

SavoyHsms 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。
1.00b	2009年8月30日	Hikaru Okada	一部の誤記を訂正。

2 目次

1	改訂履歴.....	2
2	目次.....	3
3	SavoyHsms.....	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	Log.....	12
3.1.8	LogBakCount.....	13
3.1.9	LogFileName.....	14
3.1.10	LogSize.....	15
3.1.11	LogVerbose.....	16
3.1.12	MyHostName.....	17
3.1.13	MyPortNumber.....	18
3.1.14	PortNumber.....	19
3.1.15	Selected.....	20
3.1.16	Server.....	21
3.1.17	T3.....	22
3.1.18	T5.....	23
3.1.19	T6.....	24
3.1.20	T7.....	25
3.1.21	T8.....	26
3.2	メソッド.....	27
3.2.1	AboutBox.....	27
3.2.2	Disconnect.....	28
3.2.3	LoadIniFile.....	29
3.2.4	Send.....	30
3.2.5	Setup.....	31
3.2.6	ToIPAddress.....	34
3.3	イベント.....	35
3.3.1	Connected.....	35
3.3.2	Disconnected.....	36
3.3.3	Problem.....	37
3.3.4	Received.....	39
3.3.5	SelChanged.....	40

3 SavoyHsms

SavoyHsms コントロールは SEMI E37 (HSMS)の機能を作成するための開発支援製品です。装置側ソフト、ホスト側ソフトのいずれの開発にも使えます。SavoySecsII コントロールと組み合わせて使います。

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	接続状態を取得または設定します。
IniFileName	INI ファイル名を取得または設定します。
IniSection	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	パッシブエンティティの IP アドレスを取得または設定します。
Log	ログの有効・無効を取得または設定します。
LogBakCount	ログのバックアップファイルの数を取得または設定します。
LogFileName	ログファイル名を取得または設定します。
LogSize	ログファイルのサイズを取得または設定します。
LogVerbose	詳細部分のログが有効かどうかを取得または設定します。
MyHostName	自分自身のコンピュータ名を取得します。
MyPortNumber	ローカルポート番号を取得または設定します。
PortNumber	ポート番号を取得または設定します。
Selected	セレクト状態を取得または設定します。
Server	エンティティのタイプを取得または設定します。
T3	T3 タイムアウトを取得または設定します。
T5	T5 タイムアウトを取得または設定します。
T6	T6 タイムアウトを取得または設定します。
T7	T7 タイムアウトを取得または設定します。
T8	T8 タイムアウト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 HSMS 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	指定されたメッセージを送信します。
Setup	セットアップ画面を表示します。
ToIPAddress	指定されたコンピュータ名を IP アドレスに変換します。

イベント

名前	説明
Connected	HSMS 接続が成立したときに通知されます。
Disconnected	HSMS 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	メッセージを受信したときに通知されます。
SelChanged	画面上で HSMS 接続の選択が変化したときに通知されます。

3.1 プロパティ

3.1.1 Appearance

SavoyHsms 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

SavoyGem 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

HSMS 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、SavoyHsms 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、SavoyHsms 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

```
.IPAddress = "hsms_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("hsms_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

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

```
IniFileName As String
```

Visual C++ 6.0

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

使用例

Visual Basic 6.0

```
.IniFileName = "./Savoy.ini"
```

Visual C++ 6.0

```
m_ctrl.SetIniFileName("./Savoy.ini");
```

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。1つのプロジェクトに複数の SavoyHsms コントロールが埋め込まれていても、別々のセクション名を指定することで設定内容を変えることができます。

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

HSMS 接続のパスブエンティティのコンピュータの 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
<pre>.IPAddress = "" .PortNumber = 5000 .LocalPortNumber = 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(""); 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

永続化プロパティ。

参照

3.1.7 Log

ログの有効・無効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、処理情報はログファイルに書き込まれます。もしこのプロパティがfalseの場合はログファイルには記録されません。

値	説明
True	ログファイルに記録する
False	ログファイルに記録しない

構文

Visual Basic 6.0
Log As Boolean

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

使用例

Visual Basic 6.0
.Log = True

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

特記事項

永続化プロパティ。

参照

3.1.8 LogBakCount

ログのバックアップファイルの数を取得または設定します。もしログファイルの実際のファイルサイズが LogSize プロパティを超えた場合、SavoyHsms コントロールはファイル名をリネームし、新たに空のログファイルを作成します。もしバックアップファイルの数が LogBakCount プロパティに達した場合、SavoyGem コントロールは古いバックアップファイルから消去します。

構文

Visual Basic 6.0

```
LogBakCount As Integer
```

Visual C++ 6.0

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

使用例

Visual Basic 6.0

```
.LogBakCount = 10
```

Visual C++ 6.0

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

特記事項

永続化プロパティ。

参照

3.1.9 LogFileName

ログファイル名を取得または設定します。

構文

Visual Basic 6.0

```
LogFileName As String
```

Visual C++ 6.0

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

使用例

Visual Basic 6.0

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

Visual C++ 6.0

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

特記事項

永続化プロパティ。

参照

3.1.10 LogSize

ログファイルのサイズをキロバイト単位で取得または設定します。もしログファイルの実際のファイルサイズが LogSize プロパティを超えた場合、SavoyHsms コントロールはファイル名をリネームし、新たに空のログファイルを作成します。もしバックアップファイルの数が LogBakCount プロパティに達した場合、SavoyGem コントロールは古いバックアップファイルから消去します。

構文

Visual Basic 6.0

```
LogSize As Long
```

Visual C++ 6.0

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

使用例

Visual Basic 6.0

```
.LogSize = 1024
```

Visual C++ 6.0

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

特記事項

永続化プロパティ。

参照

3.1.11 LogVerbose

詳細部分のログが有効かどうかを取得または設定します。

構文

Visual Basic 6.0

```
LogVerbose As Boolean
```

Visual C++ 6.0

```
BOOL GetLogVerbose()  
void SetLogVerbose(BOOL)
```

使用例

Visual Basic 6.0

```
.LogVerbose = False
```

Visual C++ 6.0

```
m_ctrl.SetLogVerbose(false);
```

特記事項

永続化プロパティ。

参照

3.1.12 MyHostName

自分自身のコンピュータ名を取得します。

構文

Visual Basic 6.0

```
MyHostName As String
```

Visual C++ 6.0

```
CString GetMyHostName()
```

使用例

Visual Basic 6.0

```
.IPAddress = .MyHostName
```

Visual C++ 6.0

```
m_ctrl.SetIPAddress(m_ctrl.GetMyHostName());
```

特記事項

読み出し専用プロパティ。

参照

3.1.13 MyPortNumber

HSMS 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 SavoyHsms 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。

Windows で使用しているポート番号 (http サーバなどは 80) があるため、一般的には 1024 以上の値を指定すべきです。

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

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

Visual C++ 6.0

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

特記事項

永続化プロパティ。

参照

3.1.14 PortNumber

HSMS のポート番号を取得または設定します。

Windows で使用しているポート番号 (http サーバなどは 80) があるため、一般的には 1024 以上の値を指定すべきです。

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

```
.PortNumber = 5000
```

Visual C++ 6.0

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

特記事項

永続化プロパティ。

参照

3.1.15 Selected

HSMS 通信のセレクト状態を取得または設定します。

セレクト要求を受け取ると SavoyHsms は自動的にセレクト状態に移行します。もし何らかの理由でセレクト状態にたくない場合は Selected プロパティを False にセットし直してやります。セレクト応答を受け取った場合は理由コードによってセレクト状態に移行するかどうかを判断します。このプロパティは T7 タイムアウトにも関連します。

構文

Visual Basic 6.0
Selected As Boolean

Visual C++ 6.0
BOOL GetSelected() void SetSelected(BOOL)

使用例

Visual Basic 6.0
.Selected = False

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

特記事項

参照

3.1.16 Server

エンティティのタイプ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、パッシブエンティティを意味します。もしこのプロパティが False の場合、アクティブエンティティを意味します。

値	説明
False	アクティブエンティティ(クライアント)
True	パッシブエンティティ(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
.Server = True ' Server .Server = False ' Client

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

特記事項

永続化プロパティ。

参照

3.1.17 T3

HSMS の T3 タイムアウトをミリ秒単位で取得または設定します。デフォルトは 45 秒です。

構文

Visual Basic 6.0

```
T3 As Long
```

Visual C++ 6.0

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

使用例

Visual Basic 6.0

```
.T3 = 500 ' 500 milliseconds
```

Visual C++ 6.0

```
m_ctrl.SetT3(500); // 500 milliseconds
```

特記事項

永続化プロパティ。

参照

3.1.18 T5

HSMS の T5 タイムアウトをミリ秒単位で取得または設定します。

構文

Visual Basic 6.0

```
T5 As Long
```

Visual C++ 6.0

```
long GetT5()  
void SetT5(long)
```

使用例

Visual Basic 6.0

```
.T3 = 5000 ' 5 seconds
```

Visual C++ 6.0

```
m_ctrl.SetT3(5000); // 5 seconds
```

特記事項

永続化プロパティ。

参照

3.1.19 T6

HSMS の T6 タイムアウトをミリ秒単位で取得または設定します。デフォルトは 5 秒です。

構文

Visual Basic 6.0

```
T6 As Long
```

Visual C++ 6.0

```
long GetT6()  
void SetT6(long)
```

使用例

Visual Basic 6.0

```
.T6 = 5000 ' 5 seconds
```

Visual C++ 6.0

```
m_ctrl.SetT6(5000); // 5 seconds
```

特記事項

永続化プロパティ。

参照

3.1.20 T7

HSMS の T7 タイムアウトをミリ秒単位で取得または設定します。デフォルトは 10 秒です。

構文

Visual Basic 6.0

```
T7 As Long
```

Visual C++ 6.0

```
long GetT7()  
void SetT7(long)
```

使用例

Visual Basic 6.0

```
.T7 = 10000 ' 10 seconds
```

Visual C++ 6.0

```
m_ctrl.SetT7(10000); // 10 seconds
```

特記事項

永続化プロパティ。

参照

3.1.21 T8

HSMS の T8 タイムアウトをミリ秒単位で取得または設定します。デフォルトは 0.5 秒です。

構文

Visual Basic 6.0

```
T8 As Long
```

Visual C++ 6.0

```
long GetT8()  
void SetT8(long)
```

使用例

Visual Basic 6.0

```
.T8 = 500 ' 500 milliseconds
```

Visual C++ 6.0

```
m_ctrl.SetT8(500); // 500 milliseconds
```

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 HSMS 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()  
    SwingHsms1.LoadIni  
End Sub
```

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

特記事項

参照

3.2.4 Send

指定されたメッセージを送信します。

構文

Visual Basic 6.0

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

Visual C++ 6.0

```
BOOL Send(LPCTSTR IpszMessage)
```

引数	説明
IpszMessage	送信するメッセージ

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

```
SavoyHsms1.Send SavoySecs11.Msg
```

Visual C++ 6.0

```
m_ctrl.Send(m_msg.GetMsg());
```

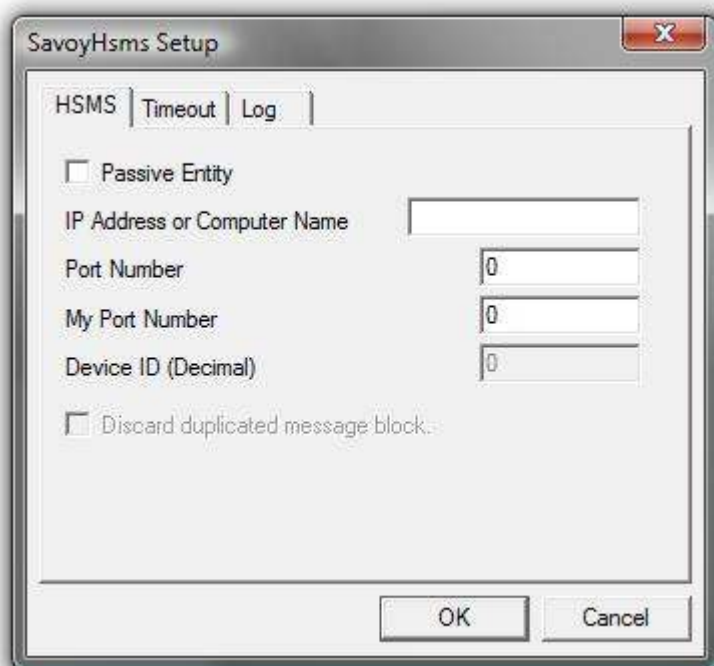
特記事項

参照

3.2.5 Setup

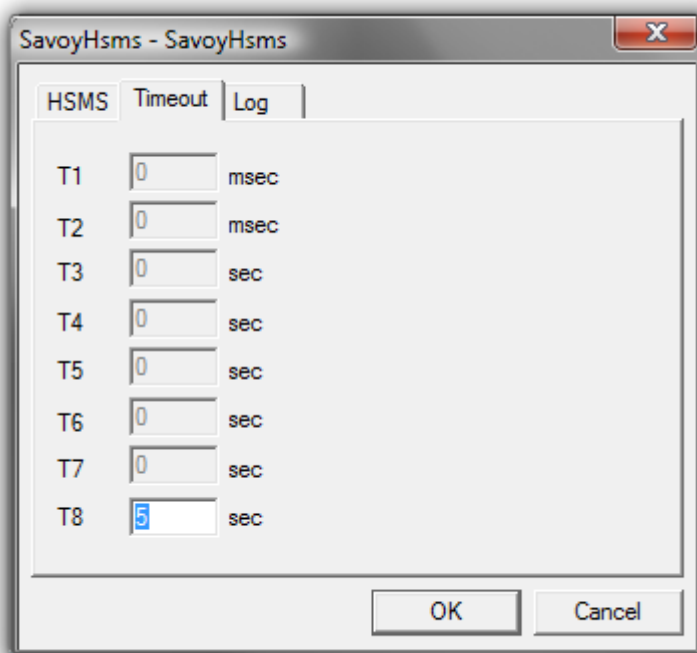
セットアップ画面を表示します。パラメータを変更して OK ボタンを押すと INI ファイルに書き込まれます。

HSMS タブ



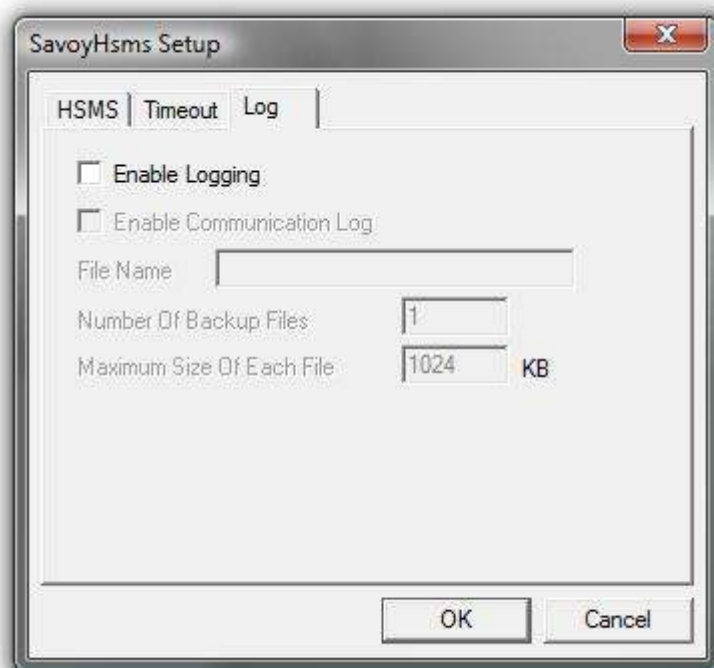
項目	説明
Passive Entity	パッシブエンティティ(サーバ)の場合はチェックをつけます。
IP Address or Computer Name	相手の IP アドレス、またはコンピュータ名。
Port Number	相手のポート番号。
My Port Number	自分のポート番号。アクティブエンティティの場合は 0 を推奨します。
Device ID (Decimal)	デバイス ID。

Timeout タブ



項目	説明
T8	T8 タイマー

Log タブ



項目	説明
Enable logging	ログに記録する場合はチェックをつけます。
Enable communication log	詳細の通信ログを記録する場合はチェックをつけます。
File name	ログファイル名。
Number of backup files	バックアップファイルの数。
Maximum size of each file	ログファイルのサイズ。

構文

Visual Basic 6.0

Function Setup(lpszCaption As String) As Boolean

Visual C++ 6.0

BOOL Setup(LPCTSTR lpszCaption)

引数	説明
lpszCaption	ダイアログボックスのキャプションタイトル。もしこの値が NULL または空文字列の場合、IniSection プロパティがキャプションタイトルとして使われます。

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、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 "HSMS Configuration"

Visual C++ 6.0

m_ctrl.Setup("HSMS Configuration");

特記事項**参照**

3.2.6 ToIPAddress

指定されたコンピュータ名を IP アドレスに変換します。

構文

Visual Basic 6.0

```
Function ToIPAddress(lpszHostName As String) As String
```

Visual C++ 6.0

```
CString ToIPAddress(LPCTSTR lpszHostName)
```

引数	説明
lpszHostName	変換したいホスト名。

戻り値

正常に変換されると IP アドレスを文字列として返します。失敗した場合は、空の文字列を返します。

使用例

Visual Basic 6.0

```
.IPAddress = .ToIPAddress("")
```

Visual C++ 6.0

```
m_ctrl.SetIPAddress(m_ctrl.ToIPAddress(""));
```

特記事項

参照

3.3 イベント

3.3.1 Connected

HSMS 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

HSMS 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

SavoyHsms の出すエラー

エラーコード	説明
-1	メッセージの送信に失敗
-2	最大バッファサイズより長いメッセージを受信
-3	(未使用)
-4	T8 タイムアウト
-5	T3 タイムアウト
-6	T5 タイムアウト
-7	T6 タイムアウト
-8	T7 タイムアウト

WinSock の出すエラー

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

SavoyHsms コントロールが HSMS 接続経由で SECS-II メッセージ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	受信したメッセージ

使用例

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
  ...
}
```

特記事項

参照

3.3.5 SelChanged

画面上で HSMS 接続の選択が変化したときに通知されます。

構文

Visual Basic 6.0

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

Visual C++ 6.0

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

引数	説明
IpszIPAddress	選択されたIPアドレス
IPortNumber	選択されたポート番号

使用例

Visual Basic 6.0

```
' Switch to different client
.IPAddress = IpszIPAddress
.PortNumber = IPortNumber
```

Visual C++ 6.0

```
// Switch to different client
m_ctrl.SetIPAddress(IpszIPAddress);
m_ctrl.SetPortNumber(IPortNumber);
```

特記事項

参照