

SavoySocket ActiveX Control
User Guide

1 Revision History

Version	Date	Name	Description
1.00	Jul, 31 st , 2009	Hikaru Okada	Created as new document
1.00a	Aug, 22 nd , 2009	Hikaru Okada	Splitted into separate document, since number of pages became large.

2 Table of Contents

1	Revision History	2
2	Table of Contents	3
3	SavoySocket	4
3.1	Properties	5
3.1.1	Appearance	5
3.1.2	BorderStyle	6
3.1.3	Connect	7
3.1.4	IniFileName	8
3.1.5	IniSection	9
3.1.6	IPAddress	10
3.1.7	MyPortNumber	11
3.1.8	PortNumber	12
3.1.9	Server	13
3.2	Methods	14
3.2.1	AboutBox	14
3.2.2	Disconnect	15
3.2.3	LoadIniFile	16
3.2.4	Send	17
3.2.5	Setup	18
3.3	Events	20
3.3.1	Connected	20
3.3.2	Disconnected	21
3.3.3	Problem	22
3.3.4	Received	23

3 SavoySocket

SavoySocket control is an assistant product to develop TCP/IP (socket) communication application software. Developing client side software generally is not hard, but server side is difficult. User can avoid whether current setting is client or server to work with SavoySocket control. Of course, multiple client connection is supported.

Properties

Name	Description
Appearance	Gets or sets the value that determines the appearance of a SavoySocket control.
BorderStyle	Gets or sets whether the SavoySocket control has a border.
Connect	Gets or sets the TCP/IP connection status.
IniFileName	Gets or sets INI file name to read/write settings.
IniSection	Gets or sets section name in INI file to read/write settings.
IPAddress	Gets or sets the IP address of passive entity computer for TCP/IP connection.
MyPortNumber	Gets or sets local portnumber for TCP/IP connection.
PortNumber	Gets or sets the port number for TCP/IP connection.
Server	Gets or sets the role of SavoySocket control.

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 data.
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 SavoySocket control received data.

3.1 Properties

3.1.1 Appearance

Gets or sets the value that determines the appearance of a SavoySocket 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 SavoySocket 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), SavoySocket 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), SavoySocket 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. SavoySocket control will take care disconnection automatically. When Server property was set to True and Connected 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

```
BOOL GetConnect()  
void SetConnect(BOOL)
```

Example

Visual Basic 6.0

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

Visual C++ 6.0

```
m_ctrl.SetIPAddress("my_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!");
```

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 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

3.1.5 IniSection

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

Default setting is "SavoySocket".

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

3.1.6 IPAddress

Gets or sets the IP address of passive entity 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 = "my_server"  
.PortNumber = 5000  
.MyPortNumber = 0  
.Server = False  
.Connect = True ' Open  
If Not .Connect then  
    MsgBox "Cannot connect to server!"  
End If
```

Visual C++ 6.0

```
m_ctrl.SetIPAddress("my_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!");
```

Remarks

Persistent property.

See Also

3.1.7 MyPortNumber

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

When SavoySocket 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

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

Visual C++ 6.0

```
m_ctrl.SetIPAddress("my_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!");
```

Remarks

Persistent property.

See Also

3.1.8 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

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

Visual C++ 6.0

```
m_ctrl.SetIPAddress("my_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!");
```

Remarks

Persistent property.

See Also

3.1.9 Server

Gets or sets the role of SavoySocket control. If Server property is true, SavoySocket control will run as server. If Server property is false, SavoySocket 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
<pre> .IPAddress = "my_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("my_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

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()  
    SwingSocket1.LoadIni  
End Sub
```

Visual C++ 6.0

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

Remarks

See Also

3.2.4 Send

Send specified data.

Syntax

Visual Basic 6.0

```
Function Send(IpszHexData As String) As Boolean
```

Visual C++ 6.0

```
BOOL Send(LPCTSTR IpszHexData)
```

Argument	Description
IpszMessage	Hexadecimal string expression of data to be sent

Return Value

Return true if transmission was successful. Otherwise return false.

Example

Visual Basic 6.0

```
.Send "00112233445566778899aabbccddeeff"
```

Visual C++ 6.0

```
m_ctrl.Send("00112233445566778899aabbccddeeff");
```

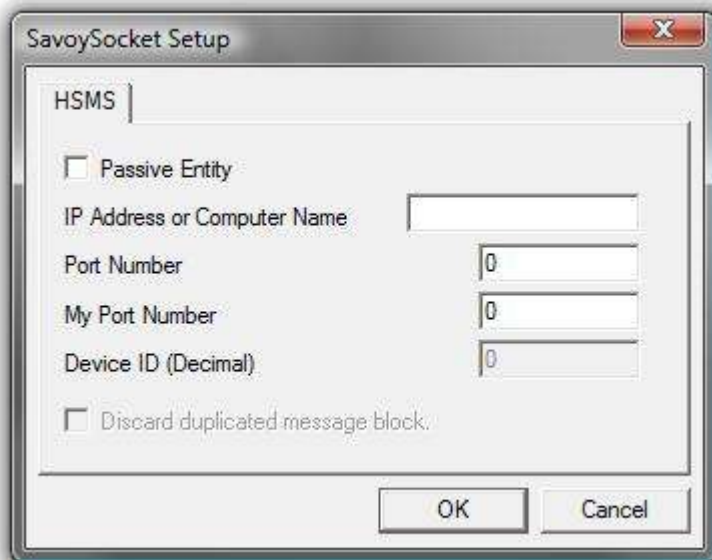
Remarks

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 Tab



Item	Description
Passive Entity	True if passive entity. False if active entity.
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 active entity.

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 "Socket Configuration"
```

```
Visual C++ 6.0
```

```
m_ctrl.Setup("Socket Configuration");
```

Remarks

See Also

3.3 Events

3.3.1 Connected

Notifies that TCP/IP connection has been established.

If server, there is no connection established until client 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 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 SavoySocket control received data.

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	Hexadecimal string expression of received data

Example

Visual Basic 6.0

```
Text1.Text = "Received - " + IpszMsg
```

Visual C++ 6.0

```
TRACE("Received - %s", IpszMsg);
```

Remarks

See Also