

SavoyLog 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	SavoyLog	4
3.1	プロパティ	5
3.1.1	Appearance	5
3.1.2	BorderStyle	6
3.1.3	Date	7
3.1.4	DateFormat	8
3.1.5	FontBold	9
3.1.6	FontFaceName	10
3.1.7	FontItalic	11
3.1.8	FontSize	12
3.1.9	IniFileName	13
3.1.10	IniSection	14
3.1.11	LastLine	15
3.1.12	LineMax	16
3.1.13	Log	17
3.1.14	LogBakCount	18
3.1.15	LogFileName	19
3.1.16	LogSize	20
3.1.17	SelectedLine	21
3.1.18	Time	22
3.2	メソッド	23
3.2.1	AboutBox	23
3.2.2	Clear	24
3.2.3	LoadIniFile	25
3.2.4	Setup	26
3.2.5	Trace	29

3 SavoyLog

SavoyLog コントロールはアプリケーションにログの機能を追加するための開発支援製品です。文字列は画面にも表示されます。もしログファイルのサイズが指定されたサイズ以上になった場合は SavoyLog はバックアップファイルを作成します。

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。
Date	日付を画面に表示するかどうかを取得または設定します。
DateFormat	日付表現の書式を取得または設定します。
FontBold	フォントが太字かどうかを取得または設定します。
FontFaceName	フォント名を取得または設定します。
FontItalic	フォントが斜体字かどうかを取得または設定します。
FontSize	フォントの高さ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。
LastLine	最終行を常に表示するかどうかを取得または設定します。
LineMax	表示する最大行数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。
SelectedLine	選択されている行を取得または設定します。
Time	時刻を画面に表示するかどうか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。
Clear	画面に表示されている文字を全て消去します。
LoadIniFile	設定内容を INI ファイルから読み出します。
Setup	セットアップ画面を表示します。
Trace	画面に文字列を追加し、ログファイルに記録します。

3.1 プロパティ

3.1.1 Appearance

SavoySecsl 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

SavoySecsl 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 Date

日付を画面に表示するかどうかを取得または設定します。このプロパティが True の場合は日付情報が各行の先頭に挿入されます。

値	説明
True	日付が表示されます。
False	日付は表示されません。

構文

```
Visual Basic 6.0  
Date As Boolean
```

```
Visual C++ 6.0  
BOOL GetDate()  
void SetDate(BOOL)
```

使用例

```
Visual Basic 6.0  
.Date = True
```

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

特記事項

永続化プロパティ。

参照

3.1.4 DateFormat

日付表現の書式を取得または設定します。下記のいずれかを選択できます。

値	説明
0	YYYY/MM/DD
1	MM/DD/YYYY
2	DD/MM/YYYY
3	MMM, DDth, YYYY

書式 0 は日本で使われます。書式 1 はアメリカで一般的です。書式 2 はヨーロッパの多くの国で使用されています。これらの地域間で数字の順序が異なることから、書式 3 はそのような曖昧さを取り除くのに有効です。書式 3 は英語の月の先頭三文字、日付の後に st, nd, rd または th を付加、そして 4 桁の西暦となります。

構文

```
Visual Basic 6.0  
  
DateFormat As Integer
```

```
Visual C++ 6.0  
  
short GetDateFormat()  
void SetDateFormat(short)
```

使用例

```
Visual Basic 6.0  
  
.DateFormat = 3
```

```
Visual C++ 6.0  
  
m_ctrl.SetDateFormat(3);
```

特記事項

永続化プロパティ。

参照

3.1.5 FontBold

フォントが太字かどうかを取得または設定します。

値	説明
True	太字
False	通常

構文

Visual Basic 6.0
FontBold As Boolean

Visual C++ 6.0
BOOL GetFontBold() void SetFontBold(BOOL)

使用例

Visual Basic 6.0
<pre>.FontFaceName = "Arial" .FontSize = 12 .FontBold = False .FontItalic = False</pre>

Visual C++ 6.0
<pre>m_ctrl.SetFontFaceName("Arial"); m_ctrl.SetFontSize(12); m_ctrl.SetFontBold(false); m_ctrl.SetFontItalic(false);</pre>

特記事項

永続化プロパティ。

参照

3.1.6 FontFaceName

フォント名を取得または設定します。

構文

Visual Basic 6.0

```
FontFaceName As String
```

Visual C++ 6.0

```
CString GetFontFaceName()  
void SetFontFaceName(LPCTSTR)
```

使用例

Visual Basic 6.0

```
.FontFaceName = "Arial"  
.FontSize = 12  
.FontBold = False  
.FontItalic = False
```

Visual C++ 6.0

```
m_ctrl.SetFontFaceName("Arial");  
m_ctrl.SetFontSize(12);  
m_ctrl.SetFontBold(false);  
m_ctrl.SetFontItalic(false);
```

特記事項

永続化プロパティ。

参照

3.1.7 FontItalic

フォントが斜体字かどうかを取得または設定します。

値	説明
True	イタリック体
False	通常

構文

Visual Basic 6.0
FontItalic As Boolean

Visual C++ 6.0
BOOL GetFontItalic() void SetFontItalic(BOOL)

使用例

Visual Basic 6.0
<pre>.FontFaceName = "Arial" .FontSize = 12 .FontBold = False .FontItalic = False</pre>

Visual C++ 6.0
<pre>m_ctrl.SetFontFaceName("Arial"); m_ctrl.SetFontSize(12); m_ctrl.SetFontBold(false); m_ctrl.SetFontItalic(false);</pre>

特記事項

永続化プロパティ。

参照

3.1.8 FontSize

フォントの高さを取得または設定します。

構文

Visual Basic 6.0

```
FontSize As Long
```

Visual C++ 6.0

```
long GetFontSize()  
void SetFontSize(long)
```

使用例

Visual Basic 6.0

```
.FontFaceName = "Arial"  
.FontSize = 12  
.FontBold = False  
.FontItalic = False
```

Visual C++ 6.0

```
m_ctrl.SetFontFaceName("Arial");  
m_ctrl.SetFontSize(12);  
m_ctrl.SetFontBold(false);  
m_ctrl.SetFontItalic(false);
```

特記事項

永続化プロパティ。

参照

3.1.9 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

3.1.10 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 SavoyLog 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「SavoyLog」です。

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

3.1.11 LastLine

最終行を常に表示するかどうかを取得または設定します。

構文

Visual Basic 6.0

```
LastLine As Boolean
```

Visual C++ 6.0

```
BOOL GetLastLine()  
void SetLastLine(BOOL)
```

使用例

Visual Basic 6.0

```
.LastLine = True
```

Visual C++ 6.0

```
m_ctrl.SetLastLine(true)
```

特記事項

永続化プロパティ。

参照

3.1.12 LineMax

表示する最大行数を取得または設定します。もしリストボックスの行数がこれより多くなると、SavoyLog コントロールは古いものから削除します。

構文

Visual Basic 6.0

```
LineMax As Integer
```

Visual C++ 6.0

```
short GetLineMax()  
void SetLineMax(short)
```

使用例

Visual Basic 6.0

```
.LineMax = 5000
```

Visual C++ 6.0

```
m_ctrl.SetLineMax(5000);
```

特記事項

永続化プロパティ。

参照

3.1.13 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

3.1.14 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、SavoyLog 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、SavoyLog 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

3.1.15 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

3.1.16 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、SavoyLog 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、SavoyLog 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

3.1.17 SelectedLine

選択されている行を取得または設定します。もしこの数字が 0 のときは、最初の行(一番上の行)が選択されます。もしマイナスの値の場合は、最終行(一番最後の行)が選択されます。もし LastLine プロパティが True の場合は、SelectedLine プロパティの設定は無視され、常に最終行が選択されます。

構文

Visual Basic 6.0

```
SelectedLine As Integer
```

Visual C++ 6.0

```
short GetSelectedLine()  
void SetSelectedLine(short)
```

使用例

Visual Basic 6.0

```
.SelectedLine = -1
```

Visual C++ 6.0

```
m_ctrl.SetSelectedLine(-1);
```

特記事項

参照

3.1.18 Time

時刻を画面に表示するかどうかを取得または設定します。このプロパティが True の場合は時刻情報が各行の先頭に挿入されます。

値	説明
True	時刻が表示されます。
False	時刻は表示されません。

構文

Visual Basic 6.0
Time As Boolean

Visual C++ 6.0
BOOL GetTime() void SetTime(BOOL)

使用例

Visual Basic 6.0
.Time = True

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

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 Clear

画面に表示されている文字を全て消去します。

構文

```
Visual Basic 6.0
```

```
Sub Clear()
```

```
Visual C++ 6.0
```

```
void Clear()
```

戻り値

ありません。

使用例

```
Visual Basic 6.0
```

```
.Clear
```

```
Visual C++ 6.0
```

```
m_ctrl.Clear();
```

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()  
    SavoyLog1.LoadIni  
End Sub
```

Visual C++ 6.0

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

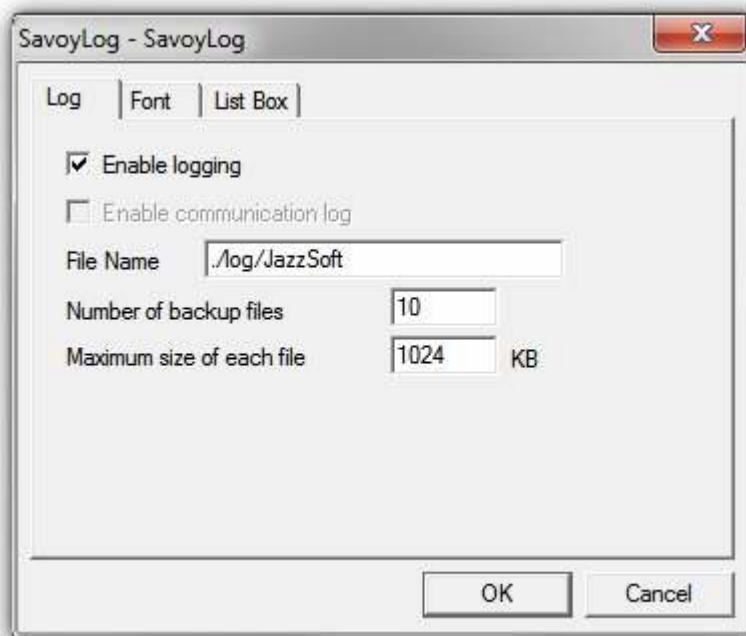
特記事項

参照

3.2.4 Setup

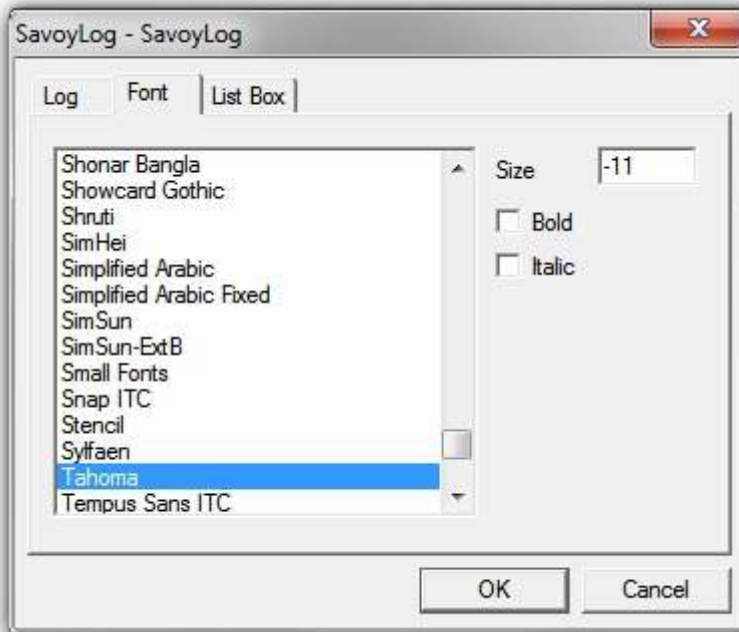
セットアップ画面を表示します。パラメータを変更して OK ボタンを押すと INI ファイルに書き込まれます。

Log タブ



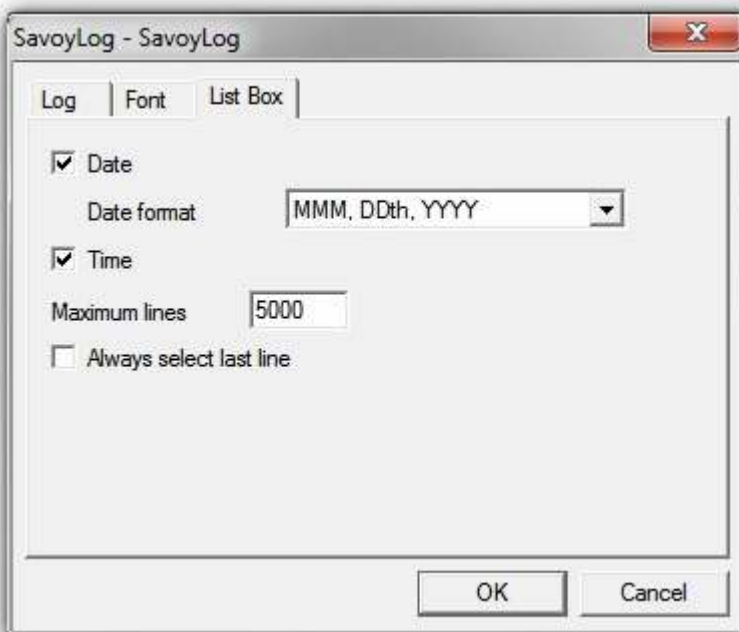
項目	説明
Enable logging	ログに記録する場合はチェックをつけます。
File name	ログファイル名。
Number of backup files	バックアップファイルの数。
Maximum size of each file	ログファイルのサイズ。

Font タブ



項目	説明
Font list box	利用可能なフォントが列挙されます。
Size	フォントの高さ。
Bold	フォントを太字にする場合はチェックをつけます。
Italic	フォントを斜体字にする場合はチェックをつけます。

List Box タブ



項目	説明
Date	日付を表示する場合はチェックをつけます。
Date format	日付の書式を指定します。下記のいずれかになります。 YYYY/MM/DD (日本) MM/DD/YYYY (アメリカ) DD/MM/YYYY (ヨーロッパ)

	MMM, DDth, YYYY (厳格な表現)
Time	時刻情報を表示する場合はチェックをつけます。
Maximum lines	画面に表示される最大行数を指定します。
Always select last line	最終行を常に表示したい場合はチェックをつけます。

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	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 "Savoy Logging Settings"

Visual C++ 6.0
m_ctrl.Setup("Savoy Logging Settings");

特記事項

参照

3.2.5 Trace

画面に文字列を追加し、ログファイルに記録します。

構文

```
Visual Basic 6.0  
Sub Trace(lpszText As String)
```

```
Visual C++ 6.0  
void Trace(LPCTSTR lpszText)
```

引数	説明
lpszText	画面に表示されログファイルに記録される文字列。

戻り値

ありません。

使用例

```
Visual Basic 6.0  
.Trace "Copyright(C) 2009 Jazz Soft, Inc."
```

```
Visual C++ 6.0  
m_ctrl.Trace("Copyright(C) 2009 Jazz Soft, Inc.");
```

特記事項

参照