

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 <i>timeout</i> 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 <i>aliveTime</i> and information.
	SetStatus(Rfc1006Status, Rfc1006Base)	Sets the status of the connection using the specified <i>value</i> , <i>sender</i> information.
	SetStatus(Rfc1006Status, Rfc1006Base, Exception)	Sets the status of the connection using the specified <i>value</i> , <i>sender</i> and <i>exception</i> 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 <i>buffer</i> Bytes to the RFC1006 communication.
	Transmit(String)	Writes the length of <i>buffer</i> Bytes to the RFC1006 communication.
	Transmit(Byte[], Int32)	Writes <i>length</i> Bytes to the RFC1006 communication.
	Transmit(Byte[], Int32, Int32)	Writes <i>length</i> 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
	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.
	FastAcknowledge	FastAcknowledge 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.
	Remoteld	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](#)

Table of Contents

Inheritance Hierarchy	1
Syntax	1
Constructors	1
Methods	1
Fields	2
Properties	2
Events	3
See Also	3
Reference	3