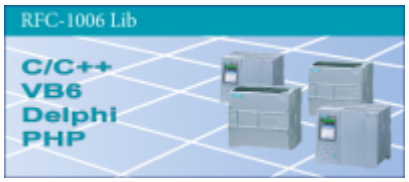
RFC 1006 Lib .NET Advanced

RFC 1006 Client- und Serverentwicklung leicht gemacht



Mit wenigen Zeilen Code zum Erfolg

Download

Features

Voraussetzungen

.NET Runtime

- .NET Framework 4.6 oder höher, oder eine .NET Standard 2.0 kompatible Runtime wie .NET Core/.NET 5

Sprachen

- C#
- VB.NET

RFC 1006 Lib .NET Advanced Class Library

Hier finden Sie die Dokumentation der API des RFC 1006 Frameworks:

- Online Handbuch: RFC 1006 Lib .NET Advanced Class Library

[1)] Mit Ihrem „License Code“ wird das Paket zur produktiven Vollversion.

- Rfc1006LibNet.Advanced Namespace
  - Licenser Class
  - Rfc1006Base Class
  - Rfc1006Client Class
  - Rfc1006IPEndPoint Class
  - Rfc1006Server Class
  - Rfc1006Status Enumeration
- Rfc1006LibNet.Advanced.EventArgs Namespace
  - ConnectionEventArgs Class
  - StatusChangedEventArgs Class
  - TimeoutEventArgs Class
  - TransferEventArgs Class

Licenser Class

Defines mechanism required to register the component.

Inheritance Hierarchy

System.Object
Rfc1006LibNet.Advanced.Licenser

Namespace: Rfc1006LibNet.Advanced
Assembly: Rfc1006LibNet.Advanced (in Rfc1006LibNet.Advanced.dll) Version: 1.0.2.0 (1.0.5.3)

Syntax

C#

```
public static class Licenser
```

The **Licenser** type exposes the following members.

Properties

—-	Name	Description
	LicenseKey	Gets or sets the license key used to license this product.

Top

Remarks

It is recommended to set the LicenseKey in the main entry point of the application. If there is no entry point available set the LicenseKey property before the first use of the assembly.

See Also

Reference

Rfc1006LibNet.Advanced Namespace

Rfc1006Base Class

Provides a generic interface to communicate via RFC1006.

Inheritance Hierarchy

System.Object
Rfc1006LibNet.Advanced.Rfc1006Base
Rfc1006LibNet.Advanced.Rfc1006Client
Rfc1006LibNet.Advanced.Rfc1006Server

Namespace: Rfc1006LibNet.Advanced
Assembly: Rfc1006LibNet.Advanced (in Rfc1006LibNet.Advanced.dll) Version: 1.0.2.0 (1.0.5.3)

Syntax

C#

```
public abstract class Rfc1006Base : IDisposable
```

The **Rfc1006Base** type exposes the following members.

Constructors

—-	Name	Description
	Rfc1006Base	Initializes a new instance of the **Rfc1006Base** class.

Top

Methods

—-	Name	Description
	CloseSocket	CLoses the Socket
	Connect()	Wait onto connection to remote.
	Connect(Int32)	Waits until a communication is connected or the *timeout* exceeded.
	Dispose()	Releases all resources used by the **Rfc1006Base** class.
	Dispose(Boolean)	Releases the unmanaged resources used by the **Rfc1006Base** class and optionally releases the managed resources.
	Finalize	Finalizes an instance of the **Rfc1006Base** class class. (Overrides Object.Finalize().)
	NewId	Allocates a new id for the server or client.
	OnConnected	Raises the Connected event using the specified event data.
	OnConnecting	Raises the Connecting event using the specified event data.

—-	Name	Description
	OnReceived	Raises the Received event using the specified event data.
	OnReceiving	Raises the Receiving event using the specified event data.
	OnTimeout	Raises the Timeout event using the specified event data.
	OnTransmitted	Raises the Transmitted event using the specified event data.
	OnTransmitting	Raises the Transmitting event using the specified event data.
	Receive	Reads a byte from the RFC1006 communication send by the opposite service.
	ReceiveString	Reads a Rfc1006Packet from the RFC1006 communication send by the opposite service.
	SetKeepAlive	Sets the keep alive time using the specified *aliveTime* and information.
	SetStatus(Rfc1006Status, Rfc1006Base)	Sets the status of the connection using the specified *value*, *sender* information.
	SetStatus(Rfc1006Status, Rfc1006Base, Exception)	Sets the status of the connection using the specified *value*, *sender* and *exception* information.
	Start	Starts the communication.
	Stop	Stops the communication.
	ToString	Returns a string that represents the current object. (Overrides Object.ToString().)
	Transmit(Byte[])	Writes the length of *buffer* Bytes to the RFC1006 communication.
	Transmit(String)	Writes the length of *buffer* Bytes to the RFC1006 communication.
	Transmit(Byte[], Int32)	Writes *length* Bytes to the RFC1006 communication.
	Transmit(Byte[], Int32, Int32)	Writes *length* Bytes to the RFC1006 communication.

Top

Fields

—-	Name	Description
	AliveInterval	Stores the interval in milliseconds when the client will send a type of broadcast.
	DefaultAliveTime	Stores the default value of the alive time.
	DefaultReceiveTimeout	Stores the default value of ReceiveTimeout
	DefaultTransmitTimeout	Stores the default value of TransmitTimeout
	TextEncoding	Stores the value of Encoding used when transmitting or receiving strings and for encoding TSAP values.

Top

Properties

—-	Name	Description

—-	Name	Description
	AliveTime	Gets or sets the time in milliseconds when the server will disconnect if the client is not responding
	AutoReceive	Gets or sets a value indicating whether the receiving thread is turned on or off.
	EndPoint	Gets or sets a network endpoint as an IP address, port number, remote TSAP and a local TSAP.
	FastAcknowlege	FastAcknowlege If partner is sending small packages in short intervals. This makes the TCP-traffic faster V 1.0.3.0
	LocalId	Gets or sets the local ID of the RFC1006 communication.
	ReceiveTimeout	Gets or sets the time in milliseconds when the service will cancel the receive.
	RemoteId	Gets or sets the remote ID of the RFC1006 communication.
	Socket	Gets or sets the socket of the RFC1006 communication.
	Status	Gets the status of the connection.
	TransmitTimeout	Gets or sets the time in milliseconds when the service will cancel the transmit.

Top

Events

—-	Name	Description
	Connected	Occurs before the connection.
	Connecting	Occurs after the connection.
	Received	Occurs after a receive.
	Receiving	Occurs before a receive.
	StatusChanged	Occurs if something gone wrong during a transfer.
	Timeout	Occurs when a timeout is appeared.
	Transmitted	Occurs after a transmit.
	Transmitting	Occurs before a transmit.

Top

See Also

Reference

Rfc1006LibNet.Advanced Namespace

Rfc1006Client Class

Defines an implementation of the Rfc1006Base class to provide a generic interface to communicate via RFC1006 from a client perspective with any RFC1006 server.

Inheritance Hierarchy

System.Object
Rfc1006LibNet.Advanced.Rfc1006Base
Rfc1006LibNet.Advanced.Rfc1006Client

Namespace: Rfc1006LibNet.Advanced
Assembly: Rfc1006LibNet.Advanced (in Rfc1006LibNet.Advanced.dll) Version: 1.0.2.0 (1.0.5.3)

Syntax

C#

```
public class Rfc1006Client : Rfc1006Base
```

The **Rfc1006Client** type exposes the following members.

Constructors

—-	Name	Description
	Rfc1006Client(Rfc1006IPEndPoint)	Initializes a new instance of the **Rfc1006Client** class using the specified *endPoint* information.
	Rfc1006Client(Int32, Rfc1006IPEndPoint)	Initializes a new instance of the **Rfc1006Client** class using the specified *endPoint* and *connectionTimeout* information.
	Rfc1006Client(Rfc1006IPEndPoint, Int32)	Initializes a new instance of the **Rfc1006Client** class using the specified *endPoint* and *aliveTime* information.
	Rfc1006Client(Int32, Rfc1006IPEndPoint, Int32)	Initializes a new instance of the **Rfc1006Client** class using the specified *endPoint*, *connectionTimeout* and *aliveTime* information.
	Rfc1006Client(IPAddress, Byte[], Byte[])	Initializes a new instance of the **Rfc1006Client** class using the specified *address*, *remoteTSAP* and *localTSAP* information.
	Rfc1006Client(IPAddress, String, String)	Initializes a new instance of the **Rfc1006Client** class using the specified *address*, *remoteTSAP* and *localTSAP* information.
	Rfc1006Client(String, Byte[], Byte[])	Initializes a new instance of the **Rfc1006Client** class using the specified *address*, *remoteTSAP* and *localTSAP* information.
	Rfc1006Client(String, String, String)	Initializes a new instance of the **Rfc1006Client** class using the specified *address*, *remoteTSAP* and *localTSAP* information.

—-	Name	Description
	Rfc1006Client(Int32, IPAddress, Byte[], Byte[])	Initializes a new instance of the **Rfc1006Client** class using the specified *address*, *remoteTSAP*, *localTSAP* and *connectionTimeout* information.
	Rfc1006Client(Int32, IPAddress, String, String)	Initializes a new instance of the **Rfc1006Client** class using the specified *address*, *remoteTSAP*, *localTSAP* and *connectionTimeout* information.
	Rfc1006Client(Int32, String, String, String)	Initializes a new instance of the **Rfc1006Client** class using the specified *address*, *remoteTSAP*, *localTSAP* and *connectionTimeout* information.
	Rfc1006Client(IPAddress, Byte[], Byte[], Int32)	Initializes a new instance of the **Rfc1006Client** class using the specified *address*, *remoteTSAP*, *localTSAP* and *aliveTime* information.
	Rfc1006Client(IPAddress, Int32, Byte[], Byte[])	Initializes a new instance of the **Rfc1006Client** class using the specified *address*, *remoteTSAP* and *localTSAP* information.
	Rfc1006Client(IPAddress, Int32, String, String)	Initializes a new instance of the **Rfc1006Client** class using the specified *address*, *remoteTSAP* and *localTSAP* information.
	Rfc1006Client(IPAddress, String, String, Int32)	Initializes a new instance of the **Rfc1006Client** class using the specified *address*, *remoteTSAP*, *localTSAP* and *aliveTime* information.
	Rfc1006Client(String, Byte[], Byte[], Int32)	Initializes a new instance of the **Rfc1006Client** class using the specified *address*, *remoteTSAP*, *localTSAP* and *aliveTime* information.
	Rfc1006Client(String, Int32, Byte[], Byte[])	Initializes a new instance of the **Rfc1006Client** class using the specified *address*, *port*, *remoteTSAP* and *localTSAP* information and DefaultConnectionTimeout.
	Rfc1006Client(String, Int32, String, String)	Initializes a new instance of the **Rfc1006Client** class using the specified *address*, *remoteTSAP* and *localTSAP* information.
	Rfc1006Client(String, String, String, Int32)	Initializes a new instance of the **Rfc1006Client** class using the specified *address*, *remoteTSAP*, *localTSAP* and *aliveTime* information.
	Rfc1006Client(Int32, IPAddress, Byte[], Byte[], Int32)	Initializes a new instance of the **Rfc1006Client** class using the specified *address*, *remoteTSAP*, *connectionTimeout*, *localTSAP* and *aliveTime* information.
	Rfc1006Client(Int32, IPAddress, Int32, Byte[], Byte[])	Initializes a new instance of the **Rfc1006Client** class using the specified *address*, *remoteTSAP*, *localTSAP* and *connectionTimeout* information.
	Rfc1006Client(Int32, IPAddress, Int32, String, String)	Initializes a new instance of the **Rfc1006Client** class using the specified *address*, *remoteTSAP*, *localTSAP* and *connectionTimeout* information.
	Rfc1006Client(Int32, IPAddress, String, String, Int32)	Initializes a new instance of the **Rfc1006Client** class using the specified *address*, *remoteTSAP*, *connectionTimeout*, *localTSAP* and *aliveTime* information.
	Rfc1006Client(Int32, String, Byte[], Byte[], Int32)	Initializes a new instance of the **Rfc1006Client** class using the specified *address*, *remoteTSAP*, *connectionTimeout*, *localTSAP* and *aliveTime* information.
	Rfc1006Client(Int32, String, Int32, Byte[], Byte[])	Initializes a new instance of the **Rfc1006Client** class using the specified *address*, *port*, *remoteTSAP*, *localTSAP* and *connectionTimeout* information and DefaultAliveTime.

—-	Name	Description
	Rfc1006Client(Int32, String, Int32, String, String)	Initializes a new instance of the **Rfc1006Client** class using the specified *address*, *remoteTSAP*, *localTSAP* and *connectionTimeout* information.
	Rfc1006Client(Int32, String, String, String, Int32)	Initializes a new instance of the **Rfc1006Client** class using the specified *address*, *remoteTSAP*, *connectionTimeout*, *localTSAP* and *aliveTime* information.
	Rfc1006Client(IPAddress, Int32, Byte[], Byte[], Int32)	Initializes a new instance of the **Rfc1006Client** class using the specified *address*, *remoteTSAP*, *localTSAP* and *aliveTime* information.
	Rfc1006Client(IPAddress, Int32, String, String, Int32)	Initializes a new instance of the **Rfc1006Client** class using the specified *address*, *remoteTSAP*, *localTSAP* and *aliveTime* information.
	Rfc1006Client(String, Int32, Byte[], Byte[], Int32)	Initializes a new instance of the **Rfc1006Client** class using the specified *address*, *port*, *remoteTSAP*, *localTSAP* and *aliveTime* information and DefaultAliveTime.
	Rfc1006Client(String, Int32, String, String, Int32)	Initializes a new instance of the **Rfc1006Client** class using the specified *address*, *remoteTSAP*, *localTSAP* and *aliveTime* information.
	Rfc1006Client(Int32, IPAddress, Int32, Byte[], Byte[], Int32)	Initializes a new instance of the **Rfc1006Client** class using the specified *address*, *remoteTSAP*, *connectionTimeout*, *localTSAP* and *aliveTime* information.
	Rfc1006Client(Int32, IPAddress, Int32, String, String, Int32)	Initializes a new instance of the **Rfc1006Client** class using the specified *address*, *remoteTSAP*, *connectionTimeout*, *localTSAP* and *aliveTime* information.
	Rfc1006Client(Int32, String, Int32, Byte[], Byte[], Int32)	Initializes a new instance of the **Rfc1006Client** class using the specified *address*, *port*, *remoteTSAP*, *connectionTimeout*, *localTSAP* and *aliveTime* information and DefaultAliveTime.
	Rfc1006Client(Int32, String, Int32, String, String, Int32)	Initializes a new instance of the **Rfc1006Client** class using the specified *address*, *remoteTSAP*, *connectionTimeout*, *localTSAP* and *aliveTime* information.

Top

Methods

—-	Name	Description
	CloseSocket	CLoses the Socket (Inherited from Rfc1006Base.)
	Connect()	Connects the client to the server using the specified AliveTime and ConnectionTimeout information. (Overrides Rfc1006Base.Connect().)
	Connect(Int32)	Connects the client to the server using the specified AliveTime and ConnectionTimeout information or the *timeout* exceeded. (Overrides Rfc1006Base.Connect(Int32).)
	Dispose()	Releases all resources used by the Rfc1006Base class. (Inherited from Rfc1006Base.)
	Dispose(Boolean)	Releases the unmanaged resource used by the **Rfc1006Client** class and optionally releases the managed resource. (Overrides Rfc1006Base.Dispose(Boolean).)

—-	Name	Description
	Finalize	Finalizes an instance of the Rfc1006Base class class. (Inherited from Rfc1006Base.)
	OnConnected	Raises the Connected event using the specified event data. (Inherited from Rfc1006Base.)
	OnConnecting	Raises the Connecting event using the specified event data. (Inherited from Rfc1006Base.)
	OnDisconnected	Raises the Disconnected event using the specified event data.
	OnDisconnecting	Raises the Disconnecting event using the specified event data.
	OnReceived	Raises the Received event using the specified event data. (Inherited from Rfc1006Base.)
	OnReceiving	Raises the Receiving event using the specified event data. (Inherited from Rfc1006Base.)
	OnTimeout	Raises the Timeout event using the specified event data. (Inherited from Rfc1006Base.)
	OnTransmitted	Raises the Transmitted event using the specified event data. (Inherited from Rfc1006Base.)
	OnTransmitting	Raises the Transmitting event using the specified event data. (Inherited from Rfc1006Base.)
	Receive	Reads a byte from the RFC1006 communication send by the opposite service. (Inherited from Rfc1006Base.)
	ReceiveString	Reads a Rfc1006Packet from the RFC1006 communication send by the opposite service. (Inherited from Rfc1006Base.)
	SetKeepAlive	Sets the keep alive time using the specified *aliveTime* and information. (Inherited from Rfc1006Base.)
	SetStatus(Rfc1006Status, Rfc1006Base)	Sets the status of the connection using the specified *value*, *sender* information. (Inherited from Rfc1006Base.)
	SetStatus(Rfc1006Status, Rfc1006Base, Exception)	Sets the status of the connection using the specified *value*, *sender* and *exception* information. (Inherited from Rfc1006Base.)
	Start	Starts the connecting of the client to the server. (Overrides Rfc1006Base.Start().)
	Stop	Disconnects the client from the server. (Overrides Rfc1006Base.Stop().)
	ToString	Returns a string that represents the current object. (Inherited from Rfc1006Base.)
	Transmit(Byte[])	Writes the length of *buffer* Bytes to the RFC1006 communication. (Inherited from Rfc1006Base.)
	Transmit(String)	Writes the length of *buffer* Bytes to the RFC1006 communication. (Inherited from Rfc1006Base.)
	Transmit(Byte[], Int32)	Writes *length* Bytes to the RFC1006 communication. (Inherited from Rfc1006Base.)
	Transmit(Byte[], Int32, Int32)	Writes *length* Bytes to the RFC1006 communication. (Inherited from Rfc1006Base.)

Top

Fields

—-	Name	Description
	DefaultConnectionTimeout	Stores the default connection timeout in milliseconds (5s).
	DefaultReconnectionInterval	Stores the default reconnection interval in milliseconds (1s).

Top

Properties

—-	Name	Description
	AliveTime	Gets or sets the time in milliseconds when the server will disconnect if the client is not responding (Inherited from Rfc1006Base.)
	AutoReceive	Gets or sets a value indicating whether the receiving thread is turned on or off. (Inherited from Rfc1006Base.)
	AutoReconnect	Gets or sets a value indicating whether the automatic reconnection is enabled or not.
	ConnectionTimeout	Gets or sets the timeout of the connection in milliseconds.
	EndPoint	Gets or sets a network endpoint as an IP address, port number, remote TSAP and a local TSAP. (Inherited from Rfc1006Base.)
	FastAcknowlege	FastAcknowlege If partner is sending small packages in short intervals. This makes the TCP-traffic faster V 1.0.3.0 (Inherited from Rfc1006Base.)
	LocalId	Gets or sets the local ID of the RFC1006 communication. (Inherited from Rfc1006Base.)
	ReceiveTimeout	Gets or sets the time in milliseconds when the service will cancel the receive. (Inherited from Rfc1006Base.)
	ReconnectionInterval	Gets or sets the reconnection interval in milliseconds.
	RemoteId	Gets or sets the remote ID of the RFC1006 communication. (Inherited from Rfc1006Base.)
	Socket	Gets or sets the socket of the RFC1006 communication. (Inherited from Rfc1006Base.)
	Status	Gets the status of the connection. (Inherited from Rfc1006Base.)
	TransmitTimeout	Gets or sets the time in milliseconds when the service will cancel the transmit. (Inherited from Rfc1006Base.)

Top

Events

—-	Name	Description
	Connected	Occurs before the connection. (Inherited from Rfc1006Base.)

—-	Name	Description
	Connecting	Occurs after the connection. (Inherited from Rfc1006Base.)
	Disconnected	Occurs before the disconnection
	Disconnecting	Occurs after the disconnection
	Received	Occurs after a receive. (Inherited from Rfc1006Base.)
	Receiving	Occurs before a receive. (Inherited from Rfc1006Base.)
	StatusChanged	Occurs if something gone wrong during a transfer. (Inherited from Rfc1006Base.)
	Timeout	Occurs when a timeout is appeared. (Inherited from Rfc1006Base.)
	Transmitted	Occurs after a transmit. (Inherited from Rfc1006Base.)
	Transmitting	Occurs before a transmit. (Inherited from Rfc1006Base.)

Top

See Also

Reference

Rfc1006LibNet.Advanced Namespace

Rfc1006IPEndPoint Class

Represents a network endpoint as an IP address, port number, remote TSAP and a local TSAP.

Inheritance Hierarchy

System.Object
System.Net.EndPoint
System.Net.IPEndPoint
Rfc1006LibNet.Advanced.Rfc1006IPEndPoint

Namespace: Rfc1006LibNet.Advanced
Assembly: Rfc1006LibNet.Advanced (in Rfc1006LibNet.Advanced.dll) Version: 1.0.2.0 (1.0.5.3)

Syntax

C#

```
public class Rfc1006IPEndPoint : IPEndPoint
```

The **Rfc1006IPEndPoint** type exposes the following members.

Constructors

—-	Name	Description
	Rfc1006IPEndPoint(Byte[], Byte[])	Initializes a new instance of the **Rfc1006IPEndPoint** class using the specified *remoteTSAP* and *localTSAP* information and the DefaultPort.
	Rfc1006IPEndPoint(String, String)	Initializes a new instance of the **Rfc1006IPEndPoint** class using the specified *remoteTSAP* and *localTSAP* information and the DefaultPort.
	Rfc1006IPEndPoint(Int32, Byte[], Byte[])	Initializes a new instance of the **Rfc1006IPEndPoint** class using the specified *port*, *localTSAP* and *remoteTSAP* information.
	Rfc1006IPEndPoint(Int32, String, String)	Initializes a new instance of the **Rfc1006IPEndPoint** class using the specified *port*, *localTSAP* and *remoteTSAP* information.
	Rfc1006IPEndPoint(IPAddress, Byte[], Byte[])	Initializes a new instance of the **Rfc1006IPEndPoint** class using the specified *address*, *remoteTSAP* and *localTSAP* information and the DefaultPort.
	Rfc1006IPEndPoint(IPAddress, String, String)	Initializes a new instance of the **Rfc1006IPEndPoint** class using the specified *address*, *remoteTSAP* and *localTSAP* information and the DefaultPort.
	Rfc1006IPEndPoint(String, Byte[], Byte[])	Initializes a new instance of the **Rfc1006IPEndPoint** class using the specified *address*, *remoteTSAP* and *localTSAP* information and the DefaultPort.

—-	Name	Description
	Rfc1006IPEndPoint(String, String, String)	Initializes a new instance of the **Rfc1006IPEndPoint** class using the specified *address*, *remoteTSAP* and *localTSAP* information and the DefaultPort.
	Rfc1006IPEndPoint(IPAddress, Int32, Byte[], Byte[])	Initializes a new instance of the **Rfc1006IPEndPoint** class using the specified *address*, *port*, *localTSAP* and *remoteTSAP* information.
	Rfc1006IPEndPoint(IPAddress, Int32, String, String)	Initializes a new instance of the **Rfc1006IPEndPoint** class using the specified *address*, *port*, *localTSAP* and *remoteTSAP* information.
	Rfc1006IPEndPoint(String, Int32, Byte[], Byte[])	Initializes a new instance of the **Rfc1006IPEndPoint** class using the specified *address*, *port*, *localTSAP* and *remoteTSAP* information.
	Rfc1006IPEndPoint(String, Int32, String, String)	Initializes a new instance of the **Rfc1006IPEndPoint** class using the specified *address*, *port*, *localTSAP* and *remoteTSAP* information.

Top

Methods

—-	Name	Description
	ToString	Returns the IP address, port number, remote TSAP and local TSAP of the specified endpoint. (Overrides IPEndPoint.ToString().)

Top

Fields

—-	Name	Description
	DefaultPort	Stores the default port number associated with the address.
	TsapEncoding	Stores the value of Encoding used for encoding and decoding TSAP strings.

Top

Properties

—-	Name	Description
	LocalTSAP	Gets or sets the local TSAP of the endpoint.
	ReceivedLocalTSAP	Gets or sets the local TSAP of the endpoint received from the partner.
	ReceivedRemoteTSAP	Gets or sets the remote TSAP of the endpoint received from the partner.
	RemoteTSAP	Gets or sets the remote TSAP of the endpoint.

Top

See Also

Reference

Rfc1006LibNet.Advanced Namespace

Rfc1006Server Class

Defines an implementation of the Rfc1006Base class to provide a generic interface to communicate via RFC1006 from a Server perspective with any RFC1006 client.

Inheritance Hierarchy

System.Object
Rfc1006LibNet.Advanced.Rfc1006Base
Rfc1006LibNet.Advanced.Rfc1006Server

Namespace: Rfc1006LibNet.Advanced
Assembly: Rfc1006LibNet.Advanced (in Rfc1006LibNet.Advanced.dll) Version: 1.0.2.0 (1.0.5.3)

Syntax

C#

```
public class Rfc1006Server : Rfc1006Base
```

The **Rfc1006Server** type exposes the following members.

Constructors

—-	Name	Description
	Rfc1006Server(Rfc1006IPEndPoint)	Initializes a new instance of the **Rfc1006Server** class using the specified *endPoint* information.
	Rfc1006Server(Byte[], Byte[])	Initializes a new instance of the **Rfc1006Server** class using the specified *remoteTSAP* and *localTSAP* information.
	Rfc1006Server(String, String)	Initializes a new instance of the **Rfc1006Server** class using the specified *remoteTSAP* and *localTSAP* information.
	Rfc1006Server(Rfc1006IPEndPoint, Int32)	Initializes a new instance of the **Rfc1006Server** class using the specified *endPoint* information.
	Rfc1006Server(Byte[], Byte[], Int32)	Initializes a new instance of the **Rfc1006Server** class using the specified *remoteTSAP*, and *localTSAP* and *aliveTime* information.
	Rfc1006Server(IPAddress, Byte[], Byte[])	Initializes a new instance of the **Rfc1006Server** class using the specified *address*, *remoteTSAP* and *localTSAP* information.
	Rfc1006Server(IPAddress, String, String)	Initializes a new instance of the **Rfc1006Server** class using the specified *address*, *remoteTSAP* and *localTSAP* information.
	Rfc1006Server(IPEndPoint, Byte[], Byte[])	Initializes a new instance of the **Rfc1006Server** class using the specified *endPoint*, *remoteTSAP* and *localTSAP* information.

—-	Name	Description
	Rfc1006Server(String, Byte[], Byte[])	Initializes a new instance of the **Rfc1006Server** class using the specified *address*, *remoteTSAP* and *localTSAP* information.
	Rfc1006Server(String, String, Int32)	Initializes a new instance of the **Rfc1006Server** class using the specified *remoteTSAP*, *localTSAP* and *aliveTime* information.
	Rfc1006Server(String, String, String)	Initializes a new instance of the **Rfc1006Server** class using the specified *address*, *remoteTSAP* and *localTSAP* information.
	Rfc1006Server(IPAddress, Byte[], Byte[], Int32)	Initializes a new instance of the **Rfc1006Server** class using the specified *address*, *remoteTSAP*, *localTSAP* and *aliveTime* information.
	Rfc1006Server(IPAddress, String, String, Int32)	Initializes a new instance of the **Rfc1006Server** class using the specified *address*, *remoteTSAP* , *localTSAP* and *aliveTime* information.
	Rfc1006Server(IPEndPoint, Byte[], Byte[], Int32)	Initializes a new instance of the **Rfc1006Server** class using the specified *endPoint*, *remoteTSAP*, *localTSAP* and *aliveTime* information.
	Rfc1006Server(String, Byte[], Byte[], Int32)	Initializes a new instance of the **Rfc1006Server** class using the specified *address*, *remoteTSAP*, *localTSAP* and *aliveTime* information.
	Rfc1006Server(String, String, String, Int32)	Initializes a new instance of the **Rfc1006Server** class using the specified *address*, *remoteTSAPlocalTSAP* and *aliveTime* information.

Top

Methods

—-	Name	Description
	CloseSocket	CLoses the Socket (Inherited from Rfc1006Base.)
	Connect()	Waits until a client is connected. (Overrides Rfc1006Base.Connect().)
	Connect(Int32)	Waits until a client is connected or the *timeout* exceeded. (Overrides Rfc1006Base.Connect(Int32).)
	Dispose()	Releases all resources used by the Rfc1006Base class. (Inherited from Rfc1006Base.)
	Dispose(Boolean)	Releases the unmanaged resources used by the **Rfc1006Server** class and optionally releases the managed resources. (Overrides Rfc1006Base.Dispose(Boolean).)
	Finalize	Finalizes an instance of the Rfc1006Base class class. (Inherited from Rfc1006Base.)
	OnConnected	Raises the Connected event using the specified event data. (Inherited from Rfc1006Base.)
	OnConnecting	Raises the Connecting event using the specified event data. (Inherited from Rfc1006Base.)
	OnReceived	Raises the Received event using the specified event data. (Inherited from Rfc1006Base.)

—-	Name	Description
	OnReceiving	Raises the Receiving event using the specified event data. (Inherited from Rfc1006Base.)
	OnStarted	Raises the Started event using the specified event data.
	OnStarting	Raises the Starting event using the specified event data.
	OnStopped	Raises the Stopped event using the specified event data.
	OnStopping	Raises the Stopping event using the specified event data.
	OnTimeout	Raises the Timeout event using the specified event data. (Inherited from Rfc1006Base.)
	OnTransmitted	Raises the Transmitted event using the specified event data. (Inherited from Rfc1006Base.)
	OnTransmitting	Raises the Transmitting event using the specified event data. (Inherited from Rfc1006Base.)
	Receive	Reads a byte from the RFC1006 communication send by the opposite service. (Inherited from Rfc1006Base.)
	ReceiveString	Reads a Rfc1006Packet from the RFC1006 communication send by the opposite service. (Inherited from Rfc1006Base.)
	SetKeepAlive	Sets the keep alive time using the specified *aliveTime* and information. (Inherited from Rfc1006Base.)
	SetStatus(Rfc1006Status, Rfc1006Base)	Sets the status of the connection using the specified *value*, *sender* information. (Inherited from Rfc1006Base.)
	SetStatus(Rfc1006Status, Rfc1006Base, Exception)	Sets the status of the connection using the specified *value*, *sender* and *exception* information. (Inherited from Rfc1006Base.)
	Start	Starts the server and accept client. (Overrides Rfc1006Base.Start().)
	Stop	Stops the server and disconnect client. (Overrides Rfc1006Base.Stop().)
	StopAll	Stops all currently active servers and disconnect the clients.
	ToString	Returns a string that represents the current object. (Inherited from Rfc1006Base.)
	Transmit(Byte[])	Writes the length of *buffer* Bytes to the RFC1006 communication. (Inherited from Rfc1006Base.)
	Transmit(String)	Writes the length of *buffer* Bytes to the RFC1006 communication. (Inherited from Rfc1006Base.)
	Transmit(Byte[], Int32)	Writes *length* Bytes to the RFC1006 communication. (Inherited from Rfc1006Base.)
	Transmit(Byte[], Int32, Int32)	Writes *length* Bytes to the RFC1006 communication. (Inherited from Rfc1006Base.)

Top

Properties

—-	Name	Description
	AliveTime	Gets or sets the time in milliseconds when the server will disconnect if the client is not responding (Inherited from Rfc1006Base.)
	AutoReceive	Gets or sets a value indicating whether the receiving thread is turned on or off. (Inherited from Rfc1006Base.)
	EndPoint	Gets or sets a network endpoint as an IP address, port number, remote TSAP and a local TSAP. (Inherited from Rfc1006Base.)
	FastAcknowlege	FastAcknowlege If partner is sending small packages in short intervals. This makes the TCP-traffic faster V 1.0.3.0 (Inherited from Rfc1006Base.)
	LocalId	Gets or sets the local ID of the RFC1006 communication. (Inherited from Rfc1006Base.)
	ReceiveTimeout	Gets or sets the time in milliseconds when the service will cancel the receive. (Inherited from Rfc1006Base.)
	RemoteId	Gets or sets the remote ID of the RFC1006 communication. (Inherited from Rfc1006Base.)
	Socket	Gets or sets the socket of the RFC1006 communication. (Inherited from Rfc1006Base.)
	Status	Gets the status of the connection. (Inherited from Rfc1006Base.)
	TransmitTimeout	Gets or sets the time in milliseconds when the service will cancel the transmit. (Inherited from Rfc1006Base.)

Top

Events

—-	Name	Description
	Connected	Occurs before the connection. (Inherited from Rfc1006Base.)
	Connecting	Occurs after the connection. (Inherited from Rfc1006Base.)
	Received	Occurs after a receive. (Inherited from Rfc1006Base.)
	Receiving	Occurs before a receive. (Inherited from Rfc1006Base.)
	Started	Occurs after the server starts
	Starting	Occurs before the server starts
	StatusChanged	Occurs if something gone wrong during a transfer. (Inherited from Rfc1006Base.)
	Stopped	Occurs after the server stops
	Stopping	Occurs before the server stops
	Timeout	Occurs when a timeout is appeared. (Inherited from Rfc1006Base.)
	Transmitted	Occurs after a transmit. (Inherited from Rfc1006Base.)
	Transmitting	Occurs before a transmit. (Inherited from Rfc1006Base.)

Top

See Also

Reference

Rfc1006LibNet.Advanced Namespace

Rfc1006Status Enumeration

Represents the status of connection.

Namespace: Rfc1006LibNet.Advanced
Assembly: Rfc1006LibNet.Advanced (in Rfc1006LibNet.Advanced.dll) Version: 1.0.2.0 (1.0.5.3)

Syntax

C#

```
public enum Rfc1006Status
```

Members

—-	Member name	Value	Description
	ConnectionClosed	0	After the close
	ConnectionClosing	1	Before the close
	Connected	2	After the connection
	Connecting	3	Before the connection
	Disconnected	4	After the disconnection
	Disconnecting	5	Before the disconnection
	Opened	6	After the open
	Opening	7	Before the open
	RfcConnecting	8	Before the RFC connection
	RfcConnected	9	After the RFC connection
	TcpConnecting	10	Before the TCP connection
	TcpConnected	11	After the TCP connection
	Starting	16	Before the server start
	Started	17	After the server start
	Stopping	18	Before the server stop
	Stopped	19	After the server stop

See Also

Reference

Rfc1006LibNet.Advanced Namespace

StatusChangedEventArgs Class

Provides data for the StatusChanged event.

Inheritance Hierarchy

System.Object
System.EventArgs
Rfc1006LibNet.Advanced.EventArgs.StatusChangedEventArgs

Namespace: Rfc1006LibNet.Advanced.EventArgs
Assembly: Rfc1006LibNet.Advanced (in Rfc1006LibNet.Advanced.dll) Version: 1.0.2.0 (1.0.5.3)

Syntax

C#

```
public class StatusChangedEventArgs : EventArgs
```

The **StatusChangedEventArgs** type exposes the following members.

Constructors

—-	Name	Description
	StatusChangedEventArgs	Initializes a new instance of the **StatusChangedEventArgs** class using the specified *source*, *exception* information.

Top

Properties

—-	Name	Description
	Exception	Gets the error that was occurred during the RFC1006 communication.
	OrginalSource	Gets the original reporting source as determined by pure hit testing, before any possible Source adjustment by a parent class.
	PreviousStatus	Gets the previous status of the connection.
	Source	Gets or sets a reference to the object that raised the event.
	Status	Gets the actual status of the connection.

Top

See Also

Reference

Rfc1006LibNet.Advanced.EventArgs Namespace

TimeoutEventArgs Class

Provides data for the Timeout event.

Inheritance Hierarchy

System.Object
System.EventArgs
Rfc1006LibNet.Advanced.EventArgs.TimeoutEventArgs

Namespace: Rfc1006LibNet.Advanced.EventArgs
Assembly: Rfc1006LibNet.Advanced (in Rfc1006LibNet.Advanced.dll) Version: 1.0.2.0 (1.0.5.3)

Syntax

C#

```
public class TimeoutEventArgs : EventArgs
```

The **TimeoutEventArgs** type exposes the following members.

Constructors

—-	Name	Description
	TimeoutEventArgs	Initializes a new instance of the **TimeoutEventArgs** class using the specified *source*, *exception* information.

Top

Properties

—-	Name	Description
	Exception	Gets the error that was occurred during the RFC1006 communication.
	OrginalSource	Gets the original reporting source as determined by pure hit testing, before any possible Source adjustment by a parent class.
	Source	Gets or sets a reference to the object that raised the event.

Top

See Also

Reference

Rfc1006LibNet.Advanced.EventArgs Namespace

Inhaltsverzeichnis

RFC 1006 Client- und Serverentwicklung leicht gemacht 1
Download 2
Features 2
Voraussetzungen 2
RFC 1006 Lib .NET Advanced Class Library 2
Inheritance Hierarchy 4
Syntax 4
Properties 4
Remarks 4
See Also 4
Reference 4
Inheritance Hierarchy 5
Syntax 5
Constructors 5
Methods 5
Fields 6
Properties 6
Events 7
See Also 7
Reference 7
Inheritance Hierarchy 8
Syntax 8
Constructors 8
Methods 10
Fields 12
Properties 12
Events 12
See Also 13
Reference 13
Inheritance Hierarchy 14
Syntax 14
Constructors 14
Methods 15
Fields 15
Properties 15
See Also 16
Reference 16
Inheritance Hierarchy 17
Syntax 17
Constructors 17
Methods 18
Properties 19
Events 20
See Also 21
Reference 21
Syntax 22
Members 22
See Also 22
Reference 22
Inheritance Hierarchy 23
Syntax 23