhoej.Record:::RecordException</td>
<td>If problem with record definition.</td>
</tr>
<tr>
<td>System.IO:::IOException</td>
<td>If problem with stream.</td>
</tr>
</tbody>
</table>
See Also

Util Class
CopyAll Overload
Vajhoej.Record Namespace
See Also
A Sandcastle Documented Class Library
Util...::ReadAll Method
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ReadAll&lt;Of &lt;&lt;'(T)&gt;&gt;&gt;(Type, array&lt;Byte&gt;[][])</code></td>
<td>Read array of struct in byte array into List of objects.</td>
</tr>
<tr>
<td><code>ReadAll&lt;Of &lt;&lt;'(T)&gt;&gt;&gt;(Type, Stream, IList&lt;Of &lt;&lt;'(T)&gt;&gt;&gt;)</code></td>
<td>Read array of struct in stream into List of objects. Note: does not work with structs containing VARSTR fields and STRUCT fields.</td>
</tr>
<tr>
<td><code>ReadAll&lt;Of &lt;&lt;'(T)&gt;&gt;&gt;(Type, Stream, Util::ObjectHandlerProcess&lt;Of &lt;&lt;'(T)&gt;&gt;&gt;)</code></td>
<td>Read array of struct in stream and processes them by handler. Note: does not work with structs containing VARSTR fields and STRUCT fields.</td>
</tr>
</tbody>
</table>
See Also

Util Class
Vajhoej.Record Namespace
See Also
A Sandcastle Documented Class Library
Util...:ReadAll<(Of <(<<T>>)>)> Method (Type, array<Byte>[][])[]
Read array of struct in byte array into List of objects.

Namespace: Vajhoj.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
Syntax

C#    Visual Basic    Visual C++

public static IList<T> ReadAll<T>(
    Type t,
    byte[] b
)
where T : class, new()

Public Shared Function ReadAll(Of T As {Class, New}) ( _
    t As Type, _
    b As Byte() _
) As IList(Of T)

public:
generic<typename T>
where T : ref class, gcnew()
static IList<T>^ ReadAll(
    Type^ t,
    array<unsigned char>^ b
)

Parameters

t
   Type
   Type.

b
   array<Byte>[]][][]
   Byte array.
Type Parameters

T
  Type.

Return Value

List of objects.
### Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vajhoej.Record...RecordException</td>
<td>If problem with record definition.</td>
</tr>
</tbody>
</table>
See Also

Util Class
ReadAll Overload
Vajhoej.Record Namespace
Read array of struct in stream into List of objects. Note: does not work with structs containing VARSTR fields and STRUCT fields.

**Namespace:** Vajhoej.Record
**Assembly:** Record (in Record.dll) Version: 0.0.0.0
# Syntax

**C#**  Visual Basic  Visual C++

```csharp
public static void ReadAll<T>(
    Type t,
    Stream stm,
    IList<T> lst
)
where T : class, new()
```

```vbnet
Public Shared Sub ReadAll(Of T As {Class, New}) ( _
    t As Type, _
    stm As Stream, _
    lst As IList(Of T) _
)
```

```cpp
public:
    generic<typename T>
    where T : ref class, gcnew()
    static void ReadAll(
        Type^ t,
        Stream^ stm,
        IList<T>^ lst
    )
```

**Parameters**

**t**  
Type  
Type.

**stm**  
Stream  
Stream.

**lst**  
IList<T>()  
List of objects.
Type Parameters

T
  Type.
## Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vajhoej.Record...::...RecordException</td>
<td>If problem with record definition.</td>
</tr>
<tr>
<td>System.IO...::...IOException</td>
<td>If problem with stream.</td>
</tr>
</tbody>
</table>
See Also

Util Class
ReadAll Overload
Vajhoej.Record Namespace
See Also
A Sandcastle Documented Class Library

Util..:::ReadAll(Of (<'T'>))> Method (Type, Stream, Util..:::ObjectHandlerProcess(Of (<'T'>))>

Read array of struct in stream and processes them by handler. Note: does not work with structs containing VARSTR fields and STRUCT fields.

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
Syntax

C#    Visual Basic    Visual C++

public static void ReadAll<T>(
    Type t,
    Stream stm,
    Util::...ObjectHandlerProcess<T> ohp
)
where T : class, new()

Public Shared Sub ReadAll(Of T As {Class, New}) ( _
    t As Type, _
    stm As Stream, _
    ohp As Util::...ObjectHandlerProcess(Of T) _
)

public:
    generic<typename T>
    where T : ref class, gcnew()
    static void ReadAll(
        Type^ t,
        Stream^ stm,
        Util::...ObjectHandlerProcess<T^> ohp
    )

Parameters

t
    Type
    Type.

stm
    Stream
    Stream.

ohp
    Util::...ObjectHandlerProcess<(Of <(T)>)>
    Processor of objects.
Type Parameters

T

Type.
### Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vajhoej.Record...RecordException</td>
<td>If problem with record definition.</td>
</tr>
<tr>
<td>System.IO...IOException</td>
<td>If problem with stream.</td>
</tr>
</tbody>
</table>
See Also

Util Class
ReadAll Overload
Vajhoej.Record Namespace
C#  
Visual Basic  
Visual C++

See Also
A Sandcastle Documented Class Library Util...::WriteAll Method
### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WriteAll&lt;&lt;'(T)&gt;&gt;&gt;(Type, IList&lt;&lt;'(T)&gt;&gt;))</td>
<td>Write List of objects into array of struct in byte array. Note: does not work with structs containing VARSTR fields and STRUCT fields.</td>
</tr>
<tr>
<td>WriteAll&lt;&lt;(Type, IList&lt;&lt;'(T)&gt;&gt;, Stream&gt;&gt;</td>
<td>Write List of objects into array of struct in stream. Note: does not work with structs containing VARSTR fields and STRUCT fields.</td>
</tr>
</tbody>
</table>
See Also

Util Class
Vajhoej.Record Namespace
Write List of objects into array of struct in byte array. Note: does not work with structs containing VARSTR fields and STRUCT fields.

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
Syntax

C#   Visual Basic   Visual C++

public static byte[] WriteAll<T>(
    Type t,
    IList<T> lst
)

Public Shared Function WriteAll(Of T) ( _
    t As Type, _
    lst As IList(Of T) _
) As Byte()

public:
    generic<typename T>
    static array<unsigned char>^ WriteAll(
        Type^ t,
        IList<T>^ lst
    )

Parameters

t
    Type
    Type.

lst
    IList<Of ("<T>")>
    List of objects.
Type Parameters

T
  Type.

Return Value

Byte array.
## Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Vajhoej.Record...RecordException</code></td>
<td>If problem with record definition.</td>
</tr>
</tbody>
</table>
See Also

Util Class
WriteAll Overload
Vajhoej.Record Namespace
See Also
A Sandcastle Documented Class Library
Util..:::WriteAll<(Of <("T")>)> Method (Type, IList<(Of <("T")>)>, Stream)
Write List of objects into array of struct in stream. Note: does not work with structs containing VARSTR fields and STRUCT fields.

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
## Syntax

### C#  Visual Basic  Visual C++

```csharp
public static void WriteAll<T>(
    Type t,
    IList<T> lst,
    Stream stm)
```

```vbnet
Public Shared Sub WriteAll(Of T) ( _
    t As Type, _
    lst As IList(Of T), _
    stm As Stream _
)
```

```cpp
public:
    generic< typename T>
    static void WriteAll( 
        Type ^ t,
        IList<T> ^ lst,
        Stream ^ stm
    )
```

### Parameters

**t**

- **Type**
  - Type.

**lst**

- **IList<Of ("T")>**
  - list of object.

**stm**

- **Stream**
  - Stream.
Type Parameters

T
Type.
## Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vajhoej.Record...RecordException</td>
<td>If problem with record definition.</td>
</tr>
<tr>
<td>System.IO...IOException</td>
<td>If problem with stream.</td>
</tr>
</tbody>
</table>
See Also

Util Class
WriteAll Overload
Vajhoej.Record Namespace
C# | Visual Basic | Visual C++

See Also
A Sandcastle Documented Class Library Util......ObjectHandlerProcess<(Of <("T">))> Delegate Process object.

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
**Syntax**

**C#**

```
public delegate void ObjectHandlerProcess<T>(
    T o
)
```

**Visual Basic**

```
Public Delegate Sub ObjectHandlerProcess(Of T) ( _
    o As T _
)
```

**Visual C++**

```
generic<typename T>
public delegate void ObjectHandlerProcess(
    T o
)
```

**Parameters**

- `o`
  - `T`: Object.
Type Parameters

T
  Type.
See Also

Vajhoej.Record Namespace
See Also

A Sandcastle Documented Class Library

Util......TransformerConvert<(Of <(<'T1, T2'>)> )> Delegate

Convert object.

Namespace: Vajhoj.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
Syntax

C#   Visual Basic   Visual C++

public delegate T2 TransformerConvert<T1, T2>(
    T1 o
)

Public Delegate Function TransformerConvert(Of T1, T2) ( _
    o As T1 _
) As T2

generic<typename T1, typename T2>
public delegate T2 TransformerConvert(
    T1 o
)

Parameters

o
    T1
    From object.
Type Parameters

T1
   From type.
T2
   To type.

Return Value

To object.
See Also

Vajhoej.Record Namespace
A Sandcastle Documented Class Library

Util2 Class

Utility class to process lists and to work with files instead of byte arrays trying to work even with variable length structs.

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
Syntax

C#    Visual Basic    Visual C++

public class Util2
Public Class Util2
public ref class Util2
# Members

<table>
<thead>
<tr>
<th>All Members</th>
<th>Constructors</th>
<th>Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>Public</td>
<td>Instance</td>
<td>Declared</td>
</tr>
<tr>
<td>XNA Framework Only</td>
<td>Static</td>
<td>Inherited</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Member</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Util2()()()()</td>
<td>Initializes a new instance of the Util2 class</td>
</tr>
<tr>
<td>CopyAll&lt;(Of &lt;&lt;(T1, T2)&gt;&gt;)(Type, array&lt;Byte&gt;[][], Type, Util2,;,,;,,TransformerConvert&lt;(Of &lt;&lt;(T1, T2)&gt;&gt;), LengthProvider, MaxLengthProvider, ElementsProvider)</td>
<td>Convert array of struct in bytes into array of struct in bytes.</td>
</tr>
<tr>
<td>CopyAll&lt;(Of &lt;&lt;(T1, T2)&gt;&gt;)(Type, Stream, Type, Stream, Util2,;,,;,,TransformerConvert&lt;(Of &lt;&lt;(T1, T2)&gt;&gt;), LengthProvider, MaxLengthProvider, ElementsProvider)</td>
<td>Convert array of struct in stream into array of struct in stream.</td>
</tr>
<tr>
<td>Equals(Object)</td>
<td>Determines whether the specified Object is equal to the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize()()()()</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (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></td>
<td>Gets the type of the current</td>
</tr>
</tbody>
</table>
GetType()()

MemberwiseClone()()

ReadAll<(Of <<'(T)>>>)(Type, array<Byte>[][][], LengthProvider, MaxLengthProvider, ElementsProvider)

ReadAll<(Of <<'(T)>>>)(Type, Stream,_IList<(Of <<'(T)>>>), LengthProvider, MaxLengthProvider, ElementsProvider)

ReadAll<(Of <<'(T)>>>)(Type, Stream, Util2::..ObjectHandlerProcess<(Of <<'(T)>>>), LengthProvider, MaxLengthProvider, ElementsProvider)

ToString()()

WriteAll<(Of <<'(T)>>>)(Type, IList<(Of <<'(T)>>>), LengthProvider, MaxLengthProvider, ElementsProvider)

WriteAll<(Of <<'(T)>>>)(Type, IList<(Of <<'(T)>>>), Stream, LengthProvider, MaxLengthProvider, ElementsProvider)

instance.
(Inherited from Object.)

Creates a shallow copy of the current Object.
(Inherited from Object.)

Read array of struct in byte array into List of objects.

Read array of struct in stream into List of objects.

Read array of struct in stream and processes them by handler.

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

Write List of objects into array of struct in byte array.

Write List of objects into array of struct in stream.
Inheritance Hierarchy

System..::..Object
Vajhoej.Record..::..Util2
See Also

Vajhoej.Record Namespace
See Also
A Sandcastle Documented Class Library
Util2 Constructor

Initializes a new instance of the Util2 class

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
**Syntax**

C#  Visual Basic  Visual C++

public Util2()

Public Sub New

public:
Util2()
See Also

Util2 Class
Vajhoej.Record Namespace
See Also
A Sandcastle Documented Class Library
Util2 Methods

The Util2 type exposes the following methods.
# Methods

<table>
<thead>
<tr>
<th>Public</th>
<th>Instance</th>
<th>Static</th>
<th>Inherited</th>
<th>.NET Compact Framework Only</th>
</tr>
</thead>
<tbody>
<tr>
<td>☑️</td>
<td>☑️</td>
<td>✗</td>
<td>☑️</td>
<td>☑️</td>
</tr>
</tbody>
</table>

## Member

<table>
<thead>
<tr>
<th>Member</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CopyAll(\langle\text{Of}\ &lt;\langle\text{T1, T2}\rangle\rangle\rangle\langle\text{Type, array&lt;Byte&gt;[][]}\rangle), Type, \langle\text{Util2::...TransformerConvert&lt;\langle\text{Of, \langle\langle\text{T1, T2}\rangle\rangle\rangle\rangle}, LengthProvider, MaxLengthProvider, ElementsProvider}\rangle)</td>
<td></td>
</tr>
<tr>
<td>Convert array of struct in bytes into array of struct in bytes.</td>
<td></td>
</tr>
<tr>
<td>CopyAll(\langle\text{Of}\ &lt;\langle\text{T1, T2}\rangle\rangle\rangle\langle\text{Type, Stream, Type, Stream}\rangle), Type, \langle\text{Util2::...TransformerConvert&lt;\langle\text{Of, \langle\langle\text{T1, T2}\rangle\rangle\rangle\rangle}, LengthProvider, MaxLengthProvider, ElementsProvider}\rangle)</td>
<td></td>
</tr>
<tr>
<td>Convert array of struct in stream into array of struct in stream.</td>
<td></td>
</tr>
<tr>
<td>✔️ Equals(\text{Object})</td>
<td></td>
</tr>
<tr>
<td>Determines whether the specified \text{Object} is equal to the current \text{Object}. (Inherited from \text{Object}.)</td>
<td></td>
</tr>
<tr>
<td>✔️ Finalize())</td>
<td></td>
</tr>
<tr>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from \text{Object}.)</td>
<td></td>
</tr>
<tr>
<td>✔️ GetHashCode())</td>
<td></td>
</tr>
<tr>
<td>Serves as a hash function for a particular type. (Inherited from \text{Object}.)</td>
<td></td>
</tr>
<tr>
<td>✔️ GetType())</td>
<td></td>
</tr>
<tr>
<td>Gets the type of the current instance. (Inherited from \text{Object}.)</td>
<td></td>
</tr>
<tr>
<td>✔️ MemberwiseClone())</td>
<td></td>
</tr>
<tr>
<td>Creates a shallow copy of the current \text{Object}. (Inherited from \text{Object}.)</td>
<td></td>
</tr>
</tbody>
</table>
ReadAll<T>(Type, array[Byte][], LengthProvider, MaxLengthProvider, ElementsProvider)
ReadAll<T>(Type, Stream, IList<T>, LengthProvider, MaxLengthProvider, ElementsProvider)
ReadAll<T>(Type, Stream, Util2...ObjectHandlerProcess<T>, LengthProvider, MaxLengthProvider, ElementsProvider)

ToString() returns a string that represents the current object. (Inherited from Object.)

WriteAll<T>(IList<T>, LengthProvider, MaxLengthProvider, ElementsProvider)
WriteAll<T>(Stream, LengthProvider, MaxLengthProvider, ElementsProvider)
WriteAll<T>(IList<T>, Stream, LengthProvider, MaxLengthProvider, ElementsProvider)

Read array of struct in byte array into List of objects.

Read array of struct in stream into List of objects.

Read array of struct in stream and processes them by handler.

Write List of objects into array of struct in byte array.

Write List of objects into array of struct in stream.
See Also

Util2 Class
Vajhoej.Record Namespace
See Also
A Sandcastle Documented Class Library
Util2...::CopyAll Method
## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CopyAll&lt;Of &lt;&lt;(T1, T2)&gt;&gt;)(Type, array&lt;Byte&gt;[][], Type, Util2::TransformerConvert&lt;Of &lt;&lt;(T1, T2)&gt;&gt;), LengthProvider, MaxLengthProvider, ElementsProvider)</td>
<td>Convert array of struct in bytes into array of struct in bytes.</td>
</tr>
<tr>
<td>CopyAll&lt;Of &lt;&lt;(T1, T2)&gt;&gt;)(Type, Stream, Type, Stream, Util2::TransformerConvert&lt;Of &lt;&lt;(T1, T2)&gt;&gt;), LengthProvider, MaxLengthProvider, ElementsProvider)</td>
<td>Convert array of struct in stream into array of struct in stream.</td>
</tr>
</tbody>
</table>
See Also

Util2 Class
Vajhoej.Record Namespace
C#  
Visual Basic  
Visual C++

| See Also |
A Sandcastle Documented Class Library

Util2...::CopyAll<(Of <(<'T1, T2 '>)>)> Method (Type, array<Byte>[][], Type, Util2...::TransformerConvert<(Of <(<'T1, T2 '>)>), LengthProvider, MaxLengthProvider, ElementsProvider)

Convert array of struct in bytes into array of struct in bytes.

Namespace:  Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
**Syntax**

C#   Visual Basic   Visual C++

```csharp
public static byte[] CopyAll<T1, T2>(
    Type t1,
    byte[] b,
    Type t2,
    Util2...TransformerConvert<T1, T2> cvt,
    LengthProvider lenpvd,
    MaxLengthProvider maxlenpvd,
    ElementsProvider elmpvd
)
where T1 : class, new()
```

```vbnet
Public Shared Function CopyAll(Of T1 As {Class, New}, T2) ( _
    t1 As Type, _
    b As Byte(), _
    t2 As Type, _
    cvt As Util2...TransformerConvert(Of T1, T2), _
    lenpvd As LengthProvider, _
    maxlenpvd As MaxLengthProvider, _
    elmpvd As ElementsProvider _
) As Byte()
```

```cpp
public:
    generic<typename T1, typename T2>
where T1 : ref class, gcnew()
static array<unsigned char>^ CopyAll( 
    Type^ t1,
    array<unsigned char>^ b,
    Type^ t2,
    Util2...TransformerConvert<T1, T2>^ cvt,
    LengthProvider^ lenpvd,
    MaxLengthProvider^ maxlenpvd,
    ElementsProvider^ elmpvd
)
```

**Parameters**

*t1*
Type
From type.

b
array<Byte>[][]
From byte array.

t2
Type
To type.

cvt
   Util2...::TransformerConvert<(Of <"T1, T2">)>
Converter of objects.

lenpvd
   LengthProvider
Length provider.

maxlenpvd
   MaxLengthProvider
Max length provider.

ejmpvd
   ElementsProvider
Element provider.
Type Parameters

T1
  From type.
T2
  To type.

Return Value

To byte array.
**Exceptions**

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Vajhoej.Record::RecordException</code></td>
<td>If problem with record definition.</td>
</tr>
</tbody>
</table>
See Also

Util2 Class
CopyAll Overload
Vajhoej.Record Namespace
| See Also
A Sandcastle Documented Class Library
Util2::CopyAll<Of <('T1, T2)> Metodo (Type, Stream, Type, Stream,
Util2.....TransformerConvert<Of <('T1, T2)>>, LengthProvider,
MaxLengthProvider, ElementsProvider)

Convert array of struct in stream into array of struct in stream.

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
Syntax

C#  Visual Basic  Visual C++

public static void CopyAll<T1, T2>(
    Type t1,
    Stream instm,
    Type t2,
    Stream outstm,
    Util2..::.TransformerConvert<T1, T2> cvt,
    LengthProvider lenpvd,
    MaxLengthProvider maxlenpvd,
    ElementsProvider elmpvd
)

where T1 : class, new()

Public Shared Sub CopyAll(Of T1 As {Class, New}, T2) ( _
    t1 As Type, _
    instm As Stream, _
    t2 As Type, _
    outstm As Stream, _
    cvt As Util2..::.TransformerConvert(Of T1, T2), _
    lenpvd As LengthProvider, _
    maxlenpvd As MaxLengthProvider, _
    elmpvd As ElementsProvider _
)

public:
    generic<typename T1, typename T2>
    where T1 : ref class, gcnew()
    static void CopyAll(
        Type^ t1,
        Stream^ instm,
        Type^ t2,
        Stream^ outstm,
        Util2..::.TransformerConvert<T1, T2>^ cvt,
        LengthProvider^ lenpvd,
        MaxLengthProvider^ maxlenpvd,
        ElementsProvider^ elmpvd
    )

Parameters

t1
   Type
   From type.

instm
   Stream
   From stream.

t2
   Type
   To type.

outstm
   Stream
   To stream.

cvt
   Util2:::TransformerConvert<Of <(T1, T2)>>
   Converter of objects.

lenpvd
   LengthProvider
   Length provider.

maxlenpvd
   MaxLengthProvider
   Max length provider.

elmpvd
   ElementsProvider
   Element provider.
Type Parameters

T1
   From type.
T2
   To type.
## Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Vajhoej.Record...RecordException</code></td>
<td>If problem with record definition.</td>
</tr>
<tr>
<td><code>System.IO...IOException</code></td>
<td>If problem with stream.</td>
</tr>
</tbody>
</table>
See Also

Util2 Class
CopyAll Overload
Vajhoej.Record Namespace
See Also
A Sandcastle Documented Class Library
Util2...::ReadAll Method
# Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ReadAll&lt;(Of &lt;&lt;'(T)&gt;&gt;)(Type, array&lt;Byte&gt;[][], LengthProvider, MaxLengthProvider, ElementsProvider)</code></td>
<td>Read array of struct in byte array into List of objects.</td>
</tr>
<tr>
<td><code>ReadAll&lt;(Of &lt;&lt;'(T)&gt;&gt;)(Type, Stream, IList&lt;(Of &lt;&lt;'(T)&gt;&gt;), LengthProvider, MaxLengthProvider, ElementsProvider)</code></td>
<td>Read array of struct in stream into List of objects.</td>
</tr>
<tr>
<td><code>ReadAll&lt;(Of &lt;&lt;'(T)&gt;&gt;)(Type, Stream, Util2,::,ObjectHandlerProcess&lt;(Of &lt;&lt;'(T)&gt;&gt;), LengthProvider, MaxLengthProvider, ElementsProvider)</code></td>
<td>Read array of struct in stream and processes them by handler.</td>
</tr>
</tbody>
</table>
See Also

Util2 Class
Vajhoej.Record Namespace
A Sandcastle Documented Class Library

Util2...::ReadAll<Of <('T'>)>> Method (Type, array<Byte>[[]][[]],
LengthProvider, MaxLengthProvider, ElementsProvider)

Read array of struct in byte array into List of objects.

**Namespace:** Vajhoej.Record  
**Assembly:** Record (in Record.dll) Version: 0.0.0.0
Syntax

C#   Visual Basic   Visual C++

```csharp
public static IList<T> ReadAll<T>(
    Type t,
    byte[] b,
    LengthProvider lenpvd,
    MaxLengthProvider maxlenpvd,
    ElementsProvider elmpvd
)
```

where T : class, new()

```vbnet
Public Shared Function ReadAll(Of T As {Class, New}) ( _
    t As Type, _
    b As Byte(), _
    lenpvd As LengthProvider, _
    maxlenpvd As MaxLengthProvider, _
    elmpvd As ElementsProvider _) _
) As IList(Of T)
```

```cpp
public:
    generic<typename T>
    where T : ref class, gcnew()
    static IList<T>^ ReadAll(
        Type^ t,
        array<unsigned char>^ b,
        LengthProvider^ lenpvd,
        MaxLengthProvider^ maxlenpvd,
        ElementsProvider^ elmpvd
    )
```

### Parameters

- **t**
  - **Type**
    - Type.

- **b**
  - array<Byte>[]0[]0[]}
Byte array.

lenpvd
   **LengthProvider**
   Length provider.

maxlenpvd
   **MaxLengthProvider**
   Max length provider.

e1mpvd
   **ElementsProvider**
   Element provider.
Type Parameters

T
  Type.

Return Value

List of objects.
## Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vajhoej.Record::RecordException</td>
<td>If problem with record definition.</td>
</tr>
</tbody>
</table>
See Also

Util2 Class
ReadAll Overload
Vajhoej.Record Namespace
See Also
A Sandcastle Documented Class Library
Util2:::ReadAll<(Of <('T')>)> Method (Type, Stream, IList<(Of <('T')>)>, LengthProvider, MaxLengthProvider, ElementsProvider)

Read array of struct in stream into List of objects.

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
Syntax

C#   Visual Basic   Visual C++

public static void ReadAll<T>(
    Type t,
    Stream stm,
    IList<T> lst,
    LengthProvider lenpvd,
    MaxLengthProvider maxlenpvd,
    ElementsProvider elmpvd
)

where T : class, new()

Public Shared Sub ReadAll(Of T As {Class, New}) ( _
    t As Type, _
    stm As Stream, _
    lst As IList(Of T), _
    lenpvd As LengthProvider, _
    maxlenpvd As MaxLengthProvider, _
    elmpvd As ElementsProvider _
)

public:
    generic<typename T>
    where T : ref class, gcnew()
    static void ReadAll(
        Type^ t,
        Stream^ stm,
        IList<T>^ lst,
        LengthProvider^ lenpvd,
        MaxLengthProvider^ maxlenpvd,
        ElementsProvider^ elmpvd
    )

Parameters

t
    Type
    Type.
**stm**

Stream

**lst**

*IList<Of <("T")>>*

List of objects.

**lent**

*LengthProvider*

Length provider.

**maxlenpvd**

*MaxLengthProvider*

Max length provider.

**elmpvd**

*ElementsProvider*

Element provider.
Type Parameters

T
  Type.
## Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Vajhoej.Record...RecordException</code></td>
<td>If problem with record definition.</td>
</tr>
<tr>
<td><code>System.IO...IOException</code></td>
<td>If problem with stream.</td>
</tr>
</tbody>
</table>
See Also

Util2 Class
ReadAll Overload
Vajhoej.Record Namespace
A Sandcastle Documented Class Library

Util2:::ReadAll<(Of <('T')?>)> Method (Type, Stream,
Util2:::ObjectHandlerProcess<(Of <('T')?>), LengthProvider,
MaxLengthProvider, ElementsProvider)

Read array of struct in stream and processes them by handler.

**Namespace:** Vajhoej.Record
**Assembly:** Record (in Record.dll) Version: 0.0.0.0
C#  Visual Basic  Visual C++

public static void ReadAll<T>(
    Type t,
    Stream stm,
    Util2...ObjectHandlerProcess<T> ohp,
    LengthProvider lenpvd,
    MaxLengthProvider maxlenpvd,
    ElementsProvider elmpvd
)

where T : class, new()

Public Shared Sub ReadAll(Of T As {Class, New}) ( _
    t As Type, _
    stm As Stream, _
    ohp As Util2...ObjectHandlerProcess(Of T), _
    lenpvd As LengthProvider, _
    maxlenpvd As MaxLengthProvider, _
    elmpvd As ElementsProvider _
)

public:

generic<typename T>
where T : ref class, gcnew()

static void ReadAll(
    Type^ t,
    Stream^ stm,
    Util2...ObjectHandlerProcess<T>^ ohp,
    LengthProvider^ lenpvd,
    MaxLengthProvider^ maxlenpvd,
    ElementsProvider^ elmpvd
)

Parameters

t
    Type
    Type.
stm
Stream
Stream.

ohp
Util\ldots\ldots ObjectHandlerProcess\langle Of \langle\langle T\rangle\rangle\rangle
Processor of objects.

lenpvd
LengthProvider
Length provider.

maxlenpvd
MaxLengthProvider
Max length provider.

elmpvd
ElementsProvider
Element provider.
Type Parameters

T
  Type.
### Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Vajhoej.Record...RecordException</code></td>
<td>If problem with record definition.</td>
</tr>
<tr>
<td><code>System.IO....IOException</code></td>
<td>If problem with stream.</td>
</tr>
</tbody>
</table>
See Also

Util2 Class
ReadAll Overload
Vajhoej.Record Namespace
See Also
A Sandcastle Documented Class Library
Util2...::WriteAll Method
# Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>WriteAll&lt;(Of &lt;&lt;'(T)&gt;&gt;)(Type, IList&lt;(Of &lt;&lt;'(T)&gt;&gt;), LengthProvider, MaxLengthProvider, ElementsProvider)</code></td>
<td>Write List of objects into array of struct in byte array.</td>
</tr>
<tr>
<td><code>WriteAll&lt;(Of &lt;&lt;'(T)&gt;&gt;)(Type, IList&lt;(Of &lt;&lt;'(T)&gt;&gt;), Stream, LengthProvider, MaxLengthProvider, ElementsProvider)</code></td>
<td>Write List of objects into array of struct in stream.</td>
</tr>
</tbody>
</table>
See Also

Util2 Class
Vajhoej.Record Namespace
C# Visual Basic Visual C++

| See Also | A Sandcastle Documented Class Library

Util2::WriteAll(Of (<'T'>)) Method (Type, IList(Of (<'T'>)), LengthProvider, MaxLengthProvider, ElementsProvider)

Write List of objects into array of struct in byte array.

**Namespace:** Vajhoej.Record  
**Assembly:** Record (in Record.dll) Version: 0.0.0.0
Syntax

C#   Visual Basic   Visual C++

public static byte[] WriteAll<T>(
    Type t,
    IList<T> lst,
    LengthProvider lenpvd,
    MaxLengthProvider maxlenpvd,
    ElementsProvider elmpvd
)

Public Shared Function WriteAll(Of T) ( _
    t As Type, _
    lst As IList(Of T), _
    lenpvd As LengthProvider, _
    maxlenpvd As MaxLengthProvider, _
    elmpvd As ElementsProvider _
) As Byte()

public:
    generic<typename T>
    static array<unsigned char>^ WriteAll(
        Type^ t,
        IList<T>^ lst,
        LengthProvider^ lenpvd,
        MaxLengthProvider^ maxlenpvd,
        ElementsProvider^ elmpvd
    )

Parameters

t
    Type
    Type.

lst
    IList<(Of <(T)>)>)
    List of objects.
lenpvd
   **LengthProvider**
   Length provider.

maxlenpvd
   **MaxLengthProvider**
   Max length provider.

ejmpvd
   **ElementsProvider**
   Element provider.
Type Parameters

T
  Type.

Return Value

Byte array.
## Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vajhoej.Record::RecordException</td>
<td>If problem with record definition.</td>
</tr>
</tbody>
</table>
See Also

Util2 Class
WriteAll Overload
Vajhoej.Record Namespace
| See Also |
A Sandcastle Documented Class Library
Util2...:WriteAll<(Of <('T')>)> Method (Type, IList<(Of <('T')>)>, Stream, LengthProvider, MaxLengthProvider, ElementsProvider)
Write List of objects into array of struct in stream.

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
### Syntax

**C#**

```
public static void WriteAll<T>(
    Type t,
    IList<T> lst,
    Stream stm,
    LengthProvider lenpvd,
    MaxLengthProvider maxlenpvd,
    ElementsProvider elmpvd
)
```

**Visual Basic**

```vbnet
Public Shared Sub WriteAll(Of T) ( _
    t As Type, _
    lst As IList(Of T), _
    stm As Stream, _
    lenpvd As LengthProvider, _
    maxlenpvd As MaxLengthProvider, _
    elmpvd As ElementsProvider _
)
```

**Visual C++**

```
public:
    template<typename T>
    static void WriteAll( 
        Type^ t,
        IList<T>^ lst,
        Stream^ stm,
        LengthProvider^ lenpvd,
        MaxLengthProvider^ maxlenpvd,
        ElementsProvider^ elmpvd
    )
```

### Parameters

- **t**
  - *Type*
  - Type.

- **lst**
IList<Of<(<T>)>>
list of object.

stm
  Stream
  Stream.

lenpvdl
  LengthProvider
  Length provider.

maxlenpvd
  MaxLengthProvider
  Max length provider.

elem pvdl
  ElementsProvider
  Element provider.
Type Parameters

T
  Type.
## Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vajhoej.Record...RecordException</td>
<td>If problem with record definition.</td>
</tr>
<tr>
<td>System.IO...IOException</td>
<td>If problem with stream.</td>
</tr>
</tbody>
</table>
See Also

Util2 Class
WriteAll Overload
Vajhoej.Record Namespace
C#  
Visual Basic  
Visual C++

See Also
A Sandcastle Documented Class Library
Util2......ObjectHandlerProcess<(Of <("T")>)> Delegate
Process object.

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
Syntax

C#   Visual Basic   Visual C++

public delegate void ObjectHandlerProcess<T>(
    T o
)

Public Delegate Sub ObjectHandlerProcess(Of T) ( _
    o As T _
)

generic<typename T>
public delegate void ObjectHandlerProcess(
    T o
)

Parameters

o
    T
    Object.
Type Parameters

T
  Type.
See Also

Vajhoej.Record Namespace
A Sandcastle Documented Class Library
Util2......TransformerConvert<(Of <('T1, T2')>)> Delegate
Convert object.

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
Syntax

C#    Visual Basic    Visual C++

public delegate T2 TransformerConvert<T1, T2>(
    T1 o
)

Public Delegate Function TransformerConvert(Of T1, T2) ( _
    o As T1 _
) As T2

generic< typename T1, typename T2>
public delegate T2 TransformerConvert(
    T1 o
)

Parameters

o
    T1
From object.
**Type Parameters**

T1
   From type.

T2
   To type.

**Return Value**

To object.
See Also

Vajhoej.Record Namespace
A Sandcastle Documented Class Library

VAXFloatUtil Class

Class VAXFloatUtil converts between VAX floating point and IEEE floating point.

**Namespace:** Vajhoej.Record  
**Assembly:** Record (in Record.dll) Version: 0.0.0.0
Syntax

C# Visual Basic Visual C++

public class VAXFloatUtil
Public Class VAXFloatUtil
public ref class VAXFloatUtil
# Members

## All Members  Constructors  Methods

### Public  Instance  Declared  XNA Framework Only  Protected  Static  Inherited  .NET Compact Framework Only

### Member  Description

- **VAXFloatUtil**
  - Initializes a new instance of the VAXFloatUtil class

- **Equals**
  - Determines whether the specified *Object* is equal to the current *Object*.

- **F2S**
  - Convert from F floating to S floating.

- **G2T**
  - Convert from G floating to T floating.

- **Finalize**
  - Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection.

- **GetHashCode**
  - Serves as a hash function for a particular type.

- **GetType**
  - Gets the type of the current instance.

- **MemberwiseClone**
  - Creates a shallow copy of the current *Object*.

- **S2F**
  - Convert from S floating to F floating.

- **T2G**
  - Convert from T floating to G floating.

- **ToString**
  - Returns a string that represents the current object.
Inheritance Hierarchy

System...Object
Vajhoj.Record...VAXFloatUtil
See Also

Vajhoej.Record Namespace
C#  
Visual Basic  
Visual C++

| See Also |
A Sandcastle Documented Class Library
VAXFloatUtil Constructor
Initializes a new instance of the [VAXFloatUtil](#) class

**Namespace:** [Vajhoej.Record](#)
**Assembly:** Record (in Record.dll) Version: 0.0.0.0
Syntax

C# Visual Basic Visual C++

public VAXFloatUtil()
Public Sub New
public:
VAXFloatUtil()
See Also

VAXFloatUtil Class
Vajhoej.Record Namespace
The `VAXFloatUtil` type exposes the following methods.
# Methods

Public  Instance  Declared  XNA Framework Only

Protected  Static  Inherited  .NET Compact Framework Only

<table>
<thead>
<tr>
<th>Member</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals(Object)</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>F2S(UInt32)</strong></td>
<td>Convert from F floating to S floating.</td>
</tr>
<tr>
<td><strong>Finalize()</strong></td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>G2T(UInt64)</strong></td>
<td>Convert from G floating to T floating.</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>MemberwiseClone()</strong></td>
<td>Creates a shallow copy of the current <a href="#">Object</a>. (Inherited from <a href="#">Object</a>.)</td>
</tr>
<tr>
<td><strong>S2F(UInt32)</strong></td>
<td>Convert from S floating to F floating.</td>
</tr>
<tr>
<td><strong>T2G(UInt64)</strong></td>
<td>Convert from T floating to G floating.</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

VAXFloatUtil Class
Vajhoej.Record Namespace
A Sandcastle Documented Class Library
VAXFloatUtil::F2S Method
Convert from F floating to S floating.

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
## Syntax

C#     Visual Basic     Visual C++

public static uint F2S(
    uint v
)

Public Shared Function F2S ( _
    v As UInteger _
) As UInteger

default:
    public static unsigned int F2S(
        unsigned int v
    )

### Parameters

v

   _UINT32
    F floating.

### Return Value

S floating.
See Also

VAXFloatUtil Class
Vajhoej.Record Namespace
See Also
A Sandcastle Documented Class Library
VAXFloatUtil::G2T Method

Convert from G floating to T floating.

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
Syntax

C#   Visual Basic   Visual C++

```csharp
public static ulong G2T(
    ulong v
)
```

```vbnet
Public Shared Function G2T ( _
    v As ULong _
) As ULong
```

```cpp
public:
static unsigned_long_long G2T(
    unsigned_long_long v
)
```

**Parameters**

v  

**Return Value**

T floating.
See Also

VAXFloatUtil Class
Vajhoej.Record Namespace
See Also
A Sandcastle Documented Class Library
VAXFloatUtil:::S2F Method
Convert from S floating to F floating.

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
Syntax

C#     Visual Basic     Visual C++

public static uint S2F(
    uint v
)

Public Shared Function S2F ( _
    v As UInteger _
) As UInteger

public:
static unsigned int S2F(
    unsigned int v
)

Parameters

v
    UInt32
    S floating.

Return Value

F floating.
See Also

VAXFloatUtil Class
Vajhoej.Record Namespace
A Sandcastle Documented Class Library
VAXFloatUtil::T2G Method

Convert from T floating to G floating.

Namespace: Vajhoej.Record
Assembly: Record (in Record.dll) Version: 0.0.0.0
**Syntax**

**C#**   Visual Basic   Visual C++

```csharp
public static ulong T2G(
    ulong v
)
```

```vbnet
Public Shared Function T2G ( _
    v As ULong _
) As ULong
```

```cpp
public:
static unsigned long long T2G(
    unsigned long long v
)
```

**Parameters**

`v`  
`UInt64`  
T floating.

**Return Value**

G floating.
See Also

VAXFloatUtil Class
Vajhoej.Record Namespace