

SavoySecsISocket 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11月16日	Hikaru Okada	新規に作成。

---

## 2 目次

1	改訂履歴 .....	2
2	目次 .....	3
3	SavoySecsISocket .....	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	DeviceID .....	9
3.1.5	IniFileName .....	10
3.1.6	IniSection .....	11
3.1.7	IPAddress .....	12
3.1.8	Log .....	13
3.1.9	LogBakCount .....	14
3.1.10	LogFileName .....	15
3.1.11	LogSize .....	16
3.1.12	LogVerbose .....	17
3.1.13	Master .....	18
3.1.14	MSEC .....	19
3.1.15	MyPortNumber .....	20
3.1.16	OneOnOne .....	21
3.1.17	PortNumber .....	22
3.1.18	Retry .....	23
3.1.19	Server .....	24
3.1.20	T1 .....	25
3.1.21	T2 .....	26
3.1.22	T3 .....	27
3.1.23	T4 .....	28
3.2	メソッド .....	29
3.2.1	AboutBox .....	29
3.2.2	Disconnect .....	30
3.2.3	LoadIniFile .....	31
3.2.4	Send .....	32
3.2.5	Setup .....	33
3.3	イベント .....	36
3.3.1	Connected .....	36
3.3.2	Disconnected .....	37
3.3.3	Problem .....	38
3.3.4	Received .....	40
3.3.5	Sent .....	41

### 3 SavoySecsISocket

SavoySecsISocket コントロールは、SEMI E4 (SECS-II) の通信を RS232 ではなく LAN (TCP/IP) 経由で送受信するための開発支援コンポーネントです。これは別名 SECS Over LAN としても知られています。SavoySecsISocket コントロールは装置側の通信ソフト開発にも、ホスト側の開発にもどちらにも利用することができます。通常 SavoySecsISocket コントロールは SavoySecsII コントロールと併用して使います。

#### 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	TCP/IP の接続状態を取得または設定します。
DeviceID	デバイス ID 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。
IP Address	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。
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。
Master	マスターかスレーブのどちらの役割で動作するかを取得または設定します。
MSEC	MSEC (三菱 SECS プロトコル) が有効かどうか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。
OneOnOne	TCP/IP 接続が一对一に制限されるか (多対一を不許可とするか) 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。
Retry	送信エラーが発生した場合に何回リトライするか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。
T1	T1 タイムアウトを取得または設定します。
T2	T2 タイムアウト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。
T4	T4 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 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 SECS-II 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。

#### 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	SECS-II メッセージを受信したときに通知されます。
Sent	SECS-II メッセージが送信されたときに通知されます。

## 3.1 プロパティ

### 3.1.1 Appearance

SavoySecsISocket 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

SavoySecsISocket 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、SavoySecsISocket 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、SavoySecsISocket 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 = "terminal_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("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!");
```

#### 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 DeviceID

デバイス ID を取得または設定します。デバイス ID は SECS-II ヘッダの先頭 2 ビット目から 15 ビットです。

SECS-I では下記のヘッダ構造が使用されます。

Byte	説明
1	R   デバイス ID
2	
3	W   ストリーム番号
4	ファンクション番号
5	E   ブロック番号
6	
7	ソース ID
8	
9	トランザクション ID
10	

#### 構文

Visual Basic 6.0
DeviceID As Long

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

#### 使用例

Visual Basic 6.0
.DeviceID = 1

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

#### 特記事項

永続化プロパティ。

#### 参照

### 3.1.5 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

### 3.1.6 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 SavoySecsISocket コントロールが含まれている場合、セクション名を変えることで同じ INI ファイルを共有することができます。

デフォルトは「SavoySecsISocket」です。

#### 構文

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

Visual C++ 6.0
<code>CString GetIniSection() void SetIniSection(LPCTSTR)</code>

#### 使用例

Visual Basic 6.0
<code>.IniSection = "Host" ' [Host] section</code>

Visual C++ 6.0
<code>m_ctrl.IniSection("Host"); // [Host] section</code>

#### 特記事項

永続化プロパティ。

#### 参照

### 3.1.7 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
<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

### 3.1.8 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

### 3.1.9 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、SavoySecsISocket 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、SavoySecsISocket 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

### 3.1.10 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

### 3.1.11 LogSize

ログファイルのサイズ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、SavoySecsISocket 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、SavoySecsISocket 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

### 3.1.12 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。

値	説明
True	詳細ログが有効
False	詳細ログを記録しない

#### 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

### 3.1.13 Master

SavoySecsISocket コントロールがマスターかスレーブのどちらの役割で動作するかを取得または設定します。SEMI の E4 スタンドは装置側をマスター、ホスト側をスレーブにするよう定義しています。

値	説明
False	スレーブ
True	マスター

#### 構文

Visual Basic 6.0
Master As Boolean

Visual C++ 6.0
<pre> BOOL GetMaster() void SetMaster(BOOL) </pre>

#### 使用例

Visual Basic 6.0
<pre> .Master = True ' Equipment .Master = False ' Host </pre>

Visual C++ 6.0
<pre> m_ctrl.SetMaster(true); ' Equipment m_ctrl.SetMaster(false); ' Host </pre>

#### 特記事項

永続化プロパティ。

#### 参照

### 3.1.14 MSEC

MSEC(三菱 SECS プロトコル)が有効かどうかを取得または設定します。

値	説明
False	SECS-I
True	MSEC

#### 構文

Visual Basic 6.0
MSEC As Boolean

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

#### 使用例

Visual Basic 6.0
.MSEC = True

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

#### 特記事項

永続化プロパティ。

#### 参照

### 3.1.15 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 SavoySecsISocket コントロールがクライアント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。

もし SavoySecsISocket コントロールがサーバとして動作している場合、MyPortNumber プロパティは、接続を受け入れるためのサーバポート番号となります。

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

### 3.1.16 OneOnOne

TCP/IP 接続が一对一に制限されるか(多対一を不許可とするか)を取得または設定します。もしこのプロパティが True で既に接続がある場合、SavoySecsISocket コントロールは現在の接続を維持し、新規の接続は受け入れません(拒否します)。もしこのプロパティが False の場合、SavoySecsISocket コントロールは複数の接続(多対一接続)を受け入れます。

値	説明
True	一对一接続のみ
False	多対一接続を許可します

#### 構文

```
Visual Basic 6.0  
  
OneOnOne As Boolean
```

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

#### 使用例

```
Visual Basic 6.0  
  
.OneOnOne = True
```

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

#### 特記事項

永続化プロパティ。

#### 参照

### 3.1.17 PortNumber

TCP/IP 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

### 3.1.18 Retry

送信エラーが発生した場合に何回リトライするかを取得または設定します。

#### 構文

Visual Basic 6.0

```
Retry As Integer
```

Visual C++ 6.0

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

#### 使用例

Visual Basic 6.0

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

Visual C++ 6.0

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

#### 特記事項

永続化プロパティ。

#### 参照

### 3.1.19 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
.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

### 3.1.20 T1

SECS-I の T1 タイムアウトを取得または設定します。単位はミリ秒。

#### 構文

Visual Basic 6.0

```
T1 As Long
```

Visual C++ 6.0

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

#### 使用例

Visual Basic 6.0

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

Visual C++ 6.0

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

#### 特記事項

永続化プロパティ。

#### 参照

### 3.1.21 T2

SECS-I の T2 タイムアウトを取得または設定します。単位はミリ秒。

#### 構文

Visual Basic 6.0

```
T2 As Long
```

Visual C++ 6.0

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

#### 使用例

Visual Basic 6.0

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

Visual C++ 6.0

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

#### 特記事項

永続化プロパティ。

#### 参照

### 3.1.22 T3

SECS-I の T3 タイムアウトを取得または設定します。単位はミリ秒。

#### 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

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

Visual C++ 6.0

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

#### 特記事項

永続化プロパティ。

#### 参照

### 3.1.23 T4

SECS-I の T4 タイムアウトを取得または設定します。単位はミリ秒。

#### 構文

Visual Basic 6.0

```
T4 As Long
```

Visual C++ 6.0

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

#### 使用例

Visual Basic 6.0

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

Visual C++ 6.0

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

#### 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よう。

#### 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()  
    SavoySecsISocket1.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	送信されるメッセージ。

#### 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

```
SavoySecsISocket1.Send SavoySecsII1.Msg
```

Visual C++ 6.0

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

#### 特記事項

SavoySecsISocket コントロールはメッセージを送信しようとします。実際の送信処理結果は Sent イベントで通知されます。

#### 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。

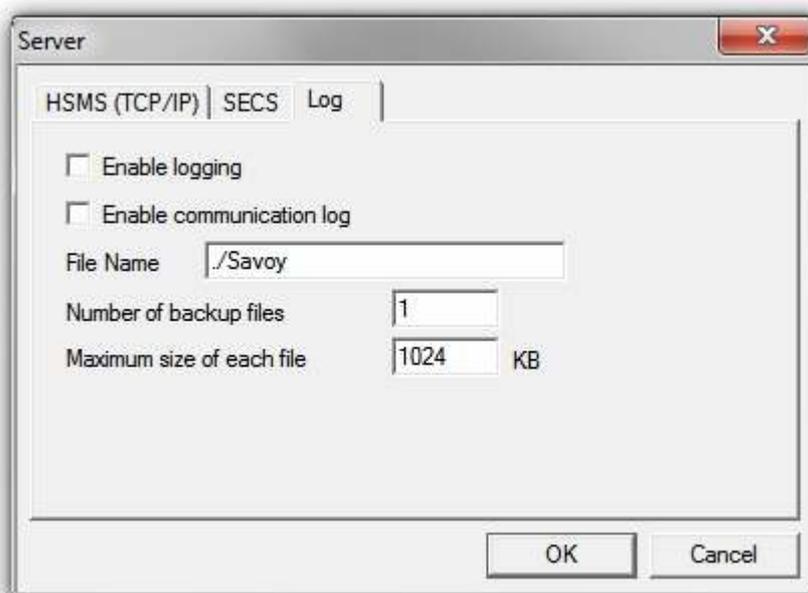
## HSMS (TCP/IP)タブ

項目	説明
Passive Entity (server)	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 (セッション ID)。
Limit to one-on-one connection	一対一接続しか許可しない場合はチェックをつけます。

## SECS タブ

項目	説明
Retry	リトライ回数
Master	マスター(装置)か、スレーブ(ホスト)かの選択
Device ID	デバイス ID
MSEC	三菱 SECS が有効かの選択
T1	T1 タイマー
T2	T2 タイマー
T3	T3 タイマー
T4	T4 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(IpszCaption As String) As Boolean
Visual C++ 6.0
BOOL Setup(LPCTSTR IpszCaption)

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

## 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 "SECS-I over LAN Configuration"
```

```
Visual C++ 6.0
```

```
m_ctrl.Setup("SECS-I over LAN 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

SavoySecsISocket からのエラー

エラーコード	列挙子	説明
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.

WinSock からのエラー

エラーコード	説明
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.

参照

### 3.3.4 Received

SavoySocket コントロールが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の 16 進法文字列

#### 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 Sent

SECS-II メッセージが送信されたときに通知されます。

#### 構文

Visual Basic 6.0

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

Visual C++ 6.0

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

引数	説明
IpszIPAddress	メッセージの発信先のIPアドレス
IPortNumber	メッセージの発信先のポート番号
IpszMsg	送信されたメッセージ

#### 使用例

Visual Basic 6.0

```
SavoySecsII1.Msg = IpszMsg
```

Visual C++ 6.0

```
m_msg.SetMsg(IpszMsg);
```

#### 特記事項

#### 参照