

SavoySocket ActiveX コントロール
ユーザーガイド

1 改訂履歴

バージョン	日付	氏名	説明
1.00	2009年7月31日	Hikaru Okada	新規に作成。
1.00a	2009年8月22日	Hikaru Okada	マニュアルのページ数が大きくなったため分割。

2 目次

1	改訂履歴.....	2
2	目次.....	3
3	SavoySocket.....	4
3.1	プロパティ.....	5
3.1.1	Appearance.....	5
3.1.2	BorderStyle.....	6
3.1.3	Connect.....	7
3.1.4	IniFileName.....	9
3.1.5	IniSection.....	10
3.1.6	IPAddress.....	11
3.1.7	MyPortNumber.....	12
3.1.8	PortNumber.....	13
3.1.9	Server.....	14
3.2	メソッド.....	15
3.2.1	AboutBox.....	15
3.2.2	Disconnect.....	16
3.2.3	LoadIniFile.....	17
3.2.4	Send.....	18
3.2.5	Setup.....	19
3.3	イベント.....	21
3.3.1	Connected.....	21
3.3.2	Disconnected.....	22
3.3.3	Problem.....	23
3.3.4	Received.....	25

3 SavoySocket

SavoySocket コントロールは TCP/IP を使った通信機能を作成するための開発支援製品です。クライアント側ソフトを作成するのは比較的容易ですが、サーバ側ソフトの開発は少々面倒です。SavoySocket コントロールはクライアント側、サーバ側を意識することなく通信ソフトを作成することができます。もちろん多対一の通信にも対応しています。

プロパティ

名前	説明
Appearance	外観を決定する値を取得または設定します。
BorderStyle	境界線スタイルを取得または設定します。
Connect	TCP/IP の接続状態を取得または設定します。
IniFileName	設定を読み書きするための INI ファイル名を取得または設定します。
IniSection	設定を読み書きするための INI ファイル中のセクション名を取得または設定します。
IPAddress	TCP/IP 接続のサーバのコンピュータの IP アドレスを取得または設定します。
MyPortNumber	TCP/IP 接続のローカルポート番号を取得または設定します。
PortNumber	TCP/IP のポート番号を取得または設定します。
Server	役割を取得または設定します。

メソッド

名前	説明
AboutBox	バージョン情報を表示します。
Disconnect	指定された TCP/IP 接続を切断します。
LoadIniFile	設定内容を INI ファイルから読み出します。
Send	指定されたデータを送信します。
Setup	セットアップ画面を表示します。

イベント

名前	説明
Connected	TCP/IP 接続が成立したときに通知されます。
Disconnected	TCP/IP 接続が切断したときに通知されます。
Problem	エラーが発生したときに通知されます。
Received	TCP/IP 接続経由でデータを受信したときに通知されます。

3.1 プロパティ

3.1.1 Appearance

SavoySocket コントロールの外観を決定する値を取得または設定します。

値	説明
0	フラット
1	凹んだ枠線

構文

Visual Basic 6.0
Appearance As Integer

Visual C++ 6.0
short GetAppearance() void SetAppearance(short)

使用例

Visual Basic 6.0
.Appearance = 0 ' flat .Appearance = 1 ' sunken

Visual C++ 6.0
m_ctrl.SetAppearance(0); // flat m_ctrl.SetAppearance(1); // sunken

特記事項

永続化プロパティ。

参照

3.1.2 BorderStyle

SavoySocket コントロールの境界線スタイルを取得または設定します。

値	説明
0	境界線なし
1	境界線あり

構文

Visual Basic 6.0
BorderStyle As Integer

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

使用例

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

特記事項

永続化プロパティ。

参照

3.1.3 Connect

TCP/IP の接続状態を取得または設定します。もし Server プロパティが false (クライアント) の時に Connect プロパティが True にセットされた場合、SavoySocket コントロールは接続を確立しようと試みます。もし接続が成立しなかった場合、Connect プロパティは False にセットされます。

もし Server プロパティが True (サーバ) の時に Connect プロパティが True にセットされた場合、SavoySocket コントロールはサーバポートをオープンし、接続を受け入れるために待ち状態になります。この時点ではまだコネクションは成立しておらず、クライアントが接続してきたときに初めて成立します。このようにコネクションが成立していなくてもポートがオープンできてクライアントの接続待ち状態になれば Connect プロパティの値は True になります。

False をセットするとコネクションを解除します。アプリケーションを終了するときにコネクションを解除し忘れても心配ありません。コントロールが自動的に解除してくれるからです。Server プロパティが True (サーバ) の時に Connect プロパティに False をセットすると接続していた全てのクライアントとのコネクションが解消されます。

構文

Visual Basic 6.0
Connect As Boolean
Visual C++ 6.0
BOOL GetConnect() void SetConnect(BOOL)

使用例

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>

特記事項

接続相手の Connect プロパティが False になると接続は切断されます。このためクライアント側なら Connect プロパティが自動的に False になるので注意が必要です。

参照

3.1.4 IniFileName

設定を読み書きするための INI ファイル名を取得または設定します。もし INI ファイル名がフルパス名かフォルダの相対参照を含む場合は、INI ファイルは指定された場所に作成されます。そうでない場合は、Windows のシステムフォルダに INI ファイルが作成されます。この理由からフォルダ名と共に使用するのが推奨されます。もしカレントディレクトリが INI ファイルの場所であれば、「./」を先頭に付けます。

「/」または「¥」をフォルダ名の区切り文字として使用できます。

構文

Visual Basic 6.0
<code>IniFileName As String</code>

Visual C++ 6.0
<code>CString GetIniFileName() void SetIniFileName(LPCTSTR)</code>

使用例

Visual Basic 6.0
<code>.IniFileName = "./Savoy.ini"</code>

Visual C++ 6.0
<code>m_ctrl.SetIniFileName("./Savoy.ini");</code>

特記事項

永続化プロパティ。

参照

3.1.5 IniSection

設定を読み書きするための INI ファイル中のセクション名を取得または設定します。

構文

Visual Basic 6.0

```
IniSection As String
```

Visual C++ 6.0

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

使用例

Visual Basic 6.0

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

Visual C++ 6.0

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

特記事項

永続化プロパティ。

参照

3.1.6 IPAddress

TCP/IP 接続のサーバのコンピュータの IP アドレスを取得または設定します。Server プロパティが true のときは、接続を待ち受けていて IP アドレスは不要なので、このプロパティは無視されます。

ローカルコンピュータ(自分自身のコンピュータ)に接続する場合は 127.0.0.1 を使用してください。

IP アドレスの代わりにコンピュータ名を使用することもできます。

構文

Visual Basic 6.0

```
IPAddress As String
```

Visual C++ 6.0

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

使用例

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!");
```

特記事項

永続化プロパティ。

参照

3.1.7 MyPortNumber

TCP/IP 接続のローカルポート番号を取得または設定します。もし SavoySocket コントロールがアクティブエンティティとして動作している場合、このプロパティは 0 であるべきです。さもなければ接続は TCP/IP レベルでのタイムアウトが発生するまで再接続ができなくなります。

構文

Visual Basic 6.0

```
MyPortNumber As Long
```

Visual C++ 6.0

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

使用例

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!");
```

特記事項

永続化プロパティ。

参照

3.1.8 PortNumber

TCP/IP のポート番号を取得または設定します。

構文

Visual Basic 6.0

```
PortNumber As Long
```

Visual C++ 6.0

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

使用例

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!");
```

特記事項

永続化プロパティ。

参照

3.1.9 Server

役割を取得または設定します。もしこのプロパティが true の場合、サーバを意味します。もしこのプロパティが false の場合、クライアントを意味します。

値	説明
False	クライアント
True	サーバ

構文

Visual Basic 6.0

```
Server As Boolean
```

Visual C++ 6.0

```
BOOL GetServer()
void SetServer(BOOL)
```

使用例

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!");
```

特記事項

永続化プロパティ。

参照

3.2 メソッド

3.2.1 AboutBox

バージョン情報を表示します。

構文

Visual Basic 6.0
Sub AboutBox()

Visual C++ 6.0
void AboutBox()

戻り値

ありません。

使用例

Visual Basic 6.0
.AboutBox

Visual C++ 6.0
m_ctrl.AboutBox();

特記事項

参照

3.2.2 Disconnect

指定された TCP/IP 接続を切断します。

構文

Visual Basic 6.0

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

Visual C++ 6.0

```
BOOL Disconnect(LPCTSTR IpszIPAddress, long IPortNumber)
```

引数	説明
IpszIPAddress	切断相手の IP アドレス
IPortNumber	切断相手のポート番号

戻り値

切断に成功した場合は True を、失敗した場合は False を返します。失敗する可能性としては、指定された IP アドレスとポート番号が間違っているか、既に切断されている場合が考えられます。

使用例

Visual Basic 6.0

```
.Disconnect "", 5000
```

Visual C++ 6.0

```
m_ctrl.Disconnect("",5000);
```

特記事項

参照

3.2.3 LoadIniFile

設定内容を INI ファイルから読み出し、プロパティを初期化します。読み込みができない場合はリソースに埋め込まれている値がセットされます。

このメソッドは Setup メソッドで保存されたプロパティを取り出すためアプリケーションの起動時に呼び出すといいでしょう。

構文

```
Visual Basic 6.0  
Function LoadIniFile() As Boolean
```

```
Visual C++ 6.0  
BOOL LoadIniFile()
```

戻り値

正しく初期化できた場合は True が、失敗した場合は False が返ります。失敗するケースとしては、IniFileName プロパティと IniSection プロパティが正しくない場合が考えられます。

使用例

```
Visual Basic 6.0  
Private Sub Form_Load()  
    SwingSocket1.LoadIni  
End Sub
```

```
Visual C++ 6.0  
void CxxxView::OnInitialUpdate()  
{  
    ...  
    m_ctrl.LoadIni();  
    ...  
}
```

特記事項

参照

3.2.4 Send

指定されたデータを送信します。

構文

Visual Basic 6.0

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

Visual C++ 6.0

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

引数	説明
IpszMessage	送信するデータの 16 進法文字列

戻り値

正しく送信できた場合は True が、失敗した場合は False が返ります。

使用例

Visual Basic 6.0

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

Visual C++ 6.0

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

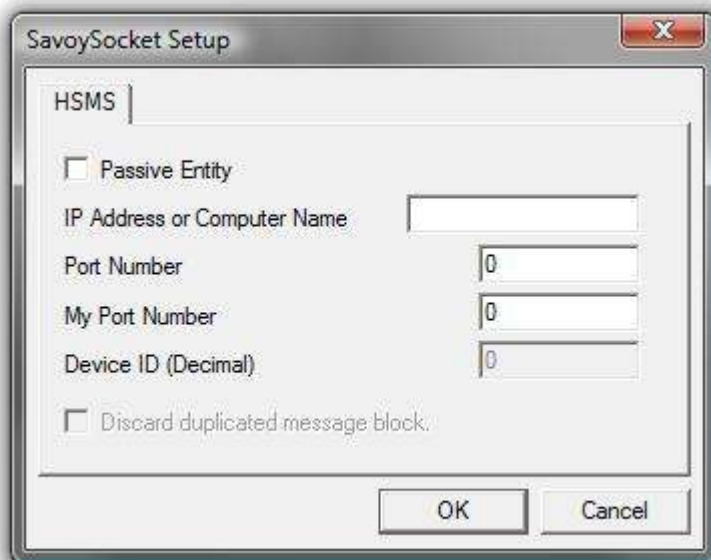
特記事項

参照

3.2.5 Setup

セットアップ画面を表示します。パラメータを変更して OK ボタンを押すと INI ファイルに書き込まれます。

HSMS タブ



項目	説明
Passive Entity	サーバの場合はチェックをつけます。
IP Address or Computer Name	相手の IP アドレス、またはコンピュータ名。
Port Number	相手のポート番号。
My Port Number	自分のポート番号。アクティブエンティティの場合は 0 を推奨します。

構文

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

```
Visual C++ 6.0
BOOL Setup(LPCTSTR IpszCaption)
```

戻り値

ユーザが OK ボタンを押し、正しく INI ファイルに保存されると True を返します。ユーザが Cancel ボタンを押したり、INI ファイルの保存に失敗すると False を返します。

使用例

```
Visual Basic 6.0
.Setup "Socket Configuration"
```

```
Visual C++ 6.0
m_ctrl.Setup("Socket Configuration");
```

特記事項

参照

3.3 イベント

3.3.1 Connected

TCP/IP 接続が成立したときに通知されます。

サーバ側の場合、Connect プロパティを True にセットしただけではコネクションは成立していません。クライアント側が接続してきてはじめてコネクションが成立したことになります。

構文

Visual Basic 6.0

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

Visual C++ 6.0

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

引数	説明
IpszIPAddress	接続相手のIPアドレス
IPortNumber	接続相手のポート番号

使用例

Visual Basic 6.0

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

Visual C++ 6.0

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

特記事項

参照

3.3.2 Disconnected

TCP/IP 接続が切断したときに通知されます。

構文

Visual Basic 6.0

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

Visual C++ 6.0

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

引数	説明
IpszIPAddress	切断されたIPアドレス
IPortNumber	切断されたポート番号

使用例

Visual Basic 6.0

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

Visual C++ 6.0

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

特記事項

参照

3.3.3 Problem

エラーが発生したときに通知されます。

構文

```
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
```

引数	説明
IpszIPAddress	エラーが発生したIPアドレス。
IPortNumber	エラーが発生したポート番号。
sErrorCode	エラーコード。エラーコードは以下のいずれかの値です。
IpszAdditionalInfo	追加情報。現在のところは未使用となっています。

使用例

```
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);
```

特記事項

エラーコード	説明
10093	ソケットの初期化がされていない
10050	ネットワークサブシステムのエラー
10048	ソケットのローカルアドレスが既に使用中
10014	ユーザアドレスが正しくない(禁則文字など)
10036	現在サービスプロバイダが処理中
10049	リモートアドレスが正しくない
10047	指定されたアドレスファミリーはこのソケットで使用できない
10061	接続拒否された
10039	?
10022	リスニングソケット
10056	既に接続されている
10024	?
10051	ネットワークに到達できなかった
10055	バッファが足りない
10038	ソケットでない
10060	コネクションが成立する前にタイムアウトした
10035	すぐに実行できない

参照

3.3.4 Received

SavoySocket コントロールが TCP/IP 接続経由でデータを受信したときに通知されます。

構文

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)
```

引数	説明
IpszIPAddress	メッセージの発信元のIPアドレス
IPortNumber	メッセージの発信元のポート番号
IpszMsg	受信したデータの 16 進法文字列

使用例

Visual Basic 6.0

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

Visual C++ 6.0

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

特記事項

参照