

SavoySecsISocket ActiveX Control
User Guide

1 Revision History

Version	Date	Name	Description
1.00	Nov, 12 th , 2009	Hikaru Okada	Created as new document

2 Table of Contents

1	Revision History.....	2
2	Table of Contents	3
3	SavoySecsSocket.....	4
3.1	Properties.....	5
3.1.1	Appearance	5
3.1.2	BorderStyle	6
3.1.3	Connect	7
3.1.4	DeviceID	8
3.1.5	IniFileName.....	9
3.1.6	IniSection	10
3.1.7	IPAddress	11
3.1.8	Log.....	12
3.1.9	LogBakCount.....	13
3.1.10	LogFileName	14
3.1.11	LogSize.....	15
3.1.12	LogVerbose	16
3.1.13	Master.....	17
3.1.14	MSEC	18
3.1.15	MyPortNumber.....	19
3.1.16	OneOnOne	20
3.1.17	PortNumber	21
3.1.18	Retry	22
3.1.19	Server	23
3.1.20	T1	24
3.1.21	T2	25
3.1.22	T3	26
3.1.23	T4	27
3.2	Methods	28
3.2.1	AboutBox	28
3.2.2	Disconnect	29
3.2.3	LoadIniFile	30
3.2.4	Send	31
3.2.5	Setup	32
3.3	Events	35
3.3.1	Connected	35
3.3.2	Disconnected	36
3.3.3	Problem	37
3.3.4	Received.....	39
3.3.5	Sent	40

3 SavoySecsISocket

SavoySecsISocket control is an assistant product to develop SEMI E4 (SECS-I) compliant communication application software, but all datum are transferred through TCP/IP. It is also known as SECS Over LAN. SavoySecsISocket control can be used for either equipment side development or host side development. Usually SavoySecsISocket control will be used with SavoySecsII control.

Properties

Name	Description
Appearance	Gets or sets the value that determines the appearance of a SavoySecsISocket control.
BorderStyle	Gets or sets whether the SavoySecsISocket control has a border.
Connect	Gets or sets the TCP/IP connection status.
DeviceID	Gets or sets the device ID.
IniFileName	Gets or sets INI file name to read/write settings.
IniSection	Gets or sets section name in INI file to read/write settings.
IP Address	Gets or sets the IP address of server computer for TCP/IP connection.
Log	Gets or sets whether logging is enabled.
LogBakCount	Gets or sets the number of back-up file for logging.
LogFileName	Get or sets the log file name.
LogSize	Gets or sets the log file size in kilobyte.
LogVerbose	Get or sets whether logging for detailed part is enabled.
Master	Gets or sets whether SavoySecsISocket control acts as master or slave.
MSEC	Gets or sets whether Mitsubishi SECS protocol is enabled.
MyPortNumber	Gets or sets local portnumber for TCP/IP connection.
OneOnOne	Gets or sets whether connection is limited to 1 on 1 connection only.
PortNumber	Gets or sets the port number for TCP/IP connection.
Retry	Gets or sets the number of attempt in case of send error.
Server	Gets or sets the entity type.
T1	Gets or sets the T1 time out in milliseconds.
T2	Gets or sets the T2 time out in milliseconds.
T3	Gets or sets the T3 time out in milliseconds.
T4	Gets or sets the T4 time out in milliseconds.

Methods

Name	Description
AboutBox	Opens version information dialog box on the screen.
Disconnect	Disconnects specified TCP/IP connection.
LoadIniFile	Loads settings from INI file and initialize properties.
Send	Send specified SECS-II message.
Setup	Opens setup dialog box on the screen.

Events

Name	Description
Connected	Notifies that TCP/IP connection has been established.
Disconnected	Notifies that TCP/IP connection has been disconnected.
Problem	Notifies that error has occurred.
Received	Notifies that SavoySecsI control received SECS-II message.
Sent	Notifies that SECS-II message has been sent.

3.1 Properties

3.1.1 Appearance

Gets or sets the value that determines the appearance of a SavoySecsISocket control.

Value	Description
0	Flat
1	Etched

Syntax

Visual Basic 6.0

```
Appearance As Integer
```

Visual C++ 6.0

```
short GetAppearance()  
void SetAppearance(short)
```

Example

Visual Basic 6.0

```
.Appearance = 0 ' flat  
.Appearance = 1 ' sunken
```

Visual C++ 6.0

```
m_ctrl.SetAppearance(0); // flat  
m_ctrl.SetAppearance(1); // sunken
```

Remarks

Persistent property.

See Also

3.1.2 BorderStyle

Gets or sets whether the SavoySecsI control has a border.

Value	Description
0	No border
1	Fixed single border

Syntax

```
Visual Basic 6.0  
  
BorderStyle As Integer
```

```
Visual C++ 6.0  
  
short GetBorderStyle()  
void SetBorderStyle(short)
```

Example

```
Visual Basic 6.0  
  
.BorderStyle = 0 ' no border  
.BorderStyle = 1 ' with border
```

```
Visual C++ 6.0  
  
m_ctrl.SetBorderStyle(0); // no border  
m_ctrl.SetBorderStyle(1); // with border
```

Remarks

Persistent property.

See Also

3.1.3 Connect

Gets or sets the TCP/IP connection status. If Connect property is set to true where Server property is false (client), SavoySecsISocket control will attempt to establish connection. If connection has not been established, Connect property will be set to false.

If Connect property is set to true where Server property is true (server), SavoySecsISocket control will attempt to open server port and start listening to prepare for future incoming connection. At this time connection has not been established until client connected server. Therefore, even no connection was made, Connect property would be True, when server port was opened.

If user sets False to Connect property, existing connection will be disconnected. User doesn't have to worry about connection to make sure it is disconnected when application is closing. SavoySecsISocket control will take care disconnection automatically. When Server property was set to True and Connect property is set to False, all existing connections with clients will be disconnected.

Syntax

Visual Basic 6.0
Connect As Boolean
Visual C++ 6.0
<pre> BOOL GetConnect() void SetConnect(BOOL) </pre>

Example

Visual Basic 6.0
<pre> .IPAddress = "terminal_server" .PortNumber = 5000 .MyPortNumber = 0 .Server = False .Connect = True ' Open If Not .Connect then MsgBox "Cannot connect to server!" End If </pre>
Visual C++ 6.0
<pre> m_ctrl.SetIPAddress("terminal_server"); m_ctrl.SetPortNumber(5000); m_ctrl.SetLocalPortNumber(0); m_ctrl.SetServer(false); m_ctrl.SetConnect(true); // Open if(!m_ctrl.GetConnect()) MessageBox("Cannot connect to server!"); </pre>

Remarks

If Connect property on server side was set to False, connection would be disconnected. Therefore, Connect property on client side would also be set to False.

See Also

3.1.4 DeviceID

Gets or sets the device ID. Device ID is 15 bits starting at second bit of SECS-II header.

For SECS-I following header structure is used.

Byte	Description
1	R Device ID
2	
3	W Stream
4	Function
5	E Block number
6	
7	Source ID
8	
9	Transaction ID
10	

Syntax

Visual Basic 6.0
DeviceID As Long

Visual C++ 6.0
long GetDeviceID() void SetDeviceID(long)

Example

Visual Basic 6.0
.DeviceID = 1

Visual C++ 6.0
m_ctrl.SetDeviceID(1);

Remarks

Persistent property.

See Also

3.1.5 IniFileName

Gets or sets INI file name to read/write settings. If INI file name is either full path name or containing relative reference of folder name, INI file will be created and read in such location. Otherwise, INI file will be created in Windows OS system folder. For this reason, it is highly recommended using with folder name. If current directory is the location, add “.” at the beginning.

Either “/” (slash) or “\” (back slash) can be used for separator of folder name.

Syntax

Visual Basic 6.0

```
IniFileName As String
```

Visual C++ 6.0

```
CString GetIniFileName()  
void SetIniFileName(LPCTSTR)
```

Example

Visual Basic 6.0

```
.IniFileName = “./Savoy.ini”
```

Visual C++ 6.0

```
m_ctrl.SetIniFileName(“./Savoy.ini”);
```

Remarks

Persistent property.

See Also

3.1.6 IniSection

Gets or sets section name in INI file to read/write settings. If multiple SavoySecsISocket control were embedded in one project, it is possible to share same INI file by changing section name.

Default setting is "SavoySecsISocket".

Syntax

Visual Basic 6.0

```
IniSection As String
```

Visual C++ 6.0

```
CString GetIniSection()  
void SetIniSection(LPCTSTR)
```

Example

Visual Basic 6.0

```
.IniSection = "Host" ' [Host] section
```

Visual C++ 6.0

```
m_ctrl.IniSection("Host"); // [Host] section
```

Remarks

Persistent property.

See Also

3.1.7 IPAddress

Gets or sets the IP address of server computer for TCP/IP connection. IPAddress property will be ignored if the Server property is set to true, because server listens incoming connection.

When connecting local computer (same computer), use "127.0.0.1" or "" (empty) string.

It is possible to use computer name instead of IP address.

Syntax

Visual Basic 6.0

```
IPAddress As String
```

Visual C++ 6.0

```
CString GetIPAddress()  
void SetIPAddress(LPCTSTR)
```

Example

Visual Basic 6.0

```
.IPAddress = ""  
.PortNumber = 5000  
.LocalPortNumber = 0  
.Server = False  
.Connect = True ' Open  
If Not .Connect then  
    MsgBox "Cannot connect to server!"  
End If
```

Visual C++ 6.0

```
m_ctrl.SetIPAddress("");  
m_ctrl.SetPortNumber(5000);  
m_ctrl.SetLocalPortNumber(0);  
m_ctrl.SetServer(false);  
m_ctrl.SetConnect(true); // Open  
if(!m_ctrl.GetConnect())  
    MessageBox("Cannot connect to server!");
```

Remarks

Persistent property.

See Also

3.1.8 Log

Gets or sets whether logging is enabled. If this property is enabled, processing information will be written in log file. If this property is disabled, nothing will be written in log file.

Value	Description
True	Write to log file
False	Do not write log file

Syntax

Visual Basic 6.0
Log As Boolean

Visual C++ 6.0
BOOL GetLog() void SetLog(BOOL)

Example

Visual Basic 6.0
.Log = True

Visual C++ 6.0
m_ctrl.SetLog(true);

Remarks

Persistent property.

See Also

3.1.9 LogBakCount

Gets or sets the number of back-up file for logging. If actual file size of log file exceeded LogSize property, SavoySecsISocket control will rename the file name and make a new empty log file. If number of back-up file reached to the value of LogBakCount property, SavoySecsISocket control will delete oldest back-up file.

Syntax

Visual Basic 6.0

```
LogBakCount As Integer
```

Visual C++ 6.0

```
short GetLogBakCount()  
void SetLogBakCount(short)
```

Example

Visual Basic 6.0

```
.LogBakCount = 10
```

Visual C++ 6.0

```
m_ctrl.SetLogBakCount(10);
```

Remarks

Persistent property.

See Also

3.1.10 LogFileName

Get or sets the log file name. Log file will be created in current directory.

Syntax

Visual Basic 6.0

```
LogFileName As String
```

Visual C++ 6.0

```
CString GetLogFileName()  
void SetLogFileName(LPCTSTR)
```

Example

Visual Basic 6.0

```
.LogFileName = "./Savoy"
```

Visual C++ 6.0

```
m_ctrl.SetLogFileName("./Savoy");
```

Remarks

Persistent property.

See Also

3.1.11 LogSize

Gets or sets the log file size in kilobyte. If actual file size of log file exceeded LogSize property, SavoySecsISocket control will rename the file name and make a new empty log file. If number of back-up file reached to the value of LogBakCount property, SavoySecsISocket control will delete oldest back-up file.

Syntax

Visual Basic 6.0

```
LogSize As Long
```

Visual C++ 6.0

```
long GetLogSize()  
void SetLogSize(long)
```

Example

Visual Basic 6.0

```
.LogSize = 1024
```

Visual C++ 6.0

```
m_ctrl.SetLogSize(1024);
```

Remarks

Persistent property.

See Also

3.1.12 LogVerbose

Get or sets whether logging for detailed part is enabled.

Value	Description
True	Verbose log is enabled.
False	Verbose log is disabled.

Syntax

Visual Basic 6.0
LogVerbose As Boolean

Visual C++ 6.0
BOOL GetLogVerbose() void SetLogVerbose(BOOL)

Example

Visual Basic 6.0
.LogVerbose = False

Visual C++ 6.0
m_ctrl.SetLogVerbose(false);

Remarks

Persistent property.

See Also

3.1.13 Master

Gets or sets whether SavoySecsISocket control acts as master or slave. SEMI E4 defines that equipment side should be master and host side should be slave.

Value	Description
False	Slave
True	Master

Syntax

Visual Basic 6.0
Master As Boolean

Visual C++ 6.0
BOOL GetMaster() void SetMaster(BOOL)

Example

Visual Basic 6.0
.Master = True ' Equipment .Master = False ' Host

Visual C++ 6.0
m_ctrl.SetMaster(true); ' Equipment m_ctrl.SetMaster(false); ' Host

Remarks

Persistent property.

See Also

3.1.14 MSEC

Gets or sets whether Mitsubishi SECS protocol is enabled.

Value	Description
False	SECS-I
True	MSEC

Syntax

Visual Basic 6.0
MSEC As Boolean

Visual C++ 6.0
BOOL GetMsec() void SetMsec(BOOL)

Example

Visual Basic 6.0
.MSEC = True

Visual C++ 6.0
m_ctrl.SetMsec(true);

Remarks

Persistent property.

See Also

3.1.15 MyPortNumber

Gets or sets local portnumber for TCP/IP connection. If SavoySecsISocket is running as client, it is highly recommend that this property should be 0. Or connection will not be re-established until TCP/IP level time-out.

When SavoySecsISocket is running as server, MyPortNumber property indicates server port number for incoming client connection.

Since some port numbers are reserved by Windows OS, the number should be greater than 1024 in general. For example http server uses port number 80.

Syntax

Visual Basic 6.0

```
MyPortNumber As Long
```

Visual C++ 6.0

```
long GetMyPortNumber()  
void SetMyPortNumber(long)
```

Example

Visual Basic 6.0

```
.MyPortNumber = 0 ' Auto determine
```

Visual C++ 6.0

```
m_ctrl.SetMyPortNumber(0); // Auto determine
```

Remarks

Persistent property.

See Also

3.1.16 OneOnOne

Gets or sets whether connection is limited to 1 on 1 connection only. If this property is True and there is an existing connection, SavoySecsISocket will keep existing connection and the new connection will not be established (rejected). If this property is False, SavoySecsISocket will accept more than one connection.

Value	Description
True	Only 1 connection is acceptable
False	More than 1 connection is acceptable

Syntax

Visual Basic 6.0
OneOnOne As Boolean

Visual C++ 6.0
BOOL GetOneOnOne() void SetOneOnOne(BOOL)

Example

Visual Basic 6.0
.OneOnOne = True

Visual C++ 6.0
m_ctrl.SetOneOnOne(true);

Remarks

Persistent property.

See Also

3.1.17 PortNumber

Gets or sets the port number for TCP/IP connection.

Since some port numbers are reserved by Windows OS, the number should be greater than 1024 in general. For example http server uses port number 80.

Syntax

Visual Basic 6.0

```
PortNumber As Long
```

Visual C++ 6.0

```
long GetPortNumber()  
void SetPortNumber(long)
```

Example

Visual Basic 6.0

```
.PortNumber = 5000
```

Visual C++ 6.0

```
m_ctrl.SetPortNumber(5000);
```

Remarks

Persistent property.

See Also

3.1.18 Retry

Gets or sets the number of attempt in case of send error.

Syntax

Visual Basic 6.0

```
Retry As Integer
```

Visual C++ 6.0

```
short GetRetry()  
void SetRetry(short)
```

Example

Visual Basic 6.0

```
.Retry = 3  
.Connect = True
```

Visual C++ 6.0

```
m_ctrl.SetRetry(3);  
m_ctrl.SetConnect(true);
```

Remarks

Persistent property.

See Also

3.1.19 Server

Gets or sets the entity type. If Server property is true, SavoySecsISocket control will run as server. If Server property is false, SavoySecsISocket control will run as client.

Value	Description
False	Active entity (client)
True	Passive entity (server)

Syntax

Visual Basic 6.0
Server As Boolean

Visual C++ 6.0
BOOL GetServer() void SetServer(BOOL)

Example

Visual Basic 6.0
.Server = True ' Server .Server = False ' Client

Visual C++ 6.0
m_ctrl.SetServer(true); // Server m_ctrl.SetServer(false); // Client

Remarks

Persistent property.

See Also

3.1.20 T1

Gets or sets the T1 time out in milliseconds. Default value is 0.5 seconds.

Syntax

Visual Basic 6.0

```
T1 As Long
```

Visual C++ 6.0

```
long GetT1()  
void SetT1(long)
```

Example

Visual Basic 6.0

```
Dim IT1 As Long  
IT1 = .T1
```

Visual C++ 6.0

```
long IT1 = m_ctrl.GetT1();
```

Remarks

Persistent property.

See Also

3.1.21 T2

Gets or sets the T2 time out in milliseconds. Default value is 10 seconds.

Syntax

Visual Basic 6.0

```
T2 As Long
```

Visual C++ 6.0

```
long GetT2()  
void SetT2(long)
```

Example

Visual Basic 6.0

```
Dim IT2 As Long  
IT2 = .T2
```

Visual C++ 6.0

```
long IT2 = m_ctrl.GetT2();
```

Remarks

Persistent property.

See Also

3.1.22 T3

Gets or sets the T3 time out in milliseconds. Default value is 45 seconds.

Syntax

Visual Basic 6.0

```
T3 As Long
```

Visual C++ 6.0

```
long GetT3()  
void SetT3(long)
```

Example

Visual Basic 6.0

```
Dim IT3 As Long  
IT3 = .T3
```

Visual C++ 6.0

```
long IT3 = m_ctrl.GetT3();
```

Remarks

Persistent property.

See Also

3.1.23 T4

Gets or sets the T4 time out in milliseconds. Default value is 45 seconds.

Syntax

Visual Basic 6.0

```
T4 As Long
```

Visual C++ 6.0

```
long GetT4()  
void SetT4(long)
```

Example

Visual Basic 6.0

```
Dim IT4 As Long  
IT4 = .T4
```

Visual C++ 6.0

```
long IT4 = m_ctrl.GetT4();
```

Remarks

Persistent property.

See Also

3.2 Methods

3.2.1 AboutBox

Opens version information dialog box on the screen.

Syntax

Visual Basic 6.0

```
Sub AboutBox()
```

Visual C++ 6.0

```
void AboutBox()
```

Return Value

None.

Example

Visual Basic 6.0

```
.AboutBox
```

Visual C++ 6.0

```
m_ctrl.AboutBox();
```

Remarks

See Also

3.2.2 Disconnect

Disconnects specified TCP/IP connection.

Syntax

```
Visual Basic 6.0  
Function Disconnect(IpszIPAddress As String, IPortNumber As Long) As Boolean
```

```
Visual C++ 6.0  
BOOL Disconnect(LPCTSTR IpszIPAddress, long IPortNumber)
```

Argument	Description
IpszIPAddress	IP address to be disconnected
IPortNumber	Port number to be disconnected

Return Value

Return true if disconnection was successfully done. Otherwise return false. If false was returned, IP address or port number might be incorrect. Or connection had already disconnected.

Example

```
Visual Basic 6.0  
.Disconnect "", 5000
```

```
Visual C++ 6.0  
m_ctrl.Disconnect("",5000);
```

Remarks

See Also

3.2.3 LoadIniFile

Loads settings from INI file and initialize properties. If loading was failed, values in persistent resource will be set.

LoadIniFile method probably is called at the beginning of application, since it retrieves saved parameters by Setup method.

Syntax

Visual Basic 6.0

```
Function LoadIniFile() As Boolean
```

Visual C++ 6.0

```
BOOL LoadIniFile()
```

Return Value

Return true if loading was successful. Otherwise return false. If false was returned, IniFileName property or IniSection property might be incorrect.

Example

Visual Basic 6.0

```
Private Sub Form_Load()  
    SavoySecsISocket1.LoadIni  
End Sub
```

Visual C++ 6.0

```
void CxxxView::OnInitialUpdate()  
{  
    ...  
    m_ctrl.LoadIni();  
    ...  
}
```

Remarks

See Also

3.2.4 Send

Send specified SECS-II message.

Syntax

Visual Basic 6.0
Function Send(IpszMessage As String) As Boolean

Visual C++ 6.0
BOOL Send(LPCTSTR IpszMessage)

Argument	Description
IpszMessage	Message to be sent

Return Value

Return true if message was successfully queued. Otherwise return false.

Example

Visual Basic 6.0
SavoySecsISocket1.Send SavoySecsII1.Msg

Visual C++ 6.0
m_ctrl.Send(m_msg.GetMsg());

Remarks

SavoySecsISocket control attempts to send message. Actual result of transmission will be reported by Sent event.

See Also

3.2.5 Setup

Opens setup dialog box on the screen. If user modified parameter and press OK button, data will be written in INI file.

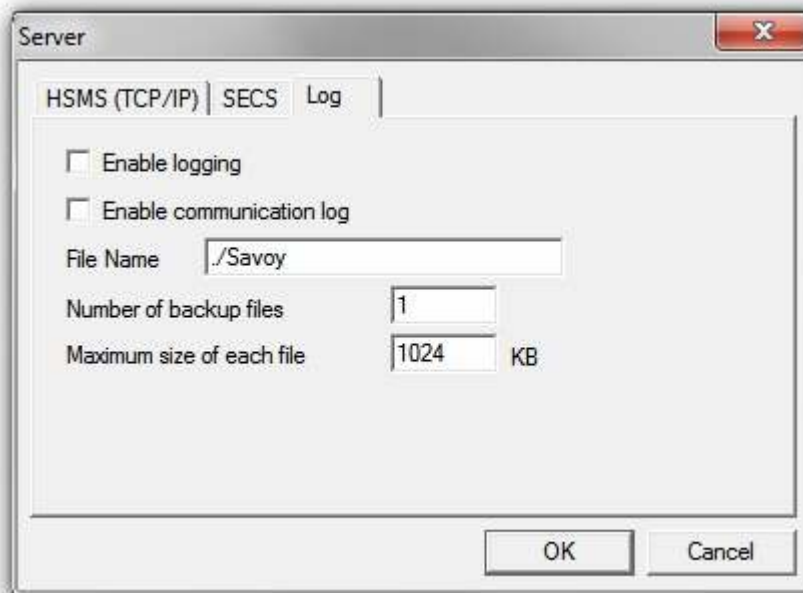
HSMS (TCP/IP) Tab

Item	Description
Passive Entity (server)	True if server. False if client.
IP Address or Computer Name	IP address or computer name of another computer to connect.
Port Number	Port number of another computer to connect.
My Port Number	Local port number. It is highly recommended to specify 0 for client.
Device ID (Decimal)	Device ID (session ID)
Limit to one-on-one connection	Only 1 connection is allowed.

SECS Tab

Item	Description
Retry	Number of retry count
Master	Master (equipment) or slave (host)
Device ID	Device ID
MSEC	Mitsubishi special SECS
T1	T1 timer
T2	T2 timer
T3	T3 timer
T4	T4 timer

Log Tab



Item	Description
Enable logging	True if logging is enabled.
Enable communication log	True if communication logging is enabled.
File name	Log file name.
Number of backup files	Number of backup files.
Maximum size of each file	Size of log file.

Syntax

Visual Basic 6.0
Function Setup(IpszCaption As String) As Boolean

Visual C++ 6.0
BOOL Setup(LPCTSTR IpszCaption)

Argument	Description
IpszCaption	Caption title of dialog box. If this value is NULL or "" (empty) string, the string of IniSection property will be used for caption title.

Return Value

If user pressed OK button and parameters were saved in INI file successfully, Setup method returns true. If user pressed Cancel button or parameter saving was failed, Setup method returns false.

Example

Visual Basic 6.0

```
.Setup "SECS-I over LAN Configuration"
```

Visual C++ 6.0

```
m_ctrl.Setup("SECS-I over LAN Configuration");
```

Remarks**See Also**

3.3 Events

3.3.1 Connected

Notifies that TCP/IP connection has been established.

If passive entity, there is no connection established until active entity will connect.

Syntax

Visual Basic 6.0

```
Event Connected(IpszIPAddress As String, IPortNumber As Long)
```

Visual C++ 6.0

```
void OnConnected(LPCTSTR IpszIPAddress, long IPortNumber)
```

Argument	Description
IpszIPAddress	IP address
IPortNumber	Port number

Example

Visual Basic 6.0

```
Text1.Text = "Connected - " + IpszIPAddress + " [" + Format$(IPortNumber) + "]"
```

Visual C++ 6.0

```
TRACE("Connected - %s [%d]",IpszIPAddress,IPortNumber);
```

Remarks

See Also

3.3.2 Disconnected

Notifies that TCP/IP connection has been disconnected.

Syntax

Visual Basic 6.0

```
Event Disconnected(lpszIPAddress As String, IPortNumber As Long)
```

Visual C++ 6.0

```
void OnDisconnected(LPCTSTR lpszIPAddress, long IPortNumber)
```

Argument	Description
lpszIPAddress	IP address
IPortNumber	Port number

Example

Visual Basic 6.0

```
Text1.Text = "Disconnected - " + lpszIPAddress + " [" + Format$(IPortNumber) + "]"
```

Visual C++ 6.0

```
TRACE("Disconnected - %s [%d]",lpszIPAddress,IPortNumber);
```

Remarks

See Also

3.3.3 Problem

Notifies that error has occurred.

Syntax

Visual Basic 6.0
Event Problem(IpszIPAddress As String, IPortNumber As Long, sErrorCode As Integer, IpszAdditionalInfo As String)

Visual C++ 6.0
void OnProblem(LPCTSTR IpszIPAddress, long IPortNumber, short sErrorCode, LPCTSTR IpszAdditionalInfo)

Argument	Description
IpszIPAddress	IP address
IPortNumber	Port number
sErrorCode	Error code (see below)
IpszAdditionalInfo	Additional information (not in use)

Example

Visual Basic 6.0
Text1.Text = "Error - " + IpszIPAddress + " [" + Format\$(IPortNumber) + "], Code : " + Format\$(sErrorCode)

Visual C++ 6.0
TRACE("Error - %s [%d], Code : ",IpszIPAddress,IPortNumber,sErrorCode);

Remarks

Error from SavoySecsISocket

Error code	Enumeration	Description
1	ErrorSecsUnexpectedChar	Received unexpected character.
2	ErrorSecsBadLength	Length byte is not correct.
3	ErrorSecsBadSum	Check sum error.
4	ErrorSecsBadParity	Parity error.
5	ErrorSecsBadMultiBlockNumber	Block number in multi-block message is not correct.
6	ErrorSecsBadBlockNumber	Block number is neither 0 nor 1.
7	ErrorSecsNak	Received NAK.
8	ErrorSecsRetry	Retry over.
9	ErrorSecsT4	T4 time out.
10	ErrorSecsT2	T2 time out.
11	ErrorSecsT1	T1 time out.
12	ErrorSecsT3	T3 time out.
13	ErrorSecsBadDeviceID	Device ID mismatched.

Error from WinSock

Error code	Description
10093	Socket has not been initialized.
10050	Network subsystem error.
10048	Socket local address is in use.
10014	Invalid user address (like invalid character).
10036	Service provider is in progress.

10049	Remote address can not be available.
10047	Cannot use specified address family for this socket.
10061	Connection has been refused.
10039	?
10022	Invalid listening socket.
10056	Already connected.
10024	?
10051	Cannot reach to network.
10055	Buffer size is not enough.
10038	Not a socket.
10060	Time out before established connection.
10035	Cannot execute right now.

See Also

3.3.4 Received

Notifies that SavoySecsISocket control received message.

Syntax

Visual Basic 6.0

```
Event Received(IpszIPAddress As String, IPortNumber As Long, IpszMsg As String)
```

Visual C++ 6.0

```
void OnReceived(LPCTSTR IpszIPAddress, long IPortNumber, LPCTSTR IpszMsg)
```

Argument	Description
IpszIPAddress	IP address
IPortNumber	Port number
IpszMsg	Received message

Example

Visual Basic 6.0

```
With SavoySecsII1
  .Msg = IpszMsg
  If .Stream = 1 AND .Function = 13 Then
    ' S1F13
    ...
  End If
End With
```

Visual C++ 6.0

```
m_msg.SetMsg(IpszMsg);
if(m_msg.GetStream()==1 && m_msg.GetFunction()==13)
{
  // S1F13
  ...
}
```

Remarks

See Also

3.3.5 Sent

Notifies that SECS-II message has been sent.

Syntax

Visual Basic 6.0

```
Event Sent(lpszIPAddress As String, IPortNumber As Long, lpszMsg As String)
```

Visual C++ 6.0

```
void OnSent(LPCTSTR lpszIPAddress, long IPortNumber, LPCTSTR lpszMsg)
```

Argument	Description
lpszIPAddress	IP address
IPortNumber	Port number
lpszMsg	Sent message

Example

Visual Basic 6.0

```
SavoySecsII1.Msg = lpszMsg
```

Visual C++ 6.0

```
m_msg.SetMsg(lpszMsg);
```

Remarks

See Also