HandShakeComplete Delegate

References the callback method to be called when the protocol negotiation is completed.

[Visual Basic] Delegate Sub HandShakeComplete(ByVal error As System.Exception )

[C#] delegate void HandShakeComplete( Exception error );

Requirements


See Also


Copyright © 2002, The KPD-Team
IAasyncProxyResult Class

A class that implements the IAsyncResult interface. Objects from this class are returned by the BeginConnect method of the ProxySocket class.

For a list of all members of this type, see IAsyncProxyResult Members.

System.Object IAsyncProxyResult

[Visual Basic]
Class IAsyncProxyResult
    Implements IAsyncResult

[C#]
class IAsyncProxyResult : IAsyncResult

Requirements


See Also


Copyright © 2002, The KPD-Team
### IAsyncResult Members

**Public Instance Constructors**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IAsyncResult Constructor</td>
<td></td>
</tr>
</tbody>
</table>

**Public Instance Properties**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AsyncState</td>
<td>Gets an object that was passed as the state parameter of the BeginXXXX method call.</td>
</tr>
<tr>
<td>AsyncWaitHandle</td>
<td>The AsyncWaitHandle property returns the WaitHandle that can use to perform a WaitHandle.WaitOne or WaitAny or WaitAll. The object which implements IAsyncResult need not derive from the System.WaitHandle classes directly. The WaitHandle wraps its underlying synchronization primitive and should be signaled after the call is completed. This enables the client to wait for the call to complete instead polling. The Runtime supplies a number of waitable objects that mirror Win32 synchronization primitives e.g. ManualResetEvent, AutoResetEvent and Mutex. WaitHandle supplies methods that support waiting for such synchronization objects to become signaled with &quot;any&quot; or &quot;all&quot; semantics i.e. WaitHandle.WaitOne, WaitAny and WaitAll. Such methods are context aware to avoid deadlocks. The</td>
</tr>
</tbody>
</table>
AsyncWaitHandle can be allocated eagerly or on demand. It is the choice of the IAsyncResult implementer.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CompletedSynchronously</td>
<td>Gets a value that indicates whether the BeginXXXX call has been completed synchronously. If this is detected in the AsyncCallback delegate, it is probable that the thread that called BeginInvoke is the current thread.</td>
</tr>
<tr>
<td>IsCompleted</td>
<td>Gets a value that indicates whether the server has completed processing the call. It is illegal for the server to use any client supplied resources outside of the agreed upon sharing semantics after it sets the IsCompleted property to &quot;true&quot;. Thus, it is safe for the client to destroy the resources after IsCompleted property returns &quot;true&quot;.</td>
</tr>
</tbody>
</table>

**Public Instance Methods**

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Equals</code> (inherited from <code>Object</code>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><code>GetHashCode</code> (inherited from <code>Object</code>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><code>GetType</code> (inherited from <code>Object</code>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><code>ToString</code> (inherited from <code>Object</code>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
</tbody>
</table>
Protected Instance Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Finalize</code> (inherited from <code>Object</code>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><code>MemberwiseClone</code> (inherited from <code>Object</code>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
</tbody>
</table>

Internal Instance Fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>m_Completed</code></td>
<td>Used internally to represent the state of the asynchronous request</td>
</tr>
</tbody>
</table>

Internal Instance Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Init</code></td>
<td>Initializes the internal variables of this object</td>
</tr>
<tr>
<td><code>Reset</code></td>
<td>Initializes the internal variables of this object</td>
</tr>
</tbody>
</table>

Private Instance Fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>m_StateObject</code></td>
<td>Holds the value of the StateObject property.</td>
</tr>
<tr>
<td><code>m_WaitHandle</code></td>
<td>Holds the value of the WaitHandle property.</td>
</tr>
</tbody>
</table>

See Also


Copyright © 2002, The KPD-Team
### IAsyncProxyResult Constructor

<table>
<thead>
<tr>
<th>[Visual Basic] Public Sub New()</th>
</tr>
</thead>
<tbody>
<tr>
<td>[C#] public IAsyncProxyResult();</td>
</tr>
</tbody>
</table>

#### See Also

- IAsyncProxyResult Class
- IAsyncProxyResult Members

Copyright © 2002, The KPD-Team
IAsyncProxyResult Fields

The fields of the **IAsyncProxyResult** class are listed below. For a complete list of **IAsyncProxyResult** class members, see the [IAsyncProxyResult Members](#) topic.

### Internal Instance Fields

| m_Completed | Used internally to represent the state of the asynchronous request |

### Private Instance Fields

| m_StateObject | Holds the value of the StateObject property. |
| m_WaitHandle  | Holds the value of the WaitHandle property. |

See Also


Copyright © 2002, The KPD-Team
**IAsyncProxyResult.m_Completed Field**

Used internally to represent the state of the asynchronous request

**[Visual Basic]**

```vbnet
Friend m_Completed As Boolean
```

**[C#]**

```csharp
internal bool m_Completed;
```

See Also

- [IAsyncProxyResult Class](#)
- [IAsyncProxyResult Members](#)

Copyright © 2002, The KPD-Team
IAsyncProxyResult.m_StateObject Field

Holds the value of the StateObject property.

[Visual Basic] Private m_StateObject As Object

[C#] private object m_StateObject;

See Also


Copyright © 2002, The KPD-Team
IAsyncProxyResult.m_WaitHandle Field

Holds the value of the WaitHandle property.

[Visual Basic] Private m_WaitHandle As System.ManualResetEvent

[C#]
private ManualResetEvent m_WaitHandle;

See Also

Copyright © 2002, The KPD-Team
The properties of the IAsyncProxyResult class are listed below. For a complete list of IAsyncProxyResult class members, see the IAsyncProxyResult Members topic.

### Public Instance Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AsyncState</td>
<td>Gets an object that was passed as the state parameter of the BeginXXXX method call.</td>
</tr>
<tr>
<td>AsyncWaitHandle</td>
<td>The AsyncWaitHandle property returns the WaitHandle that can use to perform a WaitHandle.WaitOne or WaitAny or WaitAll. The object which implements IAsyncResult need not derive from the System.WaitHandle classes directly. The WaitHandle wraps its underlying synchronization primitive and should be signaled after the call is completed. This enables the client to wait for the call to complete instead polling. The Runtime supplies a number of waitable objects that mirror Win32 synchronization primitives e.g. ManualResetEvent, AutoResetEvent and Mutex. WaitHandle supplies methods that support waiting for such synchronization objects to become signaled with &quot;any&quot; or &quot;all&quot; semantics i.e. WaitHandle.WaitOne, WaitAny and WaitAll. Such methods are context aware to avoid deadlocks. The</td>
</tr>
</tbody>
</table>
AsyncWaitHandle can be allocated eagerly or on demand. It is the choice of the IAsyncResult implementer.

<table>
<thead>
<tr>
<th><strong>CompletedSynchronously</strong></th>
<th>Gets a value that indicates whether the BeginXXXX call has been completed synchronously. If this is detected in the AsyncCallback delegate, it is probable that the thread that called BeginInvoke is the current thread.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>IsCompleted</strong></td>
<td>Gets a value that indicates whether the server has completed processing the call. It is illegal for the server to use any client supplied resources outside of the agreed upon sharing semantics after it sets the IsCompleted property to &quot;true&quot;. Thus, it is safe for the client to destroy the resources after IsCompleted property returns &quot;true&quot;.</td>
</tr>
</tbody>
</table>

See Also


Copyright © 2002, The KPD-Team
**IAsyncProxyResult.AsyncState Property**

Gets an object that was passed as the state parameter of the BeginXXXX method call.

**[Visual Basic]**

```vbnet
NotOverridable Public ReadOnly Property AsyncState As Object
```

**[C#]**

```csharp
public object AsyncState {final get;}
```

**Property Value**

The object that was passed as the state parameter of the BeginXXXX method call.

**See Also**


Copyright © 2002, The KPD-Team
The AsyncWaitHandle property returns the WaitHandle that can use to perform a WaitHandle.WaitOne or WaitAny or WaitAll. The object which implements IAsyncResult need not derive from the System.WaitHandle classes directly. The WaitHandle wraps its underlying synchronization primitive and should be signaled after the call is completed. This enables the client to wait for the call to complete instead polling. The Runtime supplies a number of waitable objects that mirror Win32 synchronization primitives e.g. ManualResetEvent, AutoResetEvent and Mutex. WaitHandle supplies methods that support waiting for such synchronization objects to become signaled with "any" or "all" semantics i.e. WaitHandle.WaitOne, WaitAny and WaitAll. Such methods are context aware to avoid deadlocks. The AsyncWaitHandle can be allocated eagerly or on demand. It is the choice of the IAsyncResult implementer.

[Visual Basic] NotOverridable Public ReadOnly Property AsyncWaitHandle As [C#] public System.Threading.WaitHandle AsyncWaitHandle

Property Value

The WaitHandle associated with this asynchronous result.

See Also


Copyright © 2002, The KPD-Team
IAsyncResult.CompletedSynchronously Property

Gets a value that indicates whether the BeginXXXX call has been completed synchronously. If this is detected in the AsyncCallback delegate, it is probable that the thread that called BeginInvoke is the current thread.

[Visual Basic] NotOverridable Public ReadOnly Property CompletedSynchronously As Boolean

[C#]
public bool CompletedSynchronously {final get;}

Property Value

Returns false.

See Also


Copyright © 2002, The KPD-Team
**IAsyncProxyResult.IsCompleted Property**

Gets a value that indicates whether the server has completed processing the call. It is illegal for the server to use any client supplied resources outside of the agreed upon sharing semantics after it sets the IsCompleted property to "true". Thus, it is safe for the client to destroy the resources after IsCompleted property returns "true".

```[Visual Basic]NotOverridable Public ReadOnly Property IsCompleted As Boolean```

```[C#]public bool IsCompleted {final get;}```

**Property Value**

A boolean that indicates whether the server has completed processing the call.

**See Also**

- IAsyncProxyResult Class
- IAsyncProxyResult Members

Copyright © 2002, The KPD-Team
The methods of the **IAsyncProxyResult** class are listed below. For a complete list of **IAsyncProxyResult** class members, see the **IAsyncProxyResult Members** topic.

### Public Instance Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong> (inherited from <strong>Object</strong>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>GetHashCode</strong> (inherited from <strong>Object</strong>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>GetType</strong> (inherited from <strong>Object</strong>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>ToString</strong> (inherited from <strong>Object</strong>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
</tbody>
</table>

### Protected Instance Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Finalize</strong> (inherited from <strong>Object</strong>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>MemberwiseClone</strong> (inherited from <strong>Object</strong>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
</tbody>
</table>

### Internal Instance Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Init</strong></td>
<td>Initializes the internal variables of this object</td>
</tr>
<tr>
<td><strong>Reset</strong></td>
<td>Initializes the internal variables of this object</td>
</tr>
</tbody>
</table>

### See Also

IAsyncProxyResult.Init Method

Initializes the internal variables of this object

[Visual Basic]  
Friend Sub Init(_  
ByVal stateObject As Object _)  
)

[C#]  
internal void Init(  
object stateObject  
);

Parameters

stateObject  
An object that contains state information for this request.

See Also


Copyright © 2002, The KPD-Team
IAsyncProxyResult.Reset Method

Initializes the internal variables of this object

[Visual Basic] Friend Sub Reset()

[C#]
internal void Reset();

See Also


Copyright © 2002, The KPD-Team
Mentalis.org ProxySocket Documentation
ProxyException Class

The exception that is thrown when a proxy error occurs.
For a list of all members of this type, see ProxyException Members.

System.Object  Exception
ProxyException

[Visual Basic]
Public Class ProxyException
    Inherits Exception
    Implements ISerializable

[C#]
public class ProxyException : Exception, ISerializable

Requirements


See Also


Copyright © 2002, The KPD-Team
## ProxyException Members

### Public Static (Shared) Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Socks5ToString</code></td>
<td>Converts a SOCKS5 error number to a human readable string.</td>
</tr>
</tbody>
</table>

### Public Instance Constructors

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ProxyException</code></td>
<td>Overloaded. Initialize a new instance of the ProxyException class.</td>
</tr>
</tbody>
</table>

### Public Instance Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>HelpLink</code></td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><code>InnerException</code></td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><code>Message</code></td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><code>Source</code></td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><code>StackTrace</code></td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><code>TargetSite</code></td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
</tbody>
</table>

### Public Instance Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Equals</code></td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><code>GetBaseException</code></td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><code>GetHashCode</code></td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
</tbody>
</table>
**Object)** | Select the method name to go to the Microsoft documentation.
---|---
**GetObjectData** (inherited from **Exception**)
**GetTypeInfo** (inherited from **Object**)
**ToString** (inherited from **Exception**)

**Protected Instance Properties**

**HResult** (inherited from **System.Exception**)

**Protected Instance Methods**

**Finalize** (inherited from **Object**)

**MemberwiseClone** (inherited from **Object**)

**Internal Instance Fields**

**_HResult**

**_message**

**Internal Instance Methods**

**InternalToString** (inherited from **Exception**)

**See Also**


**Copyright © 2002, The KPD-Team**
ProxyException

Initializes a new instance of the ProxyException class.

Overload List

Initializes a new instance of the ProxyException class.

public ProxyException();

Initializes a new instance of the ProxyException class.

public ProxyException(string);

Initializes a new instance of the ProxyException class.

public ProxyException(int);

See Also


Copyright © 2002, The KPD-Team
ProxyException Constructor ()

Initializes a new instance of the ProxyException class.

```[Visual Basic] Overloads Public Sub New()
[Visual Basic] [C#] public ProxyException();
```

See Also


Copyright © 2002, The KPD-Team
ProxyException Constructor (String)

Initializes a new instance of the ProxyException class.

[Visual Basic] Overloads Public Sub New( _
    ByVal message As String _
)

[C#]
public ProxyException(
    string message
);

Parameters

message
    The message that describes the error.

See Also


Copyright © 2002, The KPD-Team
ProxyException Constructor (Int32)

Initializes a new instance of the ProxyException class.

[Visual Basic] Overloads Public Sub New( _
ByVal socks5Error As Integer _
)

[C#]
public ProxyException(  
    int socks5Error
);

Parameters

socks5Error
The error number returned by a SOCKS5 server.

See Also

ProxyException Class | ProxyException Members |

Copyright © 2002, The KPD-Team
ProxyException Fields

The fields of the ProxyException class are listed below. For a complete list of ProxyException class members, see the ProxyException Members topic.

Internal Instance Fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>_HResult</td>
<td></td>
</tr>
<tr>
<td>_message</td>
<td></td>
</tr>
</tbody>
</table>

See Also


Copyright © 2002, The KPD-Team
ProxyException._HResult Field

[Visual Basic]
Friend _HResult As Integer

[C#]
internal int _HResult;

See Also


Copyright © 2002, The KPD-Team
**ProxyException._message Field**

[Visual Basic] `Friend _message As String`

[C#] `internal string _message;`

**See Also**

[ProxyException Class] | [ProxyException Members] | [Org.Mentalis.Network.ProxySocket Namespace]

Copyright © 2002, The KPD-Team
The methods of the `ProxyException` class are listed below. For a complete list of `ProxyException` class members, see the `ProxyException Members` topic.

### Public Static (Shared) Methods

<table>
<thead>
<tr>
<th>Method Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Socks5ToString</td>
<td>Converts a SOCKS5 error number to a human readable string.</td>
</tr>
</tbody>
</table>

### Public Instance Methods

<table>
<thead>
<tr>
<th>Method Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td>GetBaseException</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td>GetObjectData</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td>GetType</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td>ToString</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
</tbody>
</table>

### Protected Instance Methods

<table>
<thead>
<tr>
<th>Method Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Finalize</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
</tbody>
</table>

### Internal Instance Methods

<table>
<thead>
<tr>
<th>Method Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InternalToString</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
</tbody>
</table>
Exception) the Microsoft documentation.

See Also


Copyright © 2002, The KPD-Team
Mentalis.org ProxySocket Documentation
ProxyException.Socks5ToString Method

Converts a SOCKS5 error number to a human readable string.

[Visual Basic]
Public Shared Function Socks5ToString(ByVal socks5Error As Integer) As String

[C#]
public static string Socks5ToString(int socks5Error);

Parameters

socks5Error
The error number returned by a SOCKS5 server.

Return Value

A string representation of the specified SOCKS5 error number.

See Also


Copyright © 2002, The KPD-Team
**ProxySocket Class**

Implements a Socket class that can connect through a SOCKS proxy server.

For a list of all members of this type, see [ProxySocket Members](#).

```csharp
public class ProxySocket : Socket, IDisposable
```

**Remarks**

This class implements SOCKS4[A] and SOCKS5. It does not, however, implement the BIND commands, so you cannot...

**Requirements**

**Namespace:** [Org.Mentalis.Network.ProxySocket Namespace](#)

**Assembly:** Org.Mentalis.Network.ProxySocket.dll

**See Also**


---

**Copyright © 2002, The KPD-Team**
## ProxySocket Members

### Public Instance Constructors

<table>
<thead>
<tr>
<th>ProxySocket</th>
<th>Overloaded. Initialize a new instance of the ProxySocket class.</th>
</tr>
</thead>
</table>

### Public Instance Properties

<table>
<thead>
<tr>
<th>AddressFamily (inherited from System.Net.Sockets.Socket)</th>
<th>Select the method name to go to the Microsoft documentation.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Available (inherited from System.Net.Sockets.Socket)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td>Blocking (inherited from System.Net.Sockets.Socket)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td>Connected (inherited from System.Net.Sockets.Socket)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td>Handle (inherited from System.Net.Sockets.Socket)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td>LocalEndPoint (inherited from System.Net.Sockets.Socket)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td>ProtocolType (inherited from System.Net.Sockets.Socket)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td>ProxyEndPoint</td>
<td>Gets or sets the EndPoint of the proxy server.</td>
</tr>
<tr>
<td>ProxyPass</td>
<td>Gets or sets the password to use when authenticating with the proxy.</td>
</tr>
<tr>
<td>ProxyType</td>
<td>Gets or sets the type of proxy server to use.</td>
</tr>
<tr>
<td>ProxyUser</td>
<td>Gets or sets the username to use when authenticating with the proxy.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>RemoteEndPoint</strong> (inherited from System.Net.Sockets.Socket)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>SocketType</strong> (inherited from System.Net.Sockets.Socket)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
</tbody>
</table>

**Public Instance Methods**

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Accept</strong> (inherited from Socket)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>BeginAccept</strong> (inherited from Socket)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>BeginConnect</strong></td>
<td>Overloaded. Begins an asynchronous request for a connection to a network device.</td>
</tr>
<tr>
<td><strong>BeginConnect</strong> (inherited from Socket)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>BeginReceive</strong> (inherited from Socket)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>BeginReceiveFrom</strong> (inherited from Socket)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>BeginSend</strong> (inherited from Socket)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>BeginSendTo</strong> (inherited from Socket)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>Bind</strong> (inherited from Socket)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>Close</strong> (inherited from Socket)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>Connect</strong></td>
<td>Overloaded. Establishes a connection to a remote device.</td>
</tr>
<tr>
<td><strong>Connect</strong> (inherited from Socket)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>EndAccept</strong> (inherited from Socket)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><code>EndConnect</code></td>
<td>Overloaded. Ends a pending asynchronous connection request.</td>
</tr>
<tr>
<td><code>EndConnect</code> (inherited from Socket)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><code>EndReceive</code> (inherited from Socket)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><code>EndReceiveFrom</code> (inherited from Socket)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><code>EndSend</code> (inherited from Socket)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><code>EndSendTo</code> (inherited from Socket)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><code>Equals</code> (inherited from <code>Object</code>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><code>GetHashCode</code> (inherited from Socket)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><code>GetSocketOption</code> (inherited from Socket)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><code>GetSocketOption</code> (inherited from Socket)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><code>GetSocketOption</code> (inherited from Socket)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><code>GetType</code> (inherited from <code>Object</code>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><code>IOControl</code> (inherited from Socket)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><code>Listen</code> (inherited from <code>Socket</code>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><code>Poll</code> (inherited from <code>Socket</code>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td>Method</td>
<td>Inheritance</td>
</tr>
<tr>
<td>----------------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>Receive</td>
<td>(inherited from Socket)</td>
</tr>
<tr>
<td>Receive</td>
<td>(inherited from Socket)</td>
</tr>
<tr>
<td>Receive</td>
<td>(inherited from Socket)</td>
</tr>
<tr>
<td>Receive</td>
<td>(inherited from Socket)</td>
</tr>
<tr>
<td>Receive</td>
<td>(inherited from Socket)</td>
</tr>
<tr>
<td>ReceiveFrom</td>
<td>(inherited from Socket)</td>
</tr>
<tr>
<td>ReceiveFrom</td>
<td>(inherited from Socket)</td>
</tr>
<tr>
<td>ReceiveFrom</td>
<td>(inherited from Socket)</td>
</tr>
<tr>
<td>ReceiveFrom</td>
<td>(inherited from Socket)</td>
</tr>
<tr>
<td>Send</td>
<td>(inherited from Socket)</td>
</tr>
<tr>
<td>Send</td>
<td>(inherited from Socket)</td>
</tr>
<tr>
<td>Send</td>
<td>(inherited from Socket)</td>
</tr>
<tr>
<td>Send</td>
<td>(inherited from Socket)</td>
</tr>
<tr>
<td>SendTo</td>
<td>(inherited from Socket)</td>
</tr>
<tr>
<td>SendTo</td>
<td>(inherited from Socket)</td>
</tr>
<tr>
<td>SendTo</td>
<td>(inherited from Socket)</td>
</tr>
<tr>
<td>SendTo</td>
<td>(inherited from Socket)</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>SendTo</strong> (inherited from <strong>Socket</strong>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>SetSocketOption</strong> (inherited from <strong>Socket</strong>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>SetSocketOption</strong> (inherited from <strong>Socket</strong>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>SetSocketOption</strong> (inherited from <strong>Socket</strong>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>SetSocketOption</strong> (inherited from <strong>Socket</strong>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>Shutdown</strong> (inherited from <strong>Socket</strong>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>ToString</strong> (inherited from <strong>Object</strong>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
</tbody>
</table>

**Protected Instance Methods**

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Dispose</strong> (inherited from <strong>Socket</strong>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>Finalize</strong> (inherited from <strong>Socket</strong>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>MemberwiseClone</strong> (inherited from <strong>Object</strong>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
</tbody>
</table>

**Internal Instance Fields**

<table>
<thead>
<tr>
<th>Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>incallback</td>
</tr>
<tr>
<td><strong>m_AsyncEvent</strong></td>
</tr>
<tr>
<td><strong>m_BlockEventBits</strong></td>
</tr>
<tr>
<td><strong>m_Handle</strong></td>
</tr>
<tr>
<td><strong>m_RightEndPoint</strong></td>
</tr>
</tbody>
</table>

**Internal Instance Methods**

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BeginDns</strong></td>
<td>Begins an asynchronous request to resolve a DNS host name or</td>
</tr>
</tbody>
</table>
IP address in dotted-quad notation to an IPAddress instance.

### Private Instance Fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CallBack</strong></td>
<td>Holds a pointer to the method that should be called when the Socket is connected to the remote device.</td>
</tr>
<tr>
<td><strong>m_AsyncResult</strong></td>
<td>Holds the value of the AsyncResult property.</td>
</tr>
<tr>
<td><strong>m_ProxyEndPoint</strong></td>
<td>Holds the value of the ProxyEndPoint property.</td>
</tr>
<tr>
<td><strong>m_ProxyPass</strong></td>
<td>Holds the value of the ProxyPass property.</td>
</tr>
<tr>
<td><strong>m_ProxyType</strong></td>
<td>Holds the value of the ProxyType property.</td>
</tr>
<tr>
<td><strong>m_ProxyUser</strong></td>
<td>Holds the value of the ProxyUser property.</td>
</tr>
<tr>
<td><strong>m_RemotePort</strong></td>
<td>Holds the value of the RemotePort property.</td>
</tr>
<tr>
<td><strong>m_State</strong></td>
<td>Holds the value of the State property.</td>
</tr>
<tr>
<td><strong>m_ToThrow</strong></td>
<td>Holds the value of the ToThrow property.</td>
</tr>
</tbody>
</table>

### Private Instance Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AsyncResult</strong></td>
<td>Gets or sets the asynchronous result object.</td>
</tr>
<tr>
<td><strong>RemotePort</strong></td>
<td>Gets or sets the remote port the user wants to connect to.</td>
</tr>
<tr>
<td><strong>State</strong></td>
<td>Gets or sets a user-defined object.</td>
</tr>
<tr>
<td><strong>ToThrow</strong></td>
<td>Gets or sets the exception to throw when the EndConnect method is called.</td>
</tr>
<tr>
<td>-------------</td>
<td>-------------------------------------------------------------------</td>
</tr>
</tbody>
</table>

### Private Instance Methods

<table>
<thead>
<tr>
<th><strong>OnConnect</strong></th>
<th>Called when the Socket is connected to the remote host.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>OnHandShakeComplete</strong></td>
<td>Called when the Socket has finished talking to the proxy server and is ready to relay data.</td>
</tr>
<tr>
<td><strong>OnResolved</strong></td>
<td>Called when the specified hostname has been resolved.</td>
</tr>
</tbody>
</table>

### Explicit Interface Implementations

<table>
<thead>
<tr>
<th><strong>IDisposable.Dispose</strong> (inherited from <strong>Socket</strong>)</th>
<th>Select the method name to go to the Microsoft documentation.</th>
</tr>
</thead>
</table>

### See Also


Copyright © 2002, The KPD-Team
**ProxySocket**

Initializes a new instance of the ProxySocket class.

**Overload List**

Initializes a new instance of the ProxySocket class.

```csharp
public ProxySocket(AddressFamily, SocketType, ProtocolType);
```

Initializes a new instance of the ProxySocket class.

```csharp
public ProxySocket(AddressFamily, SocketType, ProtocolType,string);
```

Initializes a new instance of the ProxySocket class.

```csharp
public ProxySocket(AddressFamily, SocketType, ProtocolType,string,string);
```

**See Also**


Copyright © 2002, The KPD-Team
ProxySocket Constructor (AddressFamily, SocketType, ProtocolType)

Initializes a new instance of the ProxySocket class.

[Visual Basic] Overloads Public Sub New( _
    ByVal addressFamily As System.Net.Sockets.AddressFamily,
    ByVal socketType As System.Net.Sockets.SocketType,
    ByVal protocolType As System.Net.Sockets.ProtocolType)

[C#]
public ProxySocket(
    AddressFamily addressFamily,
    SocketType socketType,
    ProtocolType protocolType
);

Parameters

addressFamily
One of the AddressFamily values.

socketType
One of the SocketType values.

protocolType
One of the ProtocolType values.

Exceptions

<table>
<thead>
<tr>
<th>Exception Type</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>System.Net.Sockets.SocketException</td>
<td>The combination of addressFamily, socketType, and protocolType results in an invalid socket.</td>
</tr>
</tbody>
</table>

See Also

ProxySocket Class | ProxySocket Members |

Copyright © 2002, The KPD-Team
Mentalis.org ProxySocket Documentation
ProxySocket Constructor (AddressFamily, SocketType, ProtocolType, String)

Initializes a new instance of the ProxySocket class.

[Visual Basic] Overloads Public Sub New(_
    ByVal addressFamily As System.Net.Sockets.AddressFamily,
    ByVal socketType As System.Net.Sockets.SocketType,
    ByVal protocolType As System.Net.Sockets.ProtocolType,
    ByVal proxyUsername As String _
)

[C#]
public ProxySocket(
    AddressFamily addressFamily,
    SocketType socketType,
    ProtocolType protocolType,
    string proxyUsername
);

Parameters

addressFamily
    One of the AddressFamily values.

socketType
    One of the SocketType values.

protocolType
    One of the ProtocolType values.

proxyUsername
    The username to use when authenticating with the proxy server.

Exceptions

<table>
<thead>
<tr>
<th>Exception Type</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>System.Net.Sockets.SocketException</td>
<td>The combination of addressFamily, socketType, and protocolType results in</td>
</tr>
</tbody>
</table>
an invalid socket.

| System.ArgumentNullException | proxyUsername is null. |

See Also


Copyright © 2002, The KPD-Team
Mentalis.org ProxySocket Documentation
ProxySocket Constructor (AddressFamily, SocketType, ProtocolType, String, String)

Initializes a new instance of the ProxySocket class.

**[Visual Basic]**
```vbnet
Overloads Public Sub New( _
    ByVal addressFamily As System.Net.Sockets.AddressFamily, _
    ByVal socketType As System.Net.Sockets.SocketType, _
    ByVal protocolType As System.Net.Sockets.ProtocolType, _
    ByVal proxyUsername As String, _
    ByVal proxyPassword As String)
```

**[C#]**
```csharp
public ProxySocket(
    AddressFamily addressFamily, _
    SocketType socketType, _
    ProtocolType protocolType, _
    string proxyUsername, _
    string proxyPassword)
```

**Parameters**

- **addressFamily**
  One of the AddressFamily values.

- **socketType**
  One of the SocketType values.

- **protocolType**
  One of the ProtocolType values.

- **proxyUsername**
  The username to use when authenticating with the proxy server.

- **proxyPassword**
  The password to use when authenticating with the proxy server.

**Exceptions**
<table>
<thead>
<tr>
<th>Exception Type</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>System.Net.Sockets.SocketException</td>
<td>The combination of addressFamily, socketType, and protocolType results in an invalid socket.</td>
</tr>
<tr>
<td>System.ArgumentNullException</td>
<td>proxyUsername -or- proxyPassword is null.</td>
</tr>
</tbody>
</table>

See Also


Copyright © 2002, The KPD-Team
ProxySocket Fields

The fields of the ProxySocket class are listed below. For a complete list of ProxySocket class members, see the ProxySocket Members topic.

**Internal Instance Fields**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>incallback</td>
<td></td>
</tr>
<tr>
<td>m_AsyncEvent</td>
<td></td>
</tr>
<tr>
<td>m_BlockEventBits</td>
<td></td>
</tr>
<tr>
<td>m_Handle</td>
<td></td>
</tr>
<tr>
<td>m_RightEndPoint</td>
<td></td>
</tr>
</tbody>
</table>

**Private Instance Fields**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CallBack</td>
<td>Holds a pointer to the method that should be called when the Socket is connected to the remote device.</td>
</tr>
<tr>
<td>m_AsyncResult</td>
<td>Holds the value of the AsyncResult property.</td>
</tr>
<tr>
<td>m_ProxyEndPoint</td>
<td>Holds the value of the ProxyEndPoint property.</td>
</tr>
<tr>
<td>m_ProxyPass</td>
<td>Holds the value of the ProxyPass property.</td>
</tr>
<tr>
<td>m_ProxyType</td>
<td>Holds the value of the ProxyType property.</td>
</tr>
<tr>
<td>m_ProxyUser</td>
<td>Holds the value of the ProxyUser property.</td>
</tr>
<tr>
<td>m_RemotePort</td>
<td>Holds the value of the RemotePort property.</td>
</tr>
<tr>
<td>m_State</td>
<td>Holds the value of the State property.</td>
</tr>
</tbody>
</table>
m_ToThrow

Holds the value of the ToThrow property.

See Also


Copyright © 2002, The KPD-Team
ProxySocket.CallBack Field

Holds a pointer to the method that should be called when the Socket is connected to the remote device.

[Visual Basic] Private CallBack As System AsyncCallback

[C#]
private AsyncCallback CallBack;

See Also


Copyright © 2002, The KPD-Team
ProxySocket.incallback Field

[Visual Basic] Friend incallback As Boolean

[C#]
internal bool incallback;

See Also

Copyright © 2002, The KPD-Team
ProxySocket.m_AsyncEvent Field

[Visual Basic] Friend m_AsyncEvent As System.Threading.AutoResetEvent

[C#] internal AutoResetEvent m_AsyncEvent;

See Also


Copyright © 2002, The KPD-Team
ProxySocket.m_AsyncResult Field

Holds the value of the AsyncResult property.


[C#] private IAsyncProxyResult m_AsyncResult;

See Also


Copyright © 2002, The KPD-Team
ProxySocket.m_BlockEventBits Field


[C#]
internal AsyncEventBits m_BlockEventBits;

See Also


Copyright © 2002, The KPD-Team
Mentalis.org ProxySocket Documentation
ProxySocket.m_Handle Field

[Visual Basic] Friend m_Handle As System.IntPtr

[C#]
internal IntPtr m_Handle;

See Also


Copyright © 2002, The KPD-Team
ProxySocket.m_ProxyEndPoint Field

Holds the value of the ProxyEndPoint property.

[Visual Basic] Private m_ProxyEndPoint As System.Net.IPEndPoint

[C#] private IPEndPoint m_ProxyEndPoint;

See Also


Copyright © 2002, The KPD-Team
**ProxySocket.m_ProxyPass Field**

Holds the value of the ProxyPass property.

[Visual Basic] `Private m_ProxyPass As String`

[C#] `private string m_ProxyPass;`

See Also


Copyright © 2002, The KPD-Team
**ProxySocket.m_ProxyType Field**

Holds the value of the ProxyType property.

[Visual Basic] 
```plaintext
Private m_ProxyType As Org.Mentalis.Network.ProxySocket.ProxyTypes
```  
[C#] 
```plaintext
private ProxyTypes m_ProxyType;
```  

See Also


Copyright © 2002, The KPD-Team
ProxySocket.m_ProxyUser Field

Holds the value of the ProxyUser property.

[Visual Basic] Private m_ProxyUser As String

[C#] private string m_ProxyUser;

See Also


Copyright © 2002, The KPD-Team
ProxySocket.m_RemotePort Field

Holds the value of the RemotePort property.

[Visual Basic] Private m_RemotePort As Integer

[C#]
private int m_RemotePort;

See Also


Copyright © 2002, The KPD-Team
ProxySocket.m_RightEndPoint Field

[Visual Basic] Friend m_RightEndPoint As System.Net.EndPoint

[C#]
internal EndPoint m_RightEndPoint;

See Also


Copyright © 2002, The KPD-Team
ProxySocket.m_State Field

Holds the value of the State property.

[Visual Basic] Private m_State As Object

[C#]
private object m_State;

See Also


Copyright © 2002, The KPD-Team
ProxySocket.m_ToThrow Field

Holds the value of the ToThrow property.

[Visual Basic] Private m_ToThrow As System.Exception

[C#]
private Exception m_ToThrow;

See Also


Copyright © 2002, The KPD-Team
Mentalis.org ProxySocket Documentation
The properties of the **ProxySocket** class are listed below. For a complete list of **ProxySocket** class members, see the **ProxySocket Members** topic.

**Public Instance Properties**

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AddressFamily</strong></td>
<td>(inherited from <code>System.Net.Sockets.Socket</code>) Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>Available</strong></td>
<td>(inherited from <code>System.Net.Sockets.Socket</code>) Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>Blocking</strong></td>
<td>(inherited from <code>System.Net.Sockets.Socket</code>) Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>Connected</strong></td>
<td>(inherited from <code>System.Net.Sockets.Socket</code>) Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>Handle</strong></td>
<td>(inherited from <code>System.Net.Sockets.Socket</code>) Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>LocalEndPoint</strong></td>
<td>(inherited from <code>System.Net.Sockets.Socket</code>) Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>ProtocolType</strong></td>
<td>(inherited from <code>System.Net.Sockets.Socket</code>) Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>ProxyEndPoint</strong></td>
<td>Gets or sets the EndPoint of the proxy server.</td>
</tr>
<tr>
<td><strong>ProxyPass</strong></td>
<td>Gets or sets the password to use when authenticating with the proxy.</td>
</tr>
<tr>
<td><strong>ProxyType</strong></td>
<td>Gets or sets the type of proxy server to use.</td>
</tr>
<tr>
<td><strong>ProxyUser</strong></td>
<td>Gets or sets the username to use when authenticating with the proxy.</td>
</tr>
<tr>
<td><strong>RemoteEndPoint</strong></td>
<td>(inherited from <code>System.Net.Sockets.Socket</code>) Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>SocketType</strong> (inherited from System.Net.Sockets.Socket)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td>----------------------------------------------------------</td>
<td>---------------------------------------------------------------</td>
</tr>
</tbody>
</table>

**Private Instance Properties**

<table>
<thead>
<tr>
<th><strong>AsyncResult</strong></th>
<th>Gets or sets the asynchronous result object.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>RemotePort</strong></td>
<td>Gets or sets the remote port the user wants to connect to.</td>
</tr>
<tr>
<td><strong>State</strong></td>
<td>Gets or sets a user-defined object.</td>
</tr>
<tr>
<td><strong>ToThrow</strong></td>
<td>Gets or sets the exception to throw when the EndConnect method is called.</td>
</tr>
</tbody>
</table>

**See Also**


Copyright © 2002, The KPD-Team
ProxySocketAsyncResult Property

Gets or sets the asynchronous result object.

[Visual Basic] Private Property AsyncResult As [C#] private IAsyncProxyResult AsyncResult {get; set;}

Property Value

An instance of the IAsyncProxyResult class.

See Also


Copyright © 2002, The KPD-Team
ProxySocket.ProxyEndPoint Property

Gets or sets the EndPoint of the proxy server.

[Visual Basic]
Public Property ProxyEndPoint

[C#]
public System.Net.IPEndPoint ProxyEndPoint {get; set;}

Property Value

An IPEndPoint object that holds the IP address and the port of the proxy server.

See Also


Copyright © 2002, The KPD-Team
ProxySocket.ProxyPass Property

Gets or sets the password to use when authenticating with the proxy.

[Visual Basic] Public Property ProxyPass As String

[C#] public string ProxyPass {get; set;}

Property Value

A string that holds the password that's used when authenticating with the proxy.

Exceptions

<table>
<thead>
<tr>
<th>Exception Type</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>System.ArgumentNullException</td>
<td>The specified value is null.</td>
</tr>
</tbody>
</table>

See Also


Copyright © 2002, The KPD-Team
ProxySocket.ProxyType Property

Gets or sets the type of proxy server to use.

[Visual Basic]
Public Property ProxyType As Org.Mentalis.Network.ProxySocket.ProxyTypes

[C#]
public ProxyTypes ProxyType {get; set;}

Property Value

One of the ProxyTypes values.

See Also


Copyright © 2002, The KPD-Team
ProxySocket.ProxyUser Property

Gets or sets the username to use when authenticating with the proxy.

[Visual Basic] Public Property ProxyUser As String

[C#] public string ProxyUser {get; set;}

Property Value

A string that holds the username that's used when authenticating with the proxy.

Exceptions

<table>
<thead>
<tr>
<th>Exception Type</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>System.ArgumentNullException</td>
<td>The specified value is null.</td>
</tr>
</tbody>
</table>

See Also


Copyright © 2002, The KPD-Team
Mentalis.org ProxySocket Documentation
ProxySocket.RemotePort Property

Gets or sets the remote port the user wants to connect to.

[Visual Basic] Private Property RemotePort As

[C#] private int RemotePort {get; set;}

Property Value

An integer that specifies the port the user wants to connect to.

See Also


Copyright © 2002, The KPD-Team
ProxySocket.State Property

Gets or sets a user-defined object.

[Visual Basic]
Private Property State As Object

[C#]
private object State {get; set;}

Property Value

The user-defined object.

See Also


Copyright © 2002, The KPD-Team
ProxySocket.ToThrow Property

Gets or sets the exception to throw when the EndConnect method is called.

[Visual Basic] Private Property ToThrow As System.Exception

[C#] private System.Exception ToThrow {get; set;}

Property Value

An instance of the Exception class (or subclasses of Exception).

See Also


Copyright © 2002, The KPD-Team
## ProxySocket Methods

The methods of the `ProxySocket` class are listed below. For a complete list of `ProxySocket` class members, see the [ProxySocket Members](#) topic.

### Public Instance Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Accept</strong> (inherited from <code>Socket</code>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>BeginAccept</strong> (inherited from <code>Socket</code>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>BeginConnect</strong></td>
<td>Overloaded. Begins an asynchronous request for a connection to a network device.</td>
</tr>
<tr>
<td><strong>BeginConnect</strong> (inherited from <code>Socket</code>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>BeginReceive</strong> (inherited from <code>Socket</code>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>BeginReceiveFrom</strong> (inherited from <code>Socket</code>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>BeginSend</strong> (inherited from <code>Socket</code>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>BeginSendTo</strong> (inherited from <code>Socket</code>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>Bind</strong> (inherited from <code>Socket</code>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>Close</strong> (inherited from <code>Socket</code>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>Connect</strong></td>
<td>Overloaded. Establishes a connection to a remote device.</td>
</tr>
<tr>
<td><strong>Connect</strong> (inherited from <code>Socket</code>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td>Method</td>
<td>Inherited From</td>
</tr>
<tr>
<td>----------------</td>
<td>--------------------------------</td>
</tr>
<tr>
<td><code>EndAccept</code></td>
<td><code>Socket</code></td>
</tr>
<tr>
<td><code>EndConnect</code></td>
<td></td>
</tr>
<tr>
<td><code>EndConnect</code></td>
<td><code>Socket</code></td>
</tr>
<tr>
<td><code>EndReceive</code></td>
<td><code>Socket</code></td>
</tr>
<tr>
<td><code>EndReceiveFrom</code></td>
<td><code>Socket</code></td>
</tr>
<tr>
<td><code>EndSend</code></td>
<td><code>Socket</code></td>
</tr>
<tr>
<td><code>EndSendTo</code></td>
<td><code>Socket</code></td>
</tr>
<tr>
<td><code>Equals</code></td>
<td><code>Object</code></td>
</tr>
<tr>
<td><code>GetHashCode</code></td>
<td><code>Socket</code></td>
</tr>
<tr>
<td><code>GetSocketOption</code></td>
<td><code>Socket</code></td>
</tr>
<tr>
<td><code>GetSocketOption</code></td>
<td><code>Socket</code></td>
</tr>
<tr>
<td><code>GetSocketOption</code></td>
<td><code>Socket</code></td>
</tr>
<tr>
<td><code>GetType</code></td>
<td><code>Object</code></td>
</tr>
<tr>
<td><code>IOControl</code></td>
<td><code>Socket</code></td>
</tr>
<tr>
<td><code>Listen</code></td>
<td><code>Socket</code></td>
</tr>
<tr>
<td><strong>Poll</strong> (inherited from <strong>Socket</strong>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>Receive</strong> (inherited from <strong>Socket</strong>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>Receive</strong> (inherited from <strong>Socket</strong>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>Receive</strong> (inherited from <strong>Socket</strong>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>Receive</strong> (inherited from <strong>Socket</strong>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>Receive</strong> (inherited from <strong>Socket</strong>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>Receive</strong> (inherited from <strong>Socket</strong>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>Receive</strong> (inherited from <strong>Socket</strong>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>Receive</strong> (inherited from <strong>Socket</strong>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>Receive</strong> (inherited from <strong>Socket</strong>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>Receive</strong> (inherited from <strong>Socket</strong>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>Receive</strong> (inherited from <strong>Socket</strong>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>Receive</strong> (inherited from <strong>Socket</strong>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>Receive</strong> (inherited from <strong>Socket</strong>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>Receive</strong> (inherited from <strong>Socket</strong>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>Receive</strong> (inherited from <strong>Socket</strong>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>Receive</strong> (inherited from <strong>Socket</strong>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>Receive</strong> (inherited from <strong>Socket</strong>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>Receive</strong> (inherited from <strong>Socket</strong>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>Receive</strong> (inherited from <strong>Socket</strong>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>Receive</strong> (inherited from <strong>Socket</strong>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>Receive</strong> (inherited from <strong>Socket</strong>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>Receive</strong> (inherited from <strong>Socket</strong>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>Receive</strong> (inherited from <strong>Socket</strong>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>Receive</strong> (inherited from <strong>Socket</strong>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>Receive</strong> (inherited from <strong>Socket</strong>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td>Method</td>
<td>Summary</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>SendTo</strong> (inherited from <strong>Socket</strong>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>SetSocketOption</strong> (inherited from <strong>Socket</strong>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>SetSocketOption</strong> (inherited from <strong>Socket</strong>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>SetSocketOption</strong> (inherited from <strong>Socket</strong>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>SetSocketOption</strong> (inherited from <strong>Socket</strong>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>Shutdown</strong> (inherited from <strong>Socket</strong>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>ToString</strong> (inherited from <strong>Object</strong>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
</tbody>
</table>

**Protected Instance Methods**

<table>
<thead>
<tr>
<th>Method</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Dispose</strong> (inherited from <strong>Socket</strong>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>Finalize</strong> (inherited from <strong>Socket</strong>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>MemberwiseClone</strong> (inherited from <strong>Object</strong>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
</tbody>
</table>

**Internal Instance Methods**

<table>
<thead>
<tr>
<th>Method</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BeginDns</strong></td>
<td>Begins an asynchronous request to resolve a DNS host name or IP address in dotted-quad notation to an IPAddress instance.</td>
</tr>
</tbody>
</table>

**Private Instance Methods**

<table>
<thead>
<tr>
<th>Method</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>OnConnect</strong></td>
<td>Called when the Socket is connected to the remote host.</td>
</tr>
<tr>
<td><strong>OnHandShakeComplete</strong></td>
<td>Called when the Socket has finished talking to the proxy server and is ready to relay data.</td>
</tr>
<tr>
<td>--------------------------</td>
<td>------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>OnResolved</strong></td>
<td>Called when the specified hostname has been resolved.</td>
</tr>
</tbody>
</table>

### Explicit Interface Implementations

| **IDisposable.Dispose** (inherited from **Socket**) | Select the method name to go to the Microsoft documentation. |

### See Also


**Copyright © 2002, The KPD-Team**
ProxySocket.BeginConnect

Begins an asynchronous request for a connection to a network device.

Overload List

Begins an asynchronous request for a connection to a network device.

public IAsyncResult BeginConnect(EndPoint, AsyncCallback, object);

Begins an asynchronous request for a connection to a network device.

public IAsyncResult BeginConnect(string, int, AsyncCallback, object);
public IAsyncResult BeginConnect(EndPoint, AsyncCallback, object);

See Also


Copyright © 2002, The KPD-Team
Mentalis.org ProxySocket Documentation
ProxySocket.BeginConnect Method (EndPoint, AsyncCallback, Object)

Begins an asynchronous request for a connection to a network device.

[Visual Basic] Overloads Public Function BeginConnect(
  ByVal remoteEP As System.Net.EndPoint,
  ByVal callback As System.AsyncCallback,
  ByVal state As Object
) As System.IAsyncResult

[C#]
public IAsyncResult BeginConnect(
  EndPoint remoteEP,
  AsyncCallback callback,
  object state
);

Parameters

remoteEP
An EndPoint that represents the remote device.

callback
The AsyncCallback delegate.

state
An object that contains state information for this request.

Return Value
An IAsyncResult that references the asynchronous connection.

Exceptions

<table>
<thead>
<tr>
<th>Exception Type</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exception</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td><strong>System.Net.Sockets.SocketException</strong></td>
<td>An operating system error occurs while creating the Socket.</td>
</tr>
<tr>
<td><strong>System.ObjectDisposedException</strong></td>
<td>The Socket has been closed.</td>
</tr>
</tbody>
</table>

See Also


Copyright © 2002, The KPD-Team

*Mentalis.org ProxySocket Documentation*
ProxySocket.BeginConnect Method (EndPoint, AsyncCallback, Object)

Begins an asynchronous request for a connection to a network device.

[Visual Basic]
Overloads Public Function BeginConnect( _
    ByVal remoteEP As System.Net.EndPoint, _
    ByVal callback As System.AsyncCallback, _
    ByVal state As Object _
) As System.IAsyncResult

[C#]
public IAsyncResult BeginConnect(
    EndPoint remoteEP,
    AsyncCallback callback,
    object state
);

Parameters

remoteEP
An EndPoint that represents the remote device.

callback
The AsyncCallback delegate.

state
An object that contains state information for this request.

Return Value
An IAsyncResult that references the asynchronous connection.

Exceptions

<table>
<thead>
<tr>
<th>Exception Type</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>System.ArgumentNullException</td>
<td>The remoteEP parameter is a null reference (Nothing in</td>
</tr>
<tr>
<td>Exception</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------------------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td><code>System.Net.Sockets.SocketException</code></td>
<td>An operating system error occurs while creating the Socket.</td>
</tr>
<tr>
<td><code>System.ObjectDisposedException</code></td>
<td>The Socket has been closed.</td>
</tr>
</tbody>
</table>

**See Also**

- [ProxySocket Class](#)
- [ProxySocket Members](#)
- [ProxySocket.BeginConnect Overload List](#)

*Copyright © 2002, The KPD-Team*
Mentalis.org ProxySocket Documentation
Begins an asynchronous request for a connection to a network device.

[Visual Basic]
Overloads Public Function BeginConnect(
  ByVal host As String, _
  ByVal port As Integer, _
  ByVal callback As System.AsyncCallback, _
  ByVal state As Object _
) As System.IAsyncResult

[C#]
public IAsyncResult BeginConnect(
    string host,
    int port,
    AsyncCallback callback,
    object state
);

Parameters

host
  The host to connect to.

port
  The port on the remote host to connect to.

callback
  The AsyncCallback delegate.

state
  An object that contains state information for this request.

Return Value

An IAsyncResult that references the asynchronous connection.

Exceptions
<table>
<thead>
<tr>
<th>Exception Type</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>System.ArgumentNullException</td>
<td>The host parameter is a null reference (Nothing in Visual Basic).</td>
</tr>
<tr>
<td>System.ArgumentException</td>
<td>The port parameter is invalid.</td>
</tr>
<tr>
<td>System.Net.Sockets.SocketException</td>
<td>An operating system error occurs while creating the Socket.</td>
</tr>
<tr>
<td>System.ObjectDisposedException</td>
<td>The Socket has been closed.</td>
</tr>
</tbody>
</table>

See Also

- ProxySocket Class
- ProxySocket Members
- ProxySocket.BeginConnect Overload List

Copyright © 2002, The KPD-Team
ProxySocket.BeginDns Method

Begins an asynchronous request to resolve a DNS host name or IP address in dotted-quad notation to an IPAddress instance.

[Visual Basic]Friend Function BeginDns( _
  ByVal host As String, _

[C#]
internal IAsyncProxyResult BeginDns(
  string host,
  HandShakeComplete callback
);

Parameters

  host
  The host to resolve.

  callback
  The method to call when the hostname has been resolved.

Return Value

  An IAsyncResult instance that references the asynchronous request.

Exceptions

<table>
<thead>
<tr>
<th>Exception Type</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>System.Net.Sockets.SocketException</td>
<td>There was an error while trying to resolve the host.</td>
</tr>
</tbody>
</table>

See Also


Copyright © 2002, The KPD-Team
Mentalis.org ProxySocket Documentation
**ProxySocket.Connect**

Establishes a connection to a remote device.

**Overload List**

Establishes a connection to a remote device.

```java
public void Connect(EndPoint);
```

Establishes a connection to a remote device.

```java
public void Connect(string,int);
```

```java
public void Connect(EndPoint);
```

**See Also**


Copyright © 2002, The KPD-Team
ProxySocket.Connect Method (EndPoint)

Establishes a connection to a remote device.

**[Visual Basic]**

```vbnet
Overloads Public Sub Connect(
    ByVal remoteEP As System.Net.EndPoint
)
```

**[C#]**

```csharp
public void Connect(
    EndPoint remoteEP
);
```

**Parameters**

`remoteEP`

An `EndPoint` that represents the remote device.

**Exceptions**

<table>
<thead>
<tr>
<th>Exception Type</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>System.Net.Sockets.SocketException</code></td>
<td>An operating system error occurs while accessing the Socket.</td>
</tr>
<tr>
<td><code>System.ObjectDisposedException</code></td>
<td>The Socket has been closed.</td>
</tr>
<tr>
<td><code>Org.Mentalis.Network.ProxySocket.ProxyException</code></td>
<td>An error occurred while talking to the</td>
</tr>
</tbody>
</table>
See Also


Copyright © 2002, The KPD-Team

Mentalis.org ProxySocket Documentation
Establishes a connection to a remote device.

**[Visual Basic]**

```vbnet
Overloads Public Sub Connect( _
    ByVal remoteEP As System.Net.EndPoint _
)```

**[C#]**

```csharp
public void Connect(
    EndPoint remoteEP
);```

### Parameters

*remoteEP*

An EndPoint that represents the remote device.

### Exceptions

<table>
<thead>
<tr>
<th>Exception Type</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>System.Net.Sockets.SocketException</code></td>
<td>An operating system error occurs while accessing the Socket.</td>
</tr>
<tr>
<td><code>System.ObjectDisposedException</code></td>
<td>The Socket has been closed.</td>
</tr>
</tbody>
</table>
See Also


Copyright © 2002, The KPD-Team
Establishes a connection to a remote device.

```plaintext
[Visual Basic] Overloads Public Sub Connect(
  ByVal host As String, 
  ByVal port As Integer 
)

[C#]
public void Connect(
  string host, 
  int port 
);
```

Parameters

- `host`
  The remote host to connect to.

- `port`
  The remote port to connect to.

Remarks

If you use this method with a SOCKS4 server, it will let the server resolve the hostname. Not all SOCKS4 servers support this 'remote DNS' though.

Exceptions

<table>
<thead>
<tr>
<th>Exception Type</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>System.ArgumentNullException</code></td>
<td>The host parameter is a null reference (Nothing in Visual Basic).</td>
</tr>
<tr>
<td><code>System.ArgumentException</code></td>
<td>The port parameter is</td>
</tr>
<tr>
<td>Exception Class</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>-------------</td>
</tr>
<tr>
<td><strong>System.Net.Sockets.SocketException</strong></td>
<td>An operating system error occurs while accessing the Socket.</td>
</tr>
<tr>
<td><strong>System.ObjectDisposedException</strong></td>
<td>The Socket has been closed.</td>
</tr>
</tbody>
</table>

**See Also**


Copyright © 2002, The KPD-Team
ProxySocket.EndConnect Method (IAsyncResult)

Ends a pending asynchronous connection request.

[Visual Basic] Overloads Public Sub EndConnect(ByVal asyncResult As System.IAsyncResult)

[C#]
public void EndConnect(
    IAsyncResult asyncResult)

Parameters

asyncResult
   Stores state information for this asynchronous operation as well as any user-defined data.

Exceptions

<table>
<thead>
<tr>
<th>Exception Type</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>System.ArgumentNullException</td>
<td>The asyncResult parameter is a null reference (Nothing in Visual Basic).</td>
</tr>
<tr>
<td>System.ArgumentException</td>
<td>The asyncResult parameter was not returned by a call to the BeginConnect method.</td>
</tr>
<tr>
<td>System.Net.Sockets.SocketException</td>
<td>An operating system error</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Exception</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>System.ObjectDisposedException</strong></td>
<td>The Socket has been closed.</td>
</tr>
<tr>
<td><strong>System.InvalidOperationException</strong></td>
<td>EndConnect was previously called for the asynchronous connection.</td>
</tr>
</tbody>
</table>

See Also

- [ProxySocket Class](#) | [ProxySocket Members](#) |

Copyright © 2002, The KPD-Team

Mentalis.org ProxySocket Documentation
ProxySocket.EndConnect Method (IAasyncResult)

Ends a pending asynchronous connection request.

```visual-basic
Overloads Public Sub EndConnect(
    ByVal asyncResult As System.IAsyncResult
)
```

```csharp
public void EndConnect(
    IAsyncResult asyncResult
);
```

Parameters

asyncResult
 Stores state information for this asynchronous operation as well as any user-defined data.

Exceptions

<table>
<thead>
<tr>
<th>Exception Type</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>System.ArgumentNullException</strong></td>
<td>The asyncResult parameter is a null reference (Nothing in Visual Basic).</td>
</tr>
<tr>
<td><strong>System.ArgumentException</strong></td>
<td>The asyncResult parameter was not returned by a call to the BeginConnect method.</td>
</tr>
<tr>
<td><strong>System.Net.Sockets.SocketException</strong></td>
<td>An operating</td>
</tr>
<tr>
<td>Exception Type</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>-------------</td>
</tr>
<tr>
<td>System.ObjectDisposedException</td>
<td>The Socket has been closed.</td>
</tr>
<tr>
<td>System.InvalidOperationException</td>
<td>EndConnect was previously called for the asynchronous connection.</td>
</tr>
</tbody>
</table>

**See Also**


Copyright © 2002, The KPD-Team
ProxySocket.OnConnect Method

Called when the Socket is connected to the remote host.

[Visual Basic]Private Sub OnConnect( _
    ByVal asyncResult As System.IAsyncResult)_
)

[C#]
private void OnConnect(
    IAsyncResult asyncResult
);

Parameters

asyncResult
The result of the asynchronous operation.

See Also

Copyright © 2002, The KPD-Team
ProxySocket.OnHandShakeComplete Method

Called when the Socket has finished talking to the proxy server and is ready to relay data.

[Visual Basic] Private Sub OnHandShakeComplete(ByVal error As System.Exception )

[C#] private void OnHandShakeComplete( Exception error );

Parameters

error
The error to throw when the EndConnect method is called.

See Also


Copyright © 2002, The KPD-Team
Mentalis.org ProxySocket Documentation
ProxySocket.OnResolved Method

Called when the specified hostname has been resolved.

[Visual Basic]
Private Sub OnResolved( ByVal asyncResult As System.IAsyncResult )

[C#]
private void OnResolved( IAsyncResult asyncResult );

Parameters

asyncResult
The result of the asynchronous operation.

See Also


Copyright © 2002, The KPD-Team
ProxyTypes Enumeration

Specifies the type of proxy servers that an instance of the ProxySocket class can use.

[Visual Basic] Public Enum ProxyTypes

[C#] public enum ProxyTypes

Members

<table>
<thead>
<tr>
<th>Member Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>No proxy server; the ProxySocket object behaves exactly like an ordinary Socket object.</td>
</tr>
<tr>
<td>Socks4</td>
<td>A SOCKS4[A] proxy server.</td>
</tr>
<tr>
<td>Socks5</td>
<td>A SOCKS5 proxy server.</td>
</tr>
</tbody>
</table>

Requirements


See Also


Copyright © 2002, The KPD-Team
Socks4Handler Class

Implements the SOCKS4[A] protocol.
For a list of all members of this type, see Socks4Handler Members.

System.Object  SocksHandler
Socks4Handler

[Visual Basic]
NotInheritable  Class Socks4Handler
Inherits SocksHandler

[C#]
sealed class Socks4Handler : SocksHandler

Requirements


See Also


Copyright © 2002, The KPD-Team
# Socks4Handler Members

## Public Instance Constructors

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Socks4Handler Constructor</strong></td>
<td>Initializes a new instance of the SocksHandler class.</td>
</tr>
</tbody>
</table>

## Public Instance Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BeginNegotiate</strong></td>
<td>Overloaded. Starts negotiating asynchronously with a SOCKS proxy server.</td>
</tr>
<tr>
<td><strong>Equals</strong> (inherited from <strong>Object</strong>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>GetHashCode</strong> (inherited from <strong>Object</strong>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>GetType</strong> (inherited from <strong>Object</strong>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>Negotiate</strong></td>
<td>Overloaded. Starts negotiating with the SOCKS server.</td>
</tr>
<tr>
<td><strong>ToString</strong> (inherited from <strong>Object</strong>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
</tbody>
</table>

## Protected Instance Fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ProtocolComplete</strong></td>
<td></td>
</tr>
</tbody>
</table>

## Protected Instance Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AsyncResult</strong> (inherited from <strong>SocksHandler</strong>)</td>
<td>Gets or sets the return value of the BeginConnect call.</td>
</tr>
<tr>
<td><strong>Buffer</strong> (inherited from <strong>SocksHandler</strong>)</td>
<td>Gets or sets a byte buffer.</td>
</tr>
<tr>
<td><strong>Received</strong> (inherited from <strong>SocksHandler</strong>)</td>
<td>Gets or sets the number of bytes that have been received from the remote proxy server.</td>
</tr>
</tbody>
</table>
### Server (inherited from SocksHandler)
Gets or sets the socket connection with the proxy server.

### Username (inherited from SocksHandler)
Gets or sets the username to use when authenticating with the proxy server.

### Protected Instance Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AddressToBytes</strong> (inherited from SocksHandler)</td>
<td>Converts an IP address to an array of bytes.</td>
</tr>
<tr>
<td><strong>Finalize</strong> (inherited from Object)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>MemberwiseClone</strong> (inherited from Object)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>PortToBytes</strong> (inherited from SocksHandler)</td>
<td>Converts a port number to an array of bytes.</td>
</tr>
<tr>
<td><strong>ReadBytes</strong> (inherited from SocksHandler)</td>
<td>Reads a specified number of bytes from the Server socket.</td>
</tr>
</tbody>
</table>

### Private Instance Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>GetEndPointBytes</strong></td>
<td>Creates an array of bytes that has to be sent when the user wants to connect to a specific IPEndPoint.</td>
</tr>
<tr>
<td><strong>GetHostPortBytes</strong></td>
<td>Creates an array of bytes that has to be sent when the user wants to connect to a specific host/port combination.</td>
</tr>
<tr>
<td><strong>OnConnect</strong></td>
<td>Called when the Socket is connected to the remote proxy server.</td>
</tr>
<tr>
<td><strong>OnReceive</strong></td>
<td>Called when the Socket has received a reply from the remote proxy server.</td>
</tr>
</tbody>
</table>
OnSent

Called when the Socket has sent the handshake data.

See Also


Copyright © 2002, The KPD-Team
Socks4Handler Constructor

Initializes a new instance of the SocksHandler class.

**[Visual Basic]**

```vbnet
Public Sub New( _
    ByVal server As System.Net.Sockets.Socket, _
    ByVal user As String _
)
```

**[C#]**

```csharp
public Socks4Handler( _
    Socket server, _
    string user
);
```

**Parameters**

- `server`  
  The socket connection with the proxy server.

- `user`  
  The username to use when authenticating with the server.

**Exceptions**

<table>
<thead>
<tr>
<th>Exception Type</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>System.ArgumentNullException</td>
<td>server -or- user is null.</td>
</tr>
</tbody>
</table>

**See Also**

- [Socks4Handler Class](#)  
  - [Socks4Handler Members](#)  

Copyright © 2002, The KPD-Team
### Socks4Handler Fields

The fields of the `Socks4Handler` class are listed below. For a complete list of `Socks4Handler` class members, see the `Socks4Handler Members` topic.

#### Protected Instance Fields

<table>
<thead>
<tr>
<th>ProtocolComplete</th>
</tr>
</thead>
</table>

#### See Also


*[Copyright © 2002, The KPD-Team]*
Socks4Handler.ProtocolComplete Field

[Visual Basic] Protected ProtocolComplete As (}

[C#]
protected HandShakeComplete ProtocolComplete;

See Also


Copyright © 2002, The KPD-Team
Socks4Handler Methods

The methods of the Socks4Handler class are listed below. For a complete list of Socks4Handler class members, see the Socks4Handler Members topic.

Public Instance Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BeginNegotiate</strong></td>
<td>Overloaded. Starts negotiating asynchronously with a SOCKS proxy server.</td>
</tr>
<tr>
<td><strong>Equals</strong> (inherited from Object)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>GetHashCode</strong> (inherited from Object)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>GetType</strong> (inherited from Object)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>Negotiate</strong></td>
<td>Overloaded. Starts negotiating with the SOCKS server.</td>
</tr>
<tr>
<td><strong>ToString</strong> (inherited from Object)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
</tbody>
</table>

Protected Instance Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AddressToBytes</strong> (inherited from SocksHandler)</td>
<td>Converts an IP address to an array of bytes.</td>
</tr>
<tr>
<td><strong>Finalize</strong> (inherited from Object)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>MemberwiseClone</strong> (inherited from Object)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>PortToBytes</strong> (inherited from SocksHandler)</td>
<td>Converts a port number to an array of bytes.</td>
</tr>
<tr>
<td><strong>ReadBytes</strong> (inherited from SocksHandler)</td>
<td>Reads a specified number of bytes from the Server socket.</td>
</tr>
</tbody>
</table>
## Private Instance Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetEndPointBytes</td>
<td>Creates an array of bytes that has to be sent when the user wants to connect to a specific IPEndPoint.</td>
</tr>
<tr>
<td>GetHostPortBytes</td>
<td>Creates an array of bytes that has to be sent when the user wants to connect to a specific host/port combination.</td>
</tr>
<tr>
<td>OnConnect</td>
<td>Called when the Socket is connected to the remote proxy server.</td>
</tr>
<tr>
<td>OnReceive</td>
<td>Called when the Socket has received a reply from the remote proxy server.</td>
</tr>
<tr>
<td>OnSent</td>
<td>Called when the Socket has sent the handshake data.</td>
</tr>
</tbody>
</table>

### See Also


Copyright © 2002, The KPD-Team
Socks4Handler.BeginNegotiate

Starts negotiating asynchronously with a SOCKS proxy server.

Overload List

Starts negotiating asynchronously with a SOCKS proxy server.

```csharp
public override IAsyncProxyResult BeginNegotiate(string, int, HandShakeComplete, IPEndPoint);
```

Starts negotiating asynchronously with a SOCKS proxy server.

```csharp
public override IAsyncProxyResult BeginNegotiate(IPEndPoint, HandShakeComplete, IPEndPoint);
```

See Also

[Socks4Handler Class] | [Socks4Handler Members] | [Org.Mentalis.Network.ProxySocket Namespace]

Copyright © 2002, The KPD-Team
Socks4Handler.BeginNegotiate Method (IPEndPoint, HandShakeComplete, IPEndPoint)

Starts negotiating asynchronously with a SOCKS proxy server.


[C#] public override IAsyncProxyResult BeginNegotiate(   IPEndPoint remoteEP,   HandShakeComplete callback,   IPEndPoint proxyEndPoint
);

Parameters

remoteEP
An IPEndPoint that represents the remote device.

callback
The method to call when the connection has been established.

proxyEndPoint
The IPEndPoint of the SOCKS proxy server.

Return Value
An IAsyncProxyResult that references the asynchronous connection.

See Also

Copyright © 2002, The KPD-Team
**Socks4Handler.BeginNegotiate Method (String, Int32, HandShakeComplete, IPEndPoint)**

Starts negotiating asynchronously with a SOCKS proxy server.

[Visual Basic] `Overrides Overloads Public Function BeginNegotiate(ByVal host As String, _
ByVal port As Integer, _

[C#] `public override IAsyncProxyResult BeginNegotiate(string host, int port, HandShakeComplete callback, IPEndPoint proxyEndPoint);`

**Parameters**

- **host**
  The remote server to connect to.

- **port**
  The remote port to connect to.

- **callback**
  The method to call when the connection has been established.

- **proxyEndPoint**
  The IPEndPoint of the SOCKS proxy server.

**Return Value**

An IAsyncProxyResult that references the asynchronous connection.

**See Also**

[Socks4Handler Class] | [Socks4Handler Members]

Copyright © 2002, The KPD-Team
Socks4Handler.GetEndPointBytes Method

Creates an array of bytes that has to be sent when the user wants to connect to a specific IPEndPoint.

[Visual Basic] Private Function GetEndPointBytes(ByVal remoteEP As System.Net.IPEndPoint) As Byte()

[C#] private byte[] GetEndPointBytes(IPEndPoint remoteEP);

Parameters

remoteEP
The IPEndPoint to connect to.

Return Value

An array of bytes that has to be sent when the user wants to connect to a specific IPEndPoint.

Exceptions

<table>
<thead>
<tr>
<th>Exception Type</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>System.ArgumentNullException</td>
<td>remoteEP is null.</td>
</tr>
</tbody>
</table>

See Also


Copyright © 2002, The KPD-Team
Socks4Handler.GetHostPortBytes Method

Creates an array of bytes that has to be sent when the user wants to connect to a specific host/port combination.

[Visual Basic]
Private Function GetHostPortBytes(
    ByVal host As String,
    ByVal port As Integer
) As Byte()

[C#]
private byte[] GetHostPortBytes(
    string host,
    int port
);

Parameters

(host)
   The host to connect to.

(port)
   The port to connect to.

Return Value

An array of bytes that has to be sent when the user wants to connect to a specific host/port combination.

Remarks

Resolving the host name will be done at server side. Do note that some SOCKS4 servers do not implement this functionality.

Exceptions

<table>
<thead>
<tr>
<th>Exception Type</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>System.ArgumentNullException</td>
<td>host is null.</td>
</tr>
<tr>
<td>System.ArgumentException</td>
<td>port is invalid.</td>
</tr>
</tbody>
</table>
See Also


Copyright © 2002, The KPD-Team
Socks4Handler.Negotiate

Starts negotiating with the SOCKS server.

**Overload List**

Starts negotiating with the SOCKS server.

```csharp
public override void Negotiate(IPEndPoint);
```

Starts negotiating with the SOCKS server.

```csharp
public override void Negotiate(string,int);
```

Starts negotiating with the SOCKS server.

```csharp
private void Negotiate(byte[]);
```

**See Also**


Copyright © 2002, The KPD-Team
Socks4Handler.Negotiate Method (Byte[])  

Starts negotiating with the SOCKS server.

[Visual Basic] Overloads Private Sub Negotiate(
    ByVal connect As Byte() _
)

[C#]  
private void Negotiate(
    byte[] connect
);

Parameters

connect
The bytes to send when trying to authenticate.

Exceptions

<table>
<thead>
<tr>
<th>Exception Type</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>System.ArgumentNullException</td>
<td>connect is null.</td>
</tr>
<tr>
<td>System.ArgumentException</td>
<td>connect is too small.</td>
</tr>
<tr>
<td>System.Net.Sockets.SocketException</td>
<td>An operating system error occurs while accessing the Socket.</td>
</tr>
<tr>
<td>System.ObjectDisposedException</td>
<td>The Socket has been closed.</td>
</tr>
</tbody>
</table>
See Also


Copyright © 2002, The KPD-Team
Socks4Handler.Negotiate Method (IPEndPoint)

Starts negotiating with the SOCKS server.

[Visual Basic]Overrides Overloads Public Sub
ByVal remoteEP As System.Net.IPEndPoint

[C#]
public override void Negotiate(IPEndPoint remoteEP);

Parameters

remoteEP
The IPEndPoint to connect to.

Exceptions

<table>
<thead>
<tr>
<th>Exception Type</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>System.ArgumentNullException</td>
<td>remoteEP is null.</td>
</tr>
<tr>
<td>System.Net.Sockets.SocketException</td>
<td>An operating system error occurs while accessing the Socket.</td>
</tr>
<tr>
<td>System.ObjectDisposedException</td>
<td>The Socket has been closed.</td>
</tr>
</tbody>
</table>

See Also
Mentalis.org ProxySocket Documentation
Socks4Handler.Negotiate Method (String, Int32)

Starts negotiating with the SOCKS server.

[Visual Basic] Overrides Overloads Public Sub
ByVal host As String,
ByVal port As Integer
)

[C#]
public override void Negotiate(
    string host,
    int port
);

Parameters

host
The host to connect to.

port
The port to connect to.

Exceptions

<table>
<thead>
<tr>
<th>Exception Type</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>SystemArgumentNullException</td>
<td>host is null.</td>
</tr>
<tr>
<td>System.ArgumentException</td>
<td>port is invalid.</td>
</tr>
<tr>
<td>System.Net.Sockets.SocketException</td>
<td>An operating system error occurs while accessing the Socket.</td>
</tr>
</tbody>
</table>
System.ObjectDisposedException

The Socket has been closed.

See Also


Copyright © 2002, The KPD-Team
Socks4Handler.OnConnect Method

Called when the Socket is connected to the remote proxy server.

```visualbasic
Private Sub OnConnect(ByVal ar As System.IAsyncResult)
```

```csharp
private void OnConnect(IAsyncResult ar);
```

Parameters

`ar`  
Stores state information for this asynchronous operation as well as any user-defined data.

See Also

- Socks4Handler Class  
- Socks4Handler Members  

Copyright © 2002, The KPD-Team
Socks4Handler.OnReceive Method

Called when the Socket has received a reply from the remote proxy server.

[Visual Basic] Private Sub OnReceive( _
    ByVal ar As System.IAsyncResult _
)"

[C#]
private void OnReceive( 
    IAsyncResult ar
);"

Parameters

ar
Stores state information for this asynchronous operation as well as any user-defined data.

See Also


Copyright © 2002, The KPD-Team
Socks4Handler.OnSent Method

Called when the Socket has sent the handshake data.

[Visual Basic] Private Sub OnSent(_
    ByVal ar As System.IAsyncResult _
) _

[C#] private void OnSent(_
    IAsyncResult ar
); _

Parameters

ar
Stores state information for this asynchronous operation as well as any user-defined data.

See Also


Copyright © 2002, The KPD-Team
Socks5Handler Class

Implements the SOCKS5 protocol.

For a list of all members of this type, see Socks5Handler Members.

System.Object   SocksHandler
Socks5Handler

[Visual Basic]
NotInheritable   Class Socks5Handler
                 Inherits SocksHandler

[C#]
sealed class Socks5Handler : SocksHandler

Requirements


See Also


Copyright © 2002, The KPD-Team
## Socks5Handler Members

### Public Instance Constructors

| Socks5Handler | Overloaded. Initialize a new instance of the Socks5Handler class. |

### Public Instance Methods

| BeginNegotiate | Overloaded. Starts negotiating asynchronously with the SOCKS server. |
| Equals (inherited from Object) | Select the method name to go to the Microsoft documentation. |
| GetHashCode (inherited from Object) | Select the method name to go to the Microsoft documentation. |
| GetType (inherited from Object) | Select the method name to go to the Microsoft documentation. |
| Negotiate | Overloaded. Starts negotiating with the SOCKS server. |
| ToString (inherited from Object) | Select the method name to go to the Microsoft documentation. |

### Protected Instance Fields

| ProtocolComplete |

### Protected Instance Properties

| AsyncResult (inherited from SocksHandler) | Gets or sets the return value of the BeginConnect call. |
| Buffer (inherited from SocksHandler) | Gets or sets a byte buffer. |
| Received (inherited from SocksHandler) | Gets or sets the number of bytes that have been received from |
the remote proxy server.

<table>
<thead>
<tr>
<th>Server (inherited from SocksHandler)</th>
<th>Gets or sets the socket connection with the proxy server.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Username (inherited from SocksHandler)</td>
<td>Gets or sets the username to use when authenticating with the proxy server.</td>
</tr>
</tbody>
</table>

Protected Instance Methods

<table>
<thead>
<tr>
<th>AddressToBytes (inherited from SocksHandler)</th>
<th>Converts an IP address to an array of bytes.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Finalize (inherited from Object)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td>MemberwiseClone (inherited from Object)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td>PortToBytes (inherited from SocksHandler)</td>
<td>Converts a port number to an array of bytes.</td>
</tr>
<tr>
<td>ReadBytes (inherited from SocksHandler)</td>
<td>Reads a specified number of bytes from the Server socket.</td>
</tr>
</tbody>
</table>

Private Instance Fields

<table>
<thead>
<tr>
<th>m_HandShake</th>
<th>Holds the value of the HandShake property.</th>
</tr>
</thead>
<tbody>
<tr>
<td>m_Password</td>
<td>Holds the value of the Password property.</td>
</tr>
</tbody>
</table>

Private Instance Properties

<table>
<thead>
<tr>
<th>HandShake</th>
<th>Gets or sets the bytes to use when sending a connect request to the proxy server.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Password</td>
<td>Gets or sets the password to use when authenticating with the SOCKS5 server.</td>
</tr>
</tbody>
</table>
Private Instance Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authenticate</td>
<td>Starts the synchronous authentication process.</td>
</tr>
<tr>
<td>GetEndPointBytes</td>
<td>Creates an array of bytes that has to be sent when the user wants to connect to a specific IPEndPoint.</td>
</tr>
<tr>
<td>GetHostPortBytes</td>
<td>Creates an array of bytes that has to be sent when the user wants to connect to a specific host/port combination.</td>
</tr>
<tr>
<td>OnAuthenticated</td>
<td>Called when the socket has been successfully authenticated with the server.</td>
</tr>
<tr>
<td>OnAuthReceive</td>
<td>Called when an authentication reply has been received.</td>
</tr>
<tr>
<td>OnAuthSent</td>
<td>Called when the authentication bytes have been sent.</td>
</tr>
<tr>
<td>OnConnect</td>
<td>Called when the socket is connected to the remote server.</td>
</tr>
<tr>
<td>OnReadLast</td>
<td>Called when the last bytes are read from the socket.</td>
</tr>
<tr>
<td>OnReceive</td>
<td>Called when a connection reply has been received.</td>
</tr>
<tr>
<td>OnSent</td>
<td>Called when the connection request has been sent.</td>
</tr>
<tr>
<td>ProcessReply</td>
<td>Processes the received reply.</td>
</tr>
</tbody>
</table>

See Also


Copyright © 2002, The KPD-Team
**Socks5Handler**

Initializes a new Socks5Handler instance.

**Overload List**

Initializes a new Socks5Handler instance.

```csharp
public Socks5Handler(Socket);
```

Initializes a new Socks5Handler instance.

```csharp
public Socks5Handler(Socket,string);
```

Initializes a new Socks5Handler instance.

```csharp
public Socks5Handler(Socket,string,string);
```

**See Also**

[Socks5Handler Class] | [Socks5Handler Members] | [Org.Mentalis.Network.ProxySocket Namespace]

Copyright © 2002, The KPD-Team
Socks5Handler Constructor (Socket)

Initializes a new Socks5Handler instance.

[Visual Basic]
Overloads Public Sub New(_
    ByVal server As System.Net.Sockets.Socket
)

[C#]
public Socks5Handler(
    Socket server
);

Parameters

server
The socket connection with the proxy server.

Exceptions

<table>
<thead>
<tr>
<th>Exception Type</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>System.ArgumentNullException</td>
<td>server is null.</td>
</tr>
</tbody>
</table>

See Also


Copyright © 2002, The KPD-Team
Socks5Handler Constructor (Socket, String)

Initilizes a new Socks5Handler instance.

```visualbasic
Overloads Public Sub New( 
    ByVal server As System.Net.Sockets.Socket, 
    ByVal user As String 
)
```

```c#
public Socks5Handler( 
    Socket server, 
    string user 
);
```

Parameters

`server`

The socket connection with the proxy server.

`user`

The username to use.

Exceptions

<table>
<thead>
<tr>
<th>Exception Type</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>System.ArgumentNullException</td>
<td>server -or- user is null.</td>
</tr>
</tbody>
</table>

See Also


Copyright © 2002, The KPD-Team
Socks5Handler Constructor (Socket, String, String)

Initializes a new Socks5Handler instance.

[Visual Basic] Overloads Public Sub New( _
    ByVal server As System.Net.Sockets.Socket,
    ByVal user As String, _
    ByVal pass As String _
)

[C#]
public Socks5Handler(
    Socket server,  
    string user,  
    string pass
);

Parameters

server
  The socket connection with the proxy server.

user
  The username to use.

pass
  The password to use.

Exceptions

<table>
<thead>
<tr>
<th>Exception Type</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>System.ArgumentNullException</td>
<td>server -or- user -or- pass is null.</td>
</tr>
</tbody>
</table>

See Also

Copyright © 2002, The KPD-Team
Socks5Handler Fields

The fields of the Socks5Handler class are listed below. For a complete list of Socks5Handler class members, see the Socks5Handler Members topic.

Protected Instance Fields

<table>
<thead>
<tr>
<th>ProtocolComplete</th>
</tr>
</thead>
</table>

Private Instance Fields

<table>
<thead>
<tr>
<th>m_HandShake</th>
<th>Holds the value of the HandShake property.</th>
</tr>
</thead>
<tbody>
<tr>
<td>m_Password</td>
<td>Holds the value of the Password property.</td>
</tr>
</tbody>
</table>

See Also


Copyright © 2002, The KPD-Team
## Socks5Handler.m_HandShake Field

Holds the value of the HandShake property.

### Visual Basic

```vbnet
Private m_HandShake As Byte()
```

### C#

```csharp
private byte[] m_HandShake;
```

See Also

- Socks5Handler Class
- Socks5Handler Members

Copyright © 2002, The KPD-Team
**Socks5Handler.m_Password Field**

Holds the value of the Password property.

```vbnet
[Visual Basic]
Private m_Password As String
```

```csharp
[C#]
private string m_Password;
```

**See Also**

[Socks5Handler Class]  |  [Socks5Handler Members]  |  [Org.Mentalis.Network.ProxySocket Namespace]

**Copyright © 2002, The KPD-Team**
Socks5Handler.ProtocolComplete Field

[Visual Basic] Protected ProtocolComplete As (  

[C#]  

protected HandShakeComplete ProtocolComplete;

See Also


Copyright © 2002, The KPD-Team
The properties of the **Socks5Handler** class are listed below. For a complete list of **Socks5Handler** class members, see the **Socks5Handler Members** topic.

### Protected Instance Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AsyncResult</strong> (inherited from <strong>SocksHandler</strong>)</td>
<td>Gets or sets the return value of the BeginConnect call.</td>
</tr>
<tr>
<td><strong>Buffer</strong> (inherited from <strong>SocksHandler</strong>)</td>
<td>Gets or sets a byte buffer.</td>
</tr>
<tr>
<td><strong>Received</strong> (inherited from <strong>SocksHandler</strong>)</td>
<td>Gets or sets the number of bytes that have been received from the remote proxy server.</td>
</tr>
<tr>
<td><strong>Server</strong> (inherited from <strong>SocksHandler</strong>)</td>
<td>Gets or sets the socket connection with the proxy server.</td>
</tr>
<tr>
<td><strong>Username</strong> (inherited from <strong>SocksHandler</strong>)</td>
<td>Gets or sets the username to use when authenticating with the proxy server.</td>
</tr>
</tbody>
</table>

### Private Instance Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>HandShake</strong></td>
<td>Gets or sets the bytes to use when sending a connect request to the proxy server.</td>
</tr>
<tr>
<td><strong>Password</strong></td>
<td>Gets or sets the password to use when authenticating with the SOCKS5 server.</td>
</tr>
</tbody>
</table>

See Also


*Copyright © 2002, The KPD-Team*
Socks5Handler.HandShake Property

Gets or sets the bytes to use when sending a connect request to the proxy server.

[Visual Basic] Private Property HandShake As Byte()

[C#] private byte[] HandShake {get; set;}

Property Value

The array of bytes to use when sending a connect request to the proxy server.

See Also


Copyright © 2002, The KPD-Team
Socks5Handler.Password Property

Gets or sets the password to use when authenticating with the SOCKS5 server.

[Visual Basic] Private Property Password As String

[C#]
private string Password {get; set;}

Property Value

The password to use when authenticating with the SOCKS5 server.

See Also


Copyright © 2002, The KPD-Team
# Socks5Handler Methods

The methods of the **Socks5Handler** class are listed below. For a complete list of **Socks5Handler** class members, see the [Socks5Handler Members](#) topic.

## Public Instance Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BeginNegotiate</strong></td>
<td>Overloaded. Starts negotiating asynchronously with the SOCKS server.</td>
</tr>
<tr>
<td><strong>Equals</strong> (inherited from <strong>Object</strong>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>GetHashCode</strong> (inherited from <strong>Object</strong>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>GetType</strong> (inherited from <strong>Object</strong>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>Negotiate</strong></td>
<td>Overloaded. Starts negotiating with the SOCKS server.</td>
</tr>
<tr>
<td><strong>ToString</strong> (inherited from <strong>Object</strong>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
</tbody>
</table>

## Protected Instance Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AddressToBytes</strong> (inherited from <strong>SocksHandler</strong>)</td>
<td>Converts an IP address to an array of bytes.</td>
</tr>
<tr>
<td><strong>Finalize</strong> (inherited from <strong>Object</strong>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>MemberwiseClone</strong> (inherited from <strong>Object</strong>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>PortToBytes</strong> (inherited from <strong>SocksHandler</strong>)</td>
<td>Converts a port number to an array of bytes.</td>
</tr>
<tr>
<td><strong>ReadBytes</strong> (inherited from <strong>SocksHandler</strong>)</td>
<td>Reads a specified number of bytes from the Server socket.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Authenticate</strong></td>
<td>Starts the synchronous authentication process.</td>
</tr>
<tr>
<td><strong>GetEndPointBytes</strong></td>
<td>Creates an array of bytes that has to be sent when the user wants to connect to a specific IPEndPoint.</td>
</tr>
<tr>
<td><strong>GetHostPortBytes</strong></td>
<td>Creates an array of bytes that has to be sent when the user wants to connect to a specific host/port combination.</td>
</tr>
<tr>
<td><strong>OnAuthenticated</strong></td>
<td>Called when the socket has been successfully authenticated with the server.</td>
</tr>
<tr>
<td><strong>OnAuthReceive</strong></td>
<td>Called when an authentication reply has been received.</td>
</tr>
<tr>
<td><strong>OnAuthSent</strong></td>
<td>Called when the authentication bytes have been sent.</td>
</tr>
<tr>
<td><strong>OnConnect</strong></td>
<td>Called when the socket is connected to the remote server.</td>
</tr>
<tr>
<td><strong>OnReadLast</strong></td>
<td>Called when the last bytes are read from the socket.</td>
</tr>
<tr>
<td><strong>OnReceive</strong></td>
<td>Called when a connection reply has been received.</td>
</tr>
<tr>
<td><strong>OnSent</strong></td>
<td>Called when the connection request has been sent.</td>
</tr>
<tr>
<td><strong>ProcessReply</strong></td>
<td>Processes the received reply.</td>
</tr>
</tbody>
</table>

**See Also**


Copyright © 2002, The KPD-Team
Mentalis.org ProxySocket Documentation
Socks5Handler.Authenticate Method

Starts the synchronous authentication process.

[Visual Basic] Private Sub Authenticate()

[C#]
private void Authenticate();

Exceptions

<table>
<thead>
<tr>
<th>Exception Type</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>System.Net.Sockets.SocketException</td>
<td>An operating system error occurs while accessing the Socket.</td>
</tr>
<tr>
<td>System.ObjectDisposedException</td>
<td>The Socket has been closed.</td>
</tr>
</tbody>
</table>

See Also

Socks5Handler Class | Socks5Handler Members |

Copyright © 2002, The KPD-Team
Socks5Handler.BeginNegotiate

Starts negotiating asynchronously with the SOCKS server.

Overload List

Starts negotiating asynchronously with the SOCKS server.

```csharp
    public override IAsyncProxyResult BeginNegotiate(string,int,HandShakeComplete,IPEndPoint);
```

Starts negotiating asynchronously with the SOCKS server.

```csharp
    public override IAsyncProxyResult BeginNegotiate(IPEndPoint,HandShakeComplete,IPEndPoint);
```

See Also


Copyright © 2002, The KPD-Team
Socks5Handler.BeginNegotiate Method (IPEndPoint, HandShakeComplete, IPEndPoint)

Starts negotiating asynchronously with the SOCKS server.


[C#]
public override IAsyncProxyResult BeginNegotiate(    IPEndPoint remoteEP, _    HandShakeComplete callback, _    IPEndPoint proxyEndPoint
);

Parameters

remoteEP
An IPEndPoint that represents the remote device.

callback
The method to call when the negotiation is complete.

proxyEndPoint
The IPEndPoint of the SOCKS proxy server.

Return Value
An IAsyncProxyResult that references the asynchronous connection.

See Also

Copyright © 2002, The KPD-Team
Mentalis.org ProxySocket Documentation
Socks5Handler.BeginNegotiate Method (String, Int32, HandShakeComplete, IPEndPoint)

Starts negotiating asynchronously with the SOCKS server.

[Visual Basic]Overrides Overloads Public Function BeginNegotiate ByVal host As String, _
ByVal port As Integer, _
ByVal proxyEndPoint As System.Net.IPEndPoint _

[C#]
public override IAsyncProxyResult BeginNegotiate(
    string host,
    int port,
    HandShakeComplete callback,
    IPEndPoint proxyEndPoint
);

Parameters

*host*
   The host to connect to.

*port*
   The port to connect to.

*callback*
   The method to call when the negotiation is complete.

*proxyEndPoint*
   The IPEndPoint of the SOCKS proxy server.

Return Value

An IAsyncProxyResult that references the asynchronous connection.

See Also

Socks5Handler Class | Socks5Handler Members |
Socks5Handler.GetEndPointBytes Method

Creates an array of bytes that has to be sent when the user wants to connect to a specific IPEndPoint.

[Visual Basic]
Private Function GetEndPointBytes(
    ByVal remoteEP As System.Net.IPEndPoint
) As Byte()

[C#]
private byte[] GetEndPointBytes(
    IPEndPoint remoteEP
);

Parameters

remoteEP
The IPEndPoint to connect to.

Return Value

An array of bytes that has to be sent when the user wants to connect to a specific IPEndPoint.

Exceptions

<table>
<thead>
<tr>
<th>Exception Type</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>System.ArgumentNullException</td>
<td>remoteEP is null.</td>
</tr>
</tbody>
</table>

See Also


Copyright © 2002, The KPD-Team
Socks5Handler.GetHostPortBytes Method

Creates an array of bytes that has to be sent when the user wants to connect to a specific host/port combination.

[Visual Basic] Private Function GetHostPortBytes(ByVal host As String, ByVal port As Integer) As Byte()  
[C#] private byte[] GetHostPortBytes(string host, int port);

Parameters

- **host**  
  The host to connect to.

- **port**  
  The port to connect to.

Return Value

An array of bytes that has to be sent when the user wants to connect to a specific host/port combination.

Exceptions

<table>
<thead>
<tr>
<th>Exception Type</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>System.ArgumentNullException</td>
<td>host is null.</td>
</tr>
<tr>
<td>System.ArgumentException</td>
<td>port or host is invalid.</td>
</tr>
</tbody>
</table>

See Also

[Socks5Handler Class] [Socks5Handler Members] [Org.Mentalis.Network.ProxySocket Namespace]
Socks5Handler.Negotiate

Starts negotiating with the SOCKS server.

Overload List

Starts negotiating with the SOCKS server.

    public override void Negotiate(IPEndPoint);

Starts negotiating with the SOCKS server.

    public override void Negotiate(string,int);

Starts negotiating with the SOCKS server.

    private void Negotiate(byte[]);

See Also


Copyright © 2002, The KPD-Team
Socks5Handler.Negotiate Method (Byte[])
System.Net.ProtocolViolationException

The proxy server uses an invalid protocol.

See Also


Copyright © 2002, The KPD-Team
Starts negotiating with the SOCKS server.

**[Visual Basic]**
```vbnet
Overloads Overrides Public Sub Negotiate(
    ByVal remoteEP As System.Net.IPEndPoint
)```

**[C#]**
```csharp
public override void Negotiate(  
    IPEndPoint remoteEP
);```

**Parameters**

*remoteEP*

The IPEndPoint to connect to.

**Exceptions**

<table>
<thead>
<tr>
<th>Exception Type</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>System.ArgumentNullException</code></td>
<td>remoteEP is null.</td>
</tr>
<tr>
<td><code>System.Net.Sockets.SocketException</code></td>
<td>An operating system error occurs while accessing the Socket.</td>
</tr>
<tr>
<td><code>System.ObjectDisposedException</code></td>
<td>The Socket has been closed.</td>
</tr>
</tbody>
</table>
See Also


Copyright © 2002, The KPD-Team
**Socks5Handler.Negotiate Method (String, Int32)**

Starts negotiating with the SOCKS server.

**[Visual Basic]**

```vbnet
Overrides Overloads Public Sub
ByVal host As String, _
ByVal port As Integer _
)
```

**[C#]**

```csharp
public override void Negotiate(
    string host, 
    int port
);
```

**Parameters**

- **host**
  - The host to connect to.

- **port**
  - The port to connect to.

**Exceptions**

<table>
<thead>
<tr>
<th>Exception Type</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>System.ArgumentNullException</td>
<td>host is null.</td>
</tr>
<tr>
<td>System.ArgumentException</td>
<td>port is invalid.</td>
</tr>
<tr>
<td>System.Net.Sockets.SocketException</td>
<td>An operating system error occurs while accessing the Socket.</td>
</tr>
<tr>
<td>System.ObjectDisposedException</td>
<td>The Socket has been closed.</td>
</tr>
</tbody>
</table>

See Also

- Socks5Handler Class
- Socks5Handler Members
- Socks5Handler.Negotiate Overload List

Copyright © 2002, The KPD-Team
Socks5Handler.OnAuthenticated Method

Called when the socket has been successfully authenticated with the server.

[Visual Basic] Private Sub OnAuthenticated( ByVal e As System.Exception )

[C#] private void OnAuthenticated( Exception e )

Parameters

e The exception that has occurred while authenticating, or null if no error occurred.

See Also


Copyright © 2002, The KPD-Team
Socks5Handler.OnAuthReceive Method

Called when an authentication reply has been received.

[Visual Basic]
Private Sub OnAuthReceive(ByVal ar As System.IAsyncResult)

[C#]
private void OnAuthReceive(IAsyncResult ar);

Parameters

ar
Stores state information for this asynchronous operation as well as any user-defined data.

See Also


Copyright © 2002, The KPD-Team
Socks5Handler.OnAuthSent Method

Called when the authentication bytes have been sent.

[Visual Basic] Private Sub OnAuthSent( ByVal ar As System.IAsyncResult )

[C#] private void OnAuthSent( IAsyncResult ar );

Parameters

ar
Stores state information for this asynchronous operation as well as any user-defined data.

See Also


Copyright © 2002, The KPD-Team
Socks5Handler.OnConnect Method

Called when the socket is connected to the remote server.

[Visual Basic] Private Sub OnConnect( ByVal ar As System.IAsyncResult )

[C#]
private void OnConnect(IAsyncResult ar);

Parameters

ar
Stores state information for this asynchronous operation as well as any user-defined data.

See Also


Copyright © 2002, The KPD-Team
Socks5Handler.OnReadLast Method

Called when the last bytes are read from the socket.

[Visual Basic] Private Sub OnReadLast(ByVal ar As System.IAsyncResult)

[C#]
private void OnReadLast(IAsyncResult ar);

Parameters

ar
Stores state information for this asynchronous operation as well as any user-defined data.

See Also


Copyright © 2002, The KPD-Team
Mentalis.org ProxySocket Documentation
Socks5Handler.OnReceive Method

Called when a connection reply has been received.

[Visual Basic]Private Sub OnReceive( ByVal ar As System.IAsyncResult )

[C#]
private void OnReceive( IAsyncResult ar );

Parameters

ar
Stores state information for this asynchronous operation as well as any user-defined data.

See Also


Copyright © 2002, The KPD-Team
Socks5Handler.OnSent Method

Called when the connection request has been sent.

[Visual Basic] Private Sub OnSent(_
    ByVal ar As System.IAsyncResult _
)

[C#] private void OnSent(_
    IAsyncResult ar
);

Parameters

ar
Stores state information for this asynchronous operation as well as any user-defined data.

See Also


Copyright © 2002, The KPD-Team
Socks5Handler.ProcessReply Method

Processes the received reply.

[Visual Basic] Private Sub ProcessReply(_ ByVal buffer As Byte() _)  

[C#]   private void ProcessReply(_ byte[] buffer _);  

Parameters

buffer  The received reply

Exceptions

<table>
<thead>
<tr>
<th>Exception Type</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>System.Net.ProtocolViolationException</td>
<td>The received reply is invalid.</td>
</tr>
</tbody>
</table>

See Also

Copyright © 2002, The KPD-Team
SocksHandler Class

Implements a specific version of the SOCKS protocol. This is an abstract class; it must be inherited.

For a list of all members of this type, see SocksHandler Members.

System.Object  SocksHandler

[Visual Basic]
MustInherit Class SocksHandler

[C#]
abstract class SocksHandler

Requirements


See Also


Copyright © 2002, The KPD-Team
# SocksHandler Members

## Public Instance Constructors

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SocksHandler Constructor</strong></td>
<td>Initializes a new instance of the SocksHandler class.</td>
</tr>
</tbody>
</table>

## Public Instance Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BeginNegotiate</strong></td>
<td>Overloaded. Starts negotiating asynchronously with a SOCKS proxy server.</td>
</tr>
<tr>
<td><strong>Equals</strong> (inherited from <strong>Object</strong>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>GetHashCode</strong> (inherited from <strong>Object</strong>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>GetType</strong> (inherited from <strong>Object</strong>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>Negotiate</strong></td>
<td>Overloaded. Starts negotiating with a SOCKS proxy server.</td>
</tr>
<tr>
<td><strong>ToString</strong> (inherited from <strong>Object</strong>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
</tbody>
</table>

## Protected Instance Fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ProtocolComplete</strong></td>
<td>Holds the address of the method to call when the SOCKS protocol has been completed.</td>
</tr>
</tbody>
</table>

## Protected Instance Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AsyncResult</strong></td>
<td>Gets or sets the return value of the BeginConnect call.</td>
</tr>
<tr>
<td><strong>Buffer</strong></td>
<td>Gets or sets a byte buffer.</td>
</tr>
<tr>
<td><strong>Received</strong></td>
<td>Gets or sets the number of bytes that have been received from</td>
</tr>
</tbody>
</table>
the remote proxy server.

| **Server** | Gets or sets the socket connection with the proxy server. |
| **Username** | Gets or sets the username to use when authenticating with the proxy server. |

### Protected Instance Methods

<table>
<thead>
<tr>
<th>Method Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AddressToBytes</strong></td>
<td>Converts an IP address to an array of bytes.</td>
</tr>
<tr>
<td><strong>Finalize</strong> (inherited from <strong>Object</strong>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>MemberwiseClone</strong> (inherited from <strong>Object</strong>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>PortToBytes</strong></td>
<td>Converts a port number to an array of bytes.</td>
</tr>
<tr>
<td><strong>ReadBytes</strong></td>
<td>Reads a specified number of bytes from the Server socket.</td>
</tr>
</tbody>
</table>

### Private Instance Fields

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>m_AsyncResult</strong></td>
<td>Holds the value of the AsyncResult property.</td>
</tr>
<tr>
<td><strong>m_Buffer</strong></td>
<td>Holds the value of the Buffer property.</td>
</tr>
<tr>
<td><strong>m_Received</strong></td>
<td>Holds the value of the Received property.</td>
</tr>
<tr>
<td><strong>m_Server</strong></td>
<td>Holds the value of the Server property.</td>
</tr>
<tr>
<td><strong>m_Username</strong></td>
<td>Holds the value of the Username property.</td>
</tr>
</tbody>
</table>

### See Also
Namespace
Copyright © 2002, The KPD-Team
SocksHandler Constructor

Initializes a new instance of the SocksHandler class.

```
[Visual Basic] Public Sub New( _
    ByVal server As System.Net.Sockets.Socket,
    ByVal user As String )
```

```
[C#] public SocksHandler( 
    Socket server, 
    string user 
);
```

**Parameters**

- **server**
  - The socket connection with the proxy server.

- **user**
  - The username to use when authenticating with the server.

**Exceptions**

<table>
<thead>
<tr>
<th>Exception Type</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>System.ArgumentNullException</td>
<td>server -or- user is null.</td>
</tr>
</tbody>
</table>

**See Also**


Copyright © 2002, The KPD-Team
SocksHandler Fields

The fields of the SocksHandler class are listed below. For a complete list of SocksHandler class members, see the SocksHandler Members topic.

### Protected Instance Fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ProtocolComplete</td>
<td>Holds the address of the method to call when the SOCKS protocol has been completed.</td>
</tr>
</tbody>
</table>

### Private Instance Fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>m_AsyncResult</td>
<td>Holds the value of the AsyncResult property.</td>
</tr>
<tr>
<td>m_Buffer</td>
<td>Holds the value of the Buffer property.</td>
</tr>
<tr>
<td>m_Received</td>
<td>Holds the value of the Received property.</td>
</tr>
<tr>
<td>m_Server</td>
<td>Holds the value of the Server property.</td>
</tr>
<tr>
<td>m_Username</td>
<td>Holds the value of the Username property.</td>
</tr>
</tbody>
</table>

See Also


Copyright © 2002, The KPD-Team
SocksHandler.m_AsyncResult Field

Holds the value of the AsyncResult property.


[C#] private IAsyncProxyResult m_AsyncResult;

See Also


Copyright © 2002, The KPD-Team
Mentalis.org ProxySocket Documentation
SocksHandler.m_Buffer Field

Holds the value of the Buffer property.

[Visual Basic] Private m_Buffer As Byte()

[C#] private byte[] m_Buffer;

See Also


Copyright © 2002, The KPD-Team
SocksHandler.m_Received Field

Holds the value of the Received property.

[Visual Basic]
Private m_Received As Integer

[C#]
private int m_Received;

See Also


Copyright © 2002, The KPD-Team
**SocksHandler.m_Server Field**

Holds the value of the Server property.

```
[Visual Basic] Private m_Server As System.Net.Sockets.Socket
```

```
[C#] private Socket m_Server;
```

See Also

[SocksHandler Class] [SocksHandler Members] [Org.Mentalis.Network.ProxySocket Namespace]

Copyright © 2002, The KPD-Team
SocksHandler.m_Username Field

Holds the value of the Username property.

[Visual Basic] Private m_Username As String

[C#]
private string m_Username;

See Also


Copyright © 2002, The KPD-Team
SocksHandler.ProtocolComplete Field

Holds the address of the method to call when the SOCKS protocol has been completed.

[Visual Basic] Protected ProtocolComplete As ("

[C#]
protected HandShakeComplete ProtocolComplete;

See Also

Copyright © 2002, The KPD-Team
Mentalis.org ProxySocket Documentation
The properties of the SocksHandler class are listed below. For a complete list of SocksHandler class members, see the SocksHandler Members topic.

### Protected Instance Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AsyncResult</td>
<td>Gets or sets the return value of the BeginConnect call.</td>
</tr>
<tr>
<td>Buffer</td>
<td>Gets or sets a byte buffer.</td>
</tr>
<tr>
<td>Received</td>
<td>Gets or sets the number of bytes that have been received from the remote proxy server.</td>
</tr>
<tr>
<td>Server</td>
<td>Gets or sets the socket connection with the proxy server.</td>
</tr>
<tr>
<td>Username</td>
<td>Gets or sets the username to use when authenticating with the proxy server.</td>
</tr>
</tbody>
</table>

See Also

- SocksHandler Class |

Copyright © 2002, The KPD-Team
SocksHandlerAsyncResult Property

Gets or sets the return value of the BeginConnect call.

[Visual Basic] Protected Property AsyncResult

[C#]  
protected IAsyncProxyResult AsyncResult {get;}

Property Value

An IAsyncProxyResult object that is the return value of the BeginConnect call.

See Also


Copyright © 2002, The KPD-Team
SocksHandler.Buffer Property

Gets or sets a byte buffer.

[Visual Basic] Protected Property Buffer As Byte()

[C#]
protected byte[] Buffer {get; set;}

Property Value

An array of bytes.

See Also


Copyright © 2002, The KPD-Team
Mentalis.org ProxySocket Documentation
SocksHandler.Received Property

Gets or sets the number of bytes that have been received from the remote proxy server.

[Visual Basic] Protected Property Received As

[C#]
protected int Received {get; set;}

Property Value

An integer that holds the number of bytes that have been received from the remote proxy server.

See Also


Copyright © 2002, The KPD-Team
SocksHandler.Server Property

Gets or sets the socket connection with the proxy server.


[C#]
protected System.Net.Sockets.Socket Server {get; set;}

Property Value

A Socket object that represents the connection with the proxy server.

Exceptions

<table>
<thead>
<tr>
<th>Exception Type</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>System.ArgumentNullException</td>
<td>The specified value is null.</td>
</tr>
</tbody>
</table>

See Also


Copyright © 2002, The KPD-Team
Mentalis.org ProxySocket Documentation
**SocksHandler.Username Property**

Gets or sets the username to use when authenticating with the proxy server.

[Visual Basic] Protected Property Username As

[C#] protected string Username {get; set;}

**Property Value**

A string that holds the username to use when authenticating with the proxy server.

**Exceptions**

<table>
<thead>
<tr>
<th>Exception Type</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>System.ArgumentNullException</td>
<td>The specified value is null.</td>
</tr>
</tbody>
</table>

**See Also**


[Copyright © 2002, The KPD-Team](http://example.com)
### SocksHandler Methods

The methods of the **SocksHandler** class are listed below. For a complete list of **SocksHandler** class members, see the **SocksHandler Members** topic.

#### Public Instance Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BeginNegotiate</strong></td>
<td>Overloaded. Starts negotiating asynchronously with a SOCKS proxy server.</td>
</tr>
<tr>
<td><strong>Equals</strong></td>
<td>(inherited from <strong>Object</strong>) Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>(inherited from <strong>Object</strong>) Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>(inherited from <strong>Object</strong>) Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>Negotiate</strong></td>
<td>Overloaded. Starts negotiating with a SOCKS proxy server.</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>(inherited from <strong>Object</strong>) Select the method name to go to the Microsoft documentation.</td>
</tr>
</tbody>
</table>

#### Protected Instance Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AddressToBytes</strong></td>
<td>Converts an IP address to an array of bytes.</td>
</tr>
<tr>
<td><strong>Finalize</strong></td>
<td>(inherited from <strong>Object</strong>) Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>MemberwiseClone</strong></td>
<td>(inherited from <strong>Object</strong>) Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>PortToBytes</strong></td>
<td>Converts a port number to an array of bytes.</td>
</tr>
<tr>
<td><strong>ReadBytes</strong></td>
<td>Reads a specified number of bytes from the Server socket.</td>
</tr>
</tbody>
</table>
See Also


Copyright © 2002, The KPD-Team
Mentalis.org ProxySocket Documentation
SocksHandler.AddressToBytes Method

Converts an IP address to an array of bytes.

[Visual Basic] Protected Function AddressToBytes (ByVal address As Long) As Byte()

[C#] protected byte[] AddressToBytes(long address);

Parameters

address
  The IP address to convert.

Return Value

An array of four bytes that represents the specified IP address.

See Also


Copyright © 2002, The KPD-Team
SocksHandler.BeginNegotiate

Starts negotiating asynchronously with a SOCKS proxy server.

Overload List

Starts negotiating asynchronously with a SOCKS proxy server.

public abstract IAsyncProxyResult BeginNegotiate(string, int, HandShakeComplete, IPEndPoint);

Starts negotiating asynchronously with a SOCKS proxy server.

public abstract IAsyncProxyResult BeginNegotiate(IPEndPoint, HandShakeComplete, IPEndPoint);

See Also


Copyright © 2002, The KPD-Team
starts negotiating asynchronously with a SOCKS proxy server.

**Parameters**

- **remoteEP**
  - An IPEndPoint that represents the remote device.

- **callback**
  - The method to call when the connection has been established.

- **proxyEndPoint**
  - The IPEndPoint of the SOCKS proxy server.

**Return Value**

An IAsyncProxyResult that references the asynchronous connection.

**See Also**

- [SocksHandler Class](#)
- [SocksHandler Members](#)
- [SocksHandler.BeginNegotiate Overload List](#)
SocksHandler.BeginNegotiate Method (String, Int32, HandShakeComplete, IPEndPoint)

Starts negotiating asynchronously with a SOCKS proxy server.


[C#] public abstract IAsyncProxyResult BeginNegotiate( string host, int port, HandShakeComplete callback, IPEndPoint proxyEndPoint );

Parameters

host
   The remote server to connect to.

port
   The remote port to connect to.

callback
   The method to call when the connection has been established.

proxyEndPoint
   The IPEndPoint of the SOCKS proxy server.

Return Value

An IAsyncProxyResult that references the asynchronous connection.

See Also

SocksHandler Class | SocksHandler Members |
Mentalis.org ProxySocket Documentation
SocksHandler.Negotiate

Starts negotiating with a SOCKS proxy server.

**Overload List**

Starts negotiating with a SOCKS proxy server.

```csharp
public abstract void Negotiate(IPEndPoint);
```

Starts negotiating with a SOCKS proxy server.

```csharp
public abstract void Negotiate(string,int);
```

**See Also**

[SocksHandler Class] | [SocksHandler Members] | [Org.Mentalis.Network.ProxySocket Namespace]

Copyright © 2002, The KPD-Team
SocksHandler.Negotiate Method (IPEndPoint)

Starts negotiating with a SOCKS proxy server.

[Visual Basic]MustOverride Overloads Public Sub Negotiate(
    ByVal remoteEP As System.Net.IPEndPoint)

[C#]
public abstract void Negotiate(
    IPEndPoint remoteEP);

Parameters

remoteEP
    The remote endpoint to connect to.

See Also


Copyright © 2002, The KPD-Team
SocksHandler.Negotiate Method (String, Int32)

Starts negotiating with a SOCKS proxy server.

```vbnet
MustOverride Overloads Public Sub Negotiate(
    ByVal host As String, _
    ByVal port As Integer _
)
```

```csharp
public abstract void Negotiate(
    string host, _
    int port
);
```

Parameters

- **host**
  - The remote server to connect to.

- **port**
  - The remote port to connect to.

See Also


Copyright © 2002, The KPD-Team
# SocksHandler.PortToBytes Method

Converts a port number to an array of bytes.

### [Visual Basic]
```vbnet
Protected Function PortToBytes(ByVal port As Integer) As Byte()
```

### [C#]
```csharp
protected byte[] PortToBytes(int port);
```

## Parameters

- **port**
  - The port to convert.

## Return Value

An array of two bytes that represents the specified port.

## See Also


Copyright © 2002, The KPD-Team
SocksHandler.ReadBytes Method

Reads a specified number of bytes from the Server socket.

[Visual Basic]Protected Function ReadBytes(ByVal count As Integer) As Byte()

[C#]
protected byte[] ReadBytes(int count);

Parameters

count
The number of bytes to return.

Return Value

An array of bytes.

Exceptions

<table>
<thead>
<tr>
<th>Exception Type</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>System.ArgumentException</td>
<td>The number of bytes to read is invalid.</td>
</tr>
<tr>
<td>System.Net.Sockets.SocketException</td>
<td>An operating system error occurs while accessing the Socket.</td>
</tr>
<tr>
<td>System.ObjectDisposedException</td>
<td>The Socket has been closed.</td>
</tr>
</tbody>
</table>

See Also


Copyright © 2002, The KPD-Team

## Namespace hierarchy

## Classes

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AuthMethod</strong></td>
<td>Implements a SOCKS authentication scheme.</td>
</tr>
<tr>
<td><strong>AuthNone</strong></td>
<td>This class implements the 'No Authentication' scheme.</td>
</tr>
<tr>
<td><strong>AuthUserPass</strong></td>
<td>This class implements the 'username/password authentication' scheme.</td>
</tr>
</tbody>
</table>
AuthMethod Class

Implements a SOCKS authentication scheme.
For a list of all members of this type, see AuthMethod Members.

System.Object AuthMethod

[Visual Basic]
MustInherit Class AuthMethod

[C#]
abstract class AuthMethod

Remarks
This is an abstract class; it must be inherited.

Requirements

See Also

Copyright © 2002, The KPD-Team
## AuthMethod Members

### Public Instance Constructors

<table>
<thead>
<tr>
<th>AuthMethod Constructor</th>
<th>Initializes an AuthMethod instance.</th>
</tr>
</thead>
</table>

### Public Instance Methods

<table>
<thead>
<tr>
<th>Authenticate</th>
<th>Authenticates the user.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>BeginAuthenticate</th>
<th>Authenticates the user asynchronously.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Equals (inherited from Object)</th>
<th>Select the method name to go to the Microsoft documentation.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>GetHashCode (inherited from Object)</th>
<th>Select the method name to go to the Microsoft documentation.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>GetType (inherited from Object)</th>
<th>Select the method name to go to the Microsoft documentation.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>ToString (inherited from Object)</th>
<th>Select the method name to go to the Microsoft documentation.</th>
</tr>
</thead>
</table>

### Protected Instance Fields

<table>
<thead>
<tr>
<th>CallBack</th>
<th>Holds the address of the method to call when the proxy has authenticated the client.</th>
</tr>
</thead>
</table>

### Protected Instance Properties

<table>
<thead>
<tr>
<th>Buffer</th>
<th>Gets or sets a byte array that can be used to store data.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Received</th>
<th>Gets or sets the number of bytes that have been received from the remote proxy server.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Server</th>
<th>Gets or sets the socket connection with the proxy server.</th>
</tr>
</thead>
</table>
Protected Instance Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Finalize</strong> (inherited from <strong>Object</strong>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>MemberwiseClone</strong> (inherited from <strong>Object</strong>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
</tbody>
</table>

Private Instance Fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>m_Buffer</strong></td>
<td>Holds the value of the Buffer property.</td>
</tr>
<tr>
<td><strong>m_Received</strong></td>
<td>Holds the value of the Received property.</td>
</tr>
<tr>
<td><strong>m_Server</strong></td>
<td>Holds the value of the Server property.</td>
</tr>
</tbody>
</table>

See Also

- [AuthMethod Class](#)

Copyright © 2002, The KPD-Team
AuthMethod Constructor

Initializes an AuthMethod instance.

[Visual Basic] Public Sub New( _
    ByVal server As System.Net.Sockets.Socket
)

[C#]
public AuthMethod(  
    Socket server
);

Parameters

server
The socket connection with the proxy server.

See Also


Copyright © 2002, The KPD-Team
AuthMethod Fields

The fields of the AuthMethod class are listed below. For a complete list of AuthMethod class members, see the AuthMethod Members topic.

Protected Instance Fields

| CallBack   | Holds the address of the method to call when the proxy has authenticated the client. |

Private Instance Fields

| m_Buffer   | Holds the value of the Buffer property. |
| m_Received | Holds the value of the Received property. |
| m_Server   | Holds the value of the Server property. |

See Also

- AuthMethod Class

Copyright © 2002, The KPD-Team
AuthMethod.CallBack Field

Holds the address of the method to call when the proxy has authenticated the client.


[C#]
protected HandShakeComplete CallBack;

See Also


Copyright © 2002, The KPD-Team
**AuthMethod.m_Buffer Field**

Holds the value of the Buffer property.

**[Visual Basic]**

```vbnet
Private m_Buffer As Byte()
```

**[C#]**

```csharp
private byte[] m_Buffer;
```

See Also


Copyright © 2002, The KPD-Team
**AuthMethod.m_Received Field**

Holds the value of the Received property.

**[Visual Basic]**
Private m_Received As Integer

**[C#]**
private int m_Received;

See Also


Copyright © 2002, The KPD-Team
AuthMethod.m_Server Field

Holds the value of the Server property.

[Visual Basic] Private m_Server As System.Net.Socket

[C#] private Socket m_Server;

See Also


Copyright © 2002, The KPD-Team
Mentalis.org ProxySocket Documentation
AuthMethod Properties

The properties of the **AuthMethod** class are listed below. For a complete list of **AuthMethod** class members, see the **AuthMethod Members** topic.

**Protected Instance Properties**

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Buffer</strong></td>
<td>Gets or sets a byt array that can be used to store data.</td>
</tr>
<tr>
<td><strong>Received</strong></td>
<td>Gets or sets the number of bytes that have been received from the remote proxy server.</td>
</tr>
<tr>
<td><strong>Server</strong></td>
<td>Gets or sets the socket connection with the proxy server.</td>
</tr>
</tbody>
</table>

**See Also**

- [AuthMethod Class](#)

*Copyright © 2002, The KPD-Team*
AuthMethod.Buffer Property

Gets or sets a byte array that can be used to store data.

[Visual Basic] Protected Property Buffer As Byte()

[C#] protected byte[] Buffer {get; set;}

Property Value

A byte array to store data.

See Also


Copyright © 2002, The KPD-Team
AuthMethod.Received Property

Gets or sets the number of bytes that have been received from the remote proxy server.

[Visual Basic] Protected Property Received As

[C#]
protected int Received {get; set;}

Property Value
An integer that holds the number of bytes that have been received from the remote proxy server.

See Also

Copyright © 2002, The KPD-Team
AuthMethod.Server Property

Gets or sets the socket connection with the proxy server.


[C#] protected System.Net.Sockets.Socket Server {get; set;}

Property Value

The socket connection with the proxy server.

See Also

Mentalis.org ProxySocket Documentation
AuthMethod Methods

The methods of the AuthMethod class are listed below. For a complete list of AuthMethod class members, see the AuthMethod Members topic.

Public Instance Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authenticate</td>
<td>Authenticates the user.</td>
</tr>
<tr>
<td>BeginAuthenticate</td>
<td>Authenticates the user asynchronously.</td>
</tr>
<tr>
<td>Equals (inherited from Object)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td>GetHashCode (inherited from Object)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td>GetType (inherited from Object)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td>ToString (inherited from Object)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
</tbody>
</table>

Protected Instance Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Finalize (inherited from Object)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td>MemberwiseClone (inherited from Object)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
</tbody>
</table>

See Also


Copyright © 2002, The KPD-Team
Mentalis.org ProxySocket Documentation
AuthMethod.Authenticate Method

Authenticates the user.

[Visual Basic] MustOverride Public Sub Authenticate()

[C#]
public abstract void Authenticate();

Exceptions

<table>
<thead>
<tr>
<th>Exception Type</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>System.Net.Sockets.SocketException</td>
<td>An operating system error occurs while accessing the Socket.</td>
</tr>
<tr>
<td>System.ObjectDisposedException</td>
<td>The Socket has been closed.</td>
</tr>
</tbody>
</table>

See Also


Copyright © 2002, The KPD-Team
AuthMethod.BeginAuthenticate Method

Authenticates the user asynchronously.


[C#]
public abstract void BeginAuthenticate(HandShakeComplete callback);

Parameters

callback
The method to call when the authentication is complete.

Exceptions

<table>
<thead>
<tr>
<th>Exception Type</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>System.Net.Sockets.SocketException</td>
<td>An operating system error occurs while accessing the Socket.</td>
</tr>
<tr>
<td>System.ObjectDisposedException</td>
<td>The Socket has been closed.</td>
</tr>
</tbody>
</table>
See Also


Copyright © 2002, The KPD-Team
Mentalis.org ProxySocket Documentation
**AuthNone Class**

This class implements the 'No Authentication' scheme.

For a list of all members of this type, see [AuthNone Members](#).

```csharp
sealed class AuthNone : AuthMethod
```

```vbnet
NotInheritable Class AuthNone
Inherits AuthMethod
```

**Requirements**

- **Assembly:** Org.Mentalis.Network.ProxySocket.dll

**See Also**

- [AuthNone Members](#)

*Copyright © 2002, The KPD-Team*
## AuthNone Members

### Public Instance Constructors

<table>
<thead>
<tr>
<th>AuthNone Constructor</th>
<th>Initializes an AuthNone instance.</th>
</tr>
</thead>
</table>

### Public Instance Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authenticate</td>
<td>Authenticates the user.</td>
</tr>
<tr>
<td>BeginAuthenticate</td>
<td>Authenticates the user asynchronously.</td>
</tr>
<tr>
<td>Equals (inherited from Object)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td>GetHashCode (inherited from Object)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td>GetType (inherited from Object)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td>ToString (inherited from Object)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
</tbody>
</table>

### Protected Instance Fields

<table>
<thead>
<tr>
<th>Field</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>CallBack</td>
<td></td>
</tr>
</tbody>
</table>

### Protected Instance Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Buffer (inherited from AuthMethod)</td>
<td>Gets or sets a byt array that can be used to store data.</td>
</tr>
<tr>
<td>Received (inherited from AuthMethod)</td>
<td>Gets or sets the number of bytes that have been received from the remote proxy server.</td>
</tr>
<tr>
<td>Server (inherited from AuthMethod)</td>
<td>Gets or sets the socket connection with the proxy server.</td>
</tr>
</tbody>
</table>

### Protected Instance Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Finalize (inherited from Object)</td>
<td>Select the method name to go to</td>
</tr>
<tr>
<td>MemberwiseClone (inherited from <strong>Object</strong>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
</tbody>
</table>

**See Also**


Copyright © 2002, The KPD-Team
AuthNone Constructor

Initializes an AuthNone instance.

[Visual Basic]
Public Sub New(_
    ByVal server As System.Net.Sockets.Socket
)

[C#]
public AuthNone(
    Socket server
);

Parameters

- server
  The socket connection with the proxy server.

See Also


Copyright © 2002, The KPD-Team
AuthNone Fields

The fields of the AuthNone class are listed below. For a complete list of AuthNone class members, see the AuthNone Members topic.

Protected Instance Fields

| CallBack |

See Also


Copyright © 2002, The KPD-Team
AuthNone.CallBack Field


[C#] protected HandShakeComplete CallBack;

See Also

Copyright © 2002, The KPD-Team
The methods of the **AuthNone** class are listed below. For a complete list of **AuthNone** class members, see the [AuthNone Members](#) topic.

### Public Instance Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Authenticate</strong></td>
<td>Authenticates the user.</td>
</tr>
<tr>
<td><strong>BeginAuthenticate</strong></td>
<td>Authenticates the user asynchronously.</td>
</tr>
<tr>
<td><strong>Equals</strong> (inherited from <strong>Object</strong>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>GetHashCode</strong> (inherited from <strong>Object</strong>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>GetType</strong> (inherited from <strong>Object</strong>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>ToString</strong> (inherited from <strong>Object</strong>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
</tbody>
</table>

### Protected Instance Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Finalize</strong> (inherited from <strong>Object</strong>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>MemberwiseClone</strong> (inherited from <strong>Object</strong>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
</tbody>
</table>

**See Also**


Copyright © 2002, The KPD-Team
Mentalis.org ProxySocket Documentation
AuthNone.Authenticate Method

Authenticates the user.

[Visual Basic] Overrides Public Sub Authenticate()

[C#]
public override void Authenticate();

See Also


Copyright © 2002, The KPD-Team
Authenticates the user asynchronously.

### Parameters

**callback**

The method to call when the authentication is complete.

### Remarks

This method immediately calls the callback method.

### See Also

AuthUserPass Class

This class implements the 'username/password authentication' scheme.

For a list of all members of this type, see AuthUserPass Members.

System.Object  AuthMethod  
AuthUserPass

[Visual Basic]  
NotInheritable  Class AuthUserPass  
Inherits AuthMethod

[C#]  
sealed class AuthUserPass : AuthMethod

Requirements

Namespace


See Also

AuthUserPass Members  | 

Copyright © 2002, The KPD-Team
## AuthUserPass Members

### Public Instance Constructors

|--------------------------|------------------------------------------|

### Public Instance Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Authenticate</strong></td>
<td>Starts the authentication process.</td>
</tr>
<tr>
<td><strong>BeginAuthenticate</strong></td>
<td>Starts the asynchronous authentication process.</td>
</tr>
<tr>
<td><strong>Equals</strong> (inherited from <strong>Object</strong>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>GetHashCode</strong> (inherited from <strong>Object</strong>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>GetType</strong> (inherited from <strong>Object</strong>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>ToString</strong> (inherited from <strong>Object</strong>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
</tbody>
</table>

### Protected Instance Fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CallBack</strong></td>
<td></td>
</tr>
</tbody>
</table>

### Protected Instance Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Buffer</strong> (inherited from <strong>AuthMethod</strong>)</td>
<td>Gets or sets a byte array that can be used to store data.</td>
</tr>
<tr>
<td><strong>Received</strong> (inherited from <strong>AuthMethod</strong>)</td>
<td>Gets or sets the number of bytes that have been received from the remote proxy server.</td>
</tr>
<tr>
<td><strong>Server</strong> (inherited from <strong>AuthMethod</strong>)</td>
<td>Gets or sets the socket connection with the proxy server.</td>
</tr>
</tbody>
</table>
Protected Instance Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Finalize</strong> (inherited from <strong>Object</strong>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><strong>MemberwiseClone</strong> (inherited from <strong>Object</strong>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
</tbody>
</table>

Private Instance Fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>m_Password</strong></td>
<td>Holds the value of the Password property.</td>
</tr>
<tr>
<td><strong>m_Username</strong></td>
<td>Holds the value of the Username property.</td>
</tr>
</tbody>
</table>

Private Instance Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Password</strong></td>
<td>Gets or sets the password to use when authenticating with the proxy server.</td>
</tr>
<tr>
<td><strong>Username</strong></td>
<td>Gets or sets the username to use when authenticating with the proxy server.</td>
</tr>
</tbody>
</table>

Private Instance Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>GetAuthenticationBytes</strong></td>
<td>Creates an array of bytes that has to be sent if the user wants to authenticate with the username/password authentication scheme.</td>
</tr>
<tr>
<td><strong>OnReceive</strong></td>
<td>Called when the socket received an authentication reply.</td>
</tr>
<tr>
<td><strong>OnSent</strong></td>
<td>Called when the authentication bytes have been sent.</td>
</tr>
</tbody>
</table>

See Also

**AuthUserPass Class** |
AuthUserPass Constructor

Initializes a new AuthUserPass instance.

[Visual Basic] Public Sub New( _
    ByVal server As System.Net.Sockets.Socket, _
    ByVal user As String, _
    ByVal pass As String _
)

[C#]
public AuthUserPass(
    Socket server,
    string user,
    string pass
);

Parameters

server
   The socket connection with the proxy server.

user
   The username to use.

pass
   The password to use.

Exceptions

<table>
<thead>
<tr>
<th>Exception Type</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>System.ArgumentNullException</td>
<td>user -or- pass is null.</td>
</tr>
</tbody>
</table>

See Also


Copyright © 2002, The KPD-Team
# AuthUserPass Fields

The fields of the **AuthUserPass** class are listed below. For a complete list of **AuthUserPass** class members, see the [AuthUserPass Members](#) topic.

## Protected Instance Fields

<table>
<thead>
<tr>
<th>Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>CallBack</td>
</tr>
</tbody>
</table>

## Private Instance Fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>m_Password</td>
<td>Holds the value of the Password property.</td>
</tr>
<tr>
<td>m_Username</td>
<td>Holds the value of the Username property.</td>
</tr>
</tbody>
</table>

See Also

- [AuthUserPass Class](#)  

Copyright © 2002, The KPD-Team
AuthUserPass.CallBack Field


[C#] protected HandShakeComplete CallBack;

See Also


Copyright © 2002, The KPD-Team
AuthUserPass.m_Password Field

Holds the value of the Password property.

[Visual Basic] Private m_Password As String

[C#] private string m_Password;

See Also


Copyright © 2002, The KPD-Team
Mentalis.org ProxySocket Documentation
AuthUserPass.m_Username Field

Holds the value of the Username property.

[Visual Basic] Private m_Username As String

[C#] private string m_Username;

See Also


Copyright © 2002, The KPD-Team
**AuthUserPass Properties**

The properties of the **AuthUserPass** class are listed below. For a complete list of **AuthUserPass** class members, see the [AuthUserPass Members](#) topic.

### Protected Instance Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Buffer</strong> (inherited from AuthMethod)</td>
<td>Gets or sets a byte array that can be used to store data.</td>
</tr>
<tr>
<td><strong>Received</strong> (inherited from AuthMethod)</td>
<td>Gets or sets the number of bytes that have been received from the remote proxy server.</td>
</tr>
<tr>
<td><strong>Server</strong> (inherited from AuthMethod)</td>
<td>Gets or sets the socket connection with the proxy server.</td>
</tr>
</tbody>
</table>

### Private Instance Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Password</strong></td>
<td>Gets or sets the password to use when authenticating with the proxy server.</td>
</tr>
<tr>
<td><strong>Username</strong></td>
<td>Gets or sets the username to use when authenticating with the proxy server.</td>
</tr>
</tbody>
</table>

### See Also

- [AuthUserPass Class](#)

*Copyright © 2002, The KPD-Team*
Mentalis.org ProxySocket Documentation
AuthUserPass.Password Property

Gets or sets the password to use when authenticating with the proxy server.

[Visual Basic]
Private Property Password As String

[C#]
private string Password {get; set;}

Property Value

The password to use when authenticating with the proxy server.

Exceptions

<table>
<thead>
<tr>
<th>Exception Type</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>System.ArgumentNullException</td>
<td>The specified value is null.</td>
</tr>
</tbody>
</table>

See Also


Copyright © 2002, The KPD-Team
**AuthUserPass.Username Property**

Gets or sets the username to use when authenticating with the proxy server.

**[Visual Basic]**

```
Private Property Username As String
```

**[C#]**

```
private string Username {get; set;}
```

**Property Value**

The username to use when authenticating with the proxy server.

**Exceptions**

<table>
<thead>
<tr>
<th>Exception Type</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>System.ArgumentNullException</td>
<td>The specified value is null.</td>
</tr>
</tbody>
</table>

**See Also**


Copyright © 2002, The KPD-Team
**AuthUserPass Methods**

The methods of the `AuthUserPass` class are listed below. For a complete list of `AuthUserPass` class members, see the `AuthUserPass Members` topic.

### Public Instance Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Authenticate</code></td>
<td>Starts the authentication process.</td>
</tr>
<tr>
<td><code>BeginAuthenticate</code></td>
<td>Starts the asynchronous authentication process.</td>
</tr>
<tr>
<td><code>Equals</code> (inherited from <code>Object</code>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><code>GetHashCode</code> (inherited from <code>Object</code>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><code>GetType</code> (inherited from <code>Object</code>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><code>ToString</code> (inherited from <code>Object</code>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
</tbody>
</table>

### Protected Instance Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Finalize</code> (inherited from <code>Object</code>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
<tr>
<td><code>MemberwiseClone</code> (inherited from <code>Object</code>)</td>
<td>Select the method name to go to the Microsoft documentation.</td>
</tr>
</tbody>
</table>

### Private Instance Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>GetAuthenticationBytes</code></td>
<td>Creates an array of bytes that has to be sent if the user wants to authenticate with the username/password authentication scheme.</td>
</tr>
<tr>
<td><code>OnReceive</code></td>
<td>Called when the socket received an authentication reply.</td>
</tr>
<tr>
<td><strong>OnSent</strong></td>
<td>Called when the authentication bytes have been sent.</td>
</tr>
</tbody>
</table>

**See Also**

- [AuthUserPass Class](#)

Copyright © 2002, The KPD-Team
Mentalis.org ProxySocket Documentation
AuthUserPass.Authenticate Method

Starts the authentication process.

[Visual Basic] Overrides Public Sub Authenticate()

[C#]
public override void Authenticate();

See Also


Copyright © 2002, The KPD-Team
AuthUserPass.BeginAuthenticate Method

Starts the asynchronous authentication process.


[C#] public override void BeginAuthenticate( HandShakeComplete callback );

Parameters

callback
The method to call when the authentication is complete.

See Also


Copyright © 2002, The KPD-Team
AuthUserPass.GetAuthenticationBytes Method

Creates an array of bytes that has to be sent if the user wants to authenticate with the username/password authentication scheme.

[Visual Basic] Private Function GetAuthenticationBytes()

[C#] 
private byte[] GetAuthenticationBytes();

Return Value

An array of bytes that has to be sent if the user wants to authenticate with the username/password authentication scheme.

See Also


Copyright © 2002, The KPD-Team
AuthUserPass.OnReceive Method

Called when the socket received an authentication reply.

[Visual Basic] Private Sub OnReceive( _
   ByVal ar As System.IAsyncResult _
)

[C#]
private void OnReceive(    
   IAsyncResult ar
);

Parameters

ar
Stores state information for this asynchronous operation as well as any user-defined data.

See Also


Copyright © 2002, The KPD-Team
AuthUserPass.OnSent Method

Called when the authentication bytes have been sent.

[Visual Basic] Private Sub OnSent( _
    ByVal ar As System.IAsyncResult _
) 

[C#] 
private void OnSent( 
    IAsyncResult ar 
); 

Parameters

ar
Stores state information for this asynchronous operation as well as any user-defined data.

See Also


Copyright © 2002, The KPD-Team
Source Code License

Copyright © 2002, The KPD-Team
All rights reserved.
http://www.mentalis.org/

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- Neither the name of the KPD-Team, nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE,
EVEN IF ADVISED
OF THE POSSIBILITY OF SUCH DAMAGE.
System.Object - AuthMethod
  - AuthNone
  - AuthUserPass