

EPSON

Epson RC+ 8.0オプション GUI Builder 8.0

翻訳版

© Seiko Epson Corporation 2024-2025

Rev.5
JAM259S7687F

目次

1. はじめに	11
1.1 はじめに	12
1.2 商標	12
1.3 表記について	12
1.4 ご注意	12
1.5 製造元	12
1.6 お問い合わせ先	12
1.7 ご使用の前に	13
1.7.1 Epson RC+8.0のインストールフォルダーについて	13
2. GUI Builder 8.0について	14
2.1 GUI Builder 8.0について	15
2.2 特徴	15
3. インストール	16
3.1 GUI Builderのインストール	17
4. ご使用の前に	18
4.1 ご使用の前に	19
4.2 GUI Builderチュートリアル	19
5. GUI Builder環境	25
5.1 概要	26
5.2 GUI Builderを使用するための基本コンセプト	26
5.3 GUI Builderウィンドウの開き方	26
5.4 GUI Builderウィンドウのそれぞれの機能	27
5.4.1 デザインエリア	27
5.4.2 ツールバー	27
5.4.3 フォームエクスプローラー	29
5.4.4 プロパティグリッド	29
5.4.5 イベントグリッド	30
5.4.6 ステータスバー	30
5.5 フォームとコントロールの使い方	30
5.5.1 フォームの作成	30

5.5.2 フォームの削除	30
5.5.3 フォームの開閉	30
5.5.4 フォーム表示の拡大縮小	31
5.5.5 フォームサイズの変更	31
5.5.6 複数のフォームの編集	31
5.5.7 コントロールの作成	31
5.5.8 コントロールの削除	31
5.5.9 コントロールのサイズ変更と移動	32
5.5.10 コントロールのコピー、切り取り、貼りつけ	32
5.5.11 プロパティの編集	32
5.5.12 イベントハンドラーの使い方	33
5.5.13 変更の保存	34
5.6 GUI Builderの設定	34
5.7 他のプロジェクトからフォームをインポートするには	34
6. GUI Builderの構成要素	36
6.1 フォーム	37
6.1.1 解説	37
6.1.2 使い方	37
6.1.3 フォームプロパティ	38
6.1.4 フォームイベント	39
6.2 ボタンコントロール	39
6.2.1 解説	39
6.2.2 使い方	40
6.2.3 ボタンコントロールのプロパティ	40
6.2.4 ボタンコントロールのイベント	41
6.3 ラベルコントロール	41
6.3.1 解説	41
6.3.2 使い方	41
6.3.3 ラベルコントロールのプロパティ	42
6.3.4 ラベルコントロールのイベント	43
6.4 テキストボックスコントロール	43
6.4.1 解説	43
6.4.2 使い方	44
6.4.3 テキストボックスコントロールのプロパティ	44
6.4.4 テキストボックスコントロールのイベント	45

6.5 ラジオボタンコントロール	46
6.5.1 解説	46
6.5.2 使い方	46
6.5.3 ラジオボタンコントロールのプロパティ	47
6.5.4 ラジオボタンコントロールのイベント	48
6.6 チェックボックスコントロール	48
6.6.1 解説	48
6.6.2 使い方	48
6.6.3 チェックボックスコントロールのプロパティ	48
6.6.4 チェックボックスのコントロールのイベント	50
6.7 リストボックスコントロール	50
6.7.1 説明	50
6.7.2 使い方	50
6.7.3 リストボックスコントロールのプロパティ	51
6.7.4 リストボックスコントロールのイベント	52
6.8 コンボボックスコントロール	52
6.8.1 解説	52
6.8.2 使い方	52
6.8.3 コンボボックスコントロールのプロパティ	53
6.8.4 コンボボックスコントロールのイベント	54
6.9 ピクチャーボックスコントロール	54
6.9.1 解説	54
6.9.2 使い方	54
6.9.3 ピクチャーボックスコントロールのプロパティ	54
6.9.4 ピクチャーボックスコントロールのイベント	55
6.9.5 イメージリスト画面	55
6.10 グループボックスコントロール	56
6.10.1 解説	56
6.10.2 使い方	56
6.10.3 グループボックスコントロールのプロパティ	56
6.10.4 グループボックスコントロールのイベント	57
6.11 タイマーコントロール	57
6.11.1 解説	57
6.11.2 使い方	58
6.11.3 タイマーコントロールのプロパティ	58

6.11.4 タイマーコントロールのイベント	58
6.12 ビデオボックスコントロール	58
6.12.1 解説	58
6.12.2 使い方	59
6.12.3 ビデオボックスコントロールのプロパティ	59
6.12.4 ビデオボックスコントロールのイベント	60
6.13 LEDコントロール	60
6.13.1 解説	60
6.13.2 使い方	60
6.13.3 LEDコントロールのプロパティ	60
6.13.4 LEDコントロールのイベント	62
6.14 ステータスバーコントロール	62
6.14.1 解説	62
6.14.2 使い方	62
6.14.3 ステータスバーコントロールのプロパティ	62
6.14.4 ステータスバーコントロールのイベント	63
6.15 プログレスバーコントロール	63
6.15.1 解説	64
6.15.2 使い方	64
6.15.3 プログレスバーコントロールのプロパティ	64
6.15.4 プログレスバーコントロールのイベント	65
6.16 トラックバーコントロール	65
6.16.1 解説	65
6.16.2 使い方	65
6.16.3 トラックバーコントロールのプロパティ	65
6.16.4 トラックバーコントロールのイベント	67
6.17 グリッドコントロール	67
6.17.1 解説	67
6.17.2 使い方	67
6.17.3 グリッドコントロールのプロパティ	67
6.17.4 グリッドコントロールのイベント	68
6.17.5 グリッドエディター	68
6.18 タブコントロール	69
6.18.1 解説	69
6.18.2 使い方	69

6.18.3 タブコントロールのプロパティ	70
6.18.4 タブコントロールのイベント	71
6.19 ウィザードコントロール	71
6.19.1 解説	71
6.19.2 使い方	71
6.19.3 ウィザードコントロールのプロパティ	72
6.19.4 ウィザードコントロールのイベント	73
6.20 ツリービューコントロール	73
6.20.1 解説	73
6.20.2 使い方	73
6.20.3 ツリービューのプロパティ	75
6.20.4 ツリービューのイベント	77
7. 操作	78
7.1 概要	79
7.2 プログラムモードでGUI開発	79
7.2.1 GUI設計	79
7.2.2 デバッグ	79
7.3 オペレーターモード	80
7.4 一時停止、継続実行の操作	80
7.5 非常停止の操作	81
7.6 ヘルプファイルの使い方	81
8. GUI Builderリファレンス	82
8.1 概要	83
8.2 GUI Builderプロパティとイベント書式	83
8.3 A	84
8.3.1 AcceptButton プロパティ	84
8.3.2 AddItem プロパティ	85
8.3.3 AddNode プロパティ	86
8.3.4 AddRow プロパティ	87
8.3.5 AppendText プロパティ	88
8.3.6 AllowStateChange プロパティ	89
8.4 B	90
8.4.1 BackColor プロパティ	90
8.4.2 BackColorMode プロパティ	91

8.4.3 BorderStyle プロパティ	92
8.5 C	93
8.5.1 Camera プロパティ	93
8.5.2 CancelButton プロパティ	94
8.5.3 CellBackColor プロパティ	95
8.5.4 CellChanged イベント	96
8.5.5 CellForeColor プロパティ	97
8.5.6 CellText プロパティ	98
8.5.7 Checked プロパティ	99
8.5.8 Click イベント	100
8.5.9 Closed イベント	101
8.5.10 Closing イベント	102
8.5.11 ControlBox プロパティ	103
8.5.12 Controls プロパティ	104
8.5.13 Countプロパティ	105
8.6 D	106
8.6.1 DbClick イベント	106
8.6.2 DialogResult プロパティ	107
8.6.3 DropDownStyle プロパティ	108
8.7 E	108
8.7.1 Enabled プロパティ	109
8.7.2 EventTaskType プロパティ	110
8.8 F	111
8.8.1 Font プロパティ	111
8.8.2 FontBold プロパティ	112
8.8.3 FontItalic プロパティ	113
8.8.4 FontName プロパティ	114
8.8.5 FontSize プロパティ	115
8.8.6 ForeColor プロパティ	116
8.8.7 FormBorderStyle プロパティ	117
8.8.8 FormTemplate プロパティ	118
8.9 G	119
8.9.1 GClose ステートメント	119
8.9.2 GGet ステートメント	120
8.9.3 GraphicsEnabled プロパティ	121

8.9.4 GridEditor プロパティ	122
8.9.5 GSet ステートメント	124
8.9.6 GShow ステートメント	125
8.9.7 GShowDialog ファンクション	126
8.9.8 GShowDialog ステートメント	127
8.10 H	128
8.10.1 Height プロパティ	128
8.10.2 HelpButton プロパティ	129
8.10.3 HelpID プロパティ	130
8.11 I	131
8.11.1 Iconプロパティ	131
8.11.2 Image プロパティ	132
8.11.3 ImageAlign プロパティ	133
8.11.4 ImageOff プロパティ	134
8.11.5 ImageOn プロパティ	135
8.11.6 ImageIndex プロパティ	136
8.11.7 Interval プロパティ	137
8.11.8 IOBit プロパティ	138
8.11.9 IOType プロパティ	139
8.12 K	140
8.12.1 KeyPressイベント	140
8.13 L	141
8.13.1 LargeChange プロパティ	141
8.13.2 Left プロパティ	142
8.13.3 List プロパティ	143
8.13.4 ListCount プロパティ	144
8.13.5 Loadイベント	145
8.14 M	146
8.14.1 MaximizeBox プロパティ	146
8.14.2 Maximum プロパティ	147
8.14.3 MinimizeBox プロパティ	148
8.14.4 Minimum プロパティ	149
8.14.5 MultiLine プロパティ	150
8.15 N	151
8.15.1 Name プロパティ	151

8.16 O	152
8.16.1 Orientation プロパティ	152
8.17 P	153
8.17.1 Parent プロパティ	153
8.17.2 PasswordChar プロパティ	154
8.17.3 PressDelay プロパティ	155
8.17.4 PressSound プロパティ	156
8.17.5 ProgressBarStyle プロパティ	157
8.18 R	158
8.18.1 ReadOnly プロパティ	158
8.18.2 RemoveNode プロパティ	159
8.18.3 RemoveRow プロパティ	160
8.18.4 Resize イベント	161
8.18.5 RobotNumber プロパティ	162
8.18.6 RowCount プロパティ	163
8.19 S	164
8.19.1 Scroll イベント	164
8.19.2 ScrollBars プロパティ	165
8.19.3 SelectedNodeChanged イベント	166
8.19.4 SelectedIndex プロパティ	167
8.19.5 SelectionChanged イベント	168
8.19.6 ShowDateTime プロパティ	169
8.19.7 ShowEStop プロパティ	170
8.19.8 ShowNavigators プロパティ	171
8.19.9 ShowPrint プロパティ	172
8.19.10 ShowRobot プロパティ	173
8.19.11 ShowSafeguard プロパティ	174
8.19.12 SizeMode プロパティ	175
8.19.13 SmallChange プロパティ	176
8.19.14 Sorted プロパティ	177
8.19.15 StartPosition プロパティ	178
8.20 T	179
8.20.1 TabAlignment プロパティ	179
8.20.2 TabEditor プロパティ	180
8.20.3 TabHeaderSize プロパティ	181

8.20.4 TabHeaderTextAlign プロパティ	182
8.20.5 TabHeaderAutoFit プロパティ	183
8.20.6 TabHeaderHeight プロパティ	184
8.20.7 TabHeaderWidth プロパティ	185
8.20.8 TabIndex プロパティ	186
8.20.9 TabPageBackColor プロパティ	187
8.20.10 TabPageFont プロパティ	188
8.20.11 TabPageFontBold プロパティ	189
8.20.12 TabPageFontItalic プロパティ	190
8.20.13 TabPageFontName プロパティ	191
8.20.14 TabPageFontSize プロパティ	192
8.20.15 TabPageIcon プロパティ	193
8.20.16 TabPageText プロパティ	194
8.20.17 Text プロパティ	195
8.20.18 TextAlign プロパティ	196
8.20.19 Tickイベント	197
8.20.20 TickFrequency プロパティ	198
8.20.21 TickStyle プロパティ	199
8.20.22 ToolTipText プロパティ	200
8.20.23 Top プロパティ	201
8.20.24 TreeNodeEditor プロパティ	202
8.20.25 Typeプロパティ	203
8.21 U	204
8.21.1 Updateプロパティ	204
8.22 V	205
8.22.1 Value プロパティ	205
8.22.2 Variable プロパティ	206
8.22.3 VideoEnabled プロパティ	207
8.22.4 Visible プロパティ	208
8.23 W	209
8.23.1 Width プロパティ	209
8.23.2 WindowState プロパティ	210
8.23.3 WizardEditor プロパティ	211
8.23.4 WordWrap プロパティ	212

1. はじめに

1.1 はじめに

このたびは当社のロボットシステムをお求めいただきましてありがとうございます。本マニュアルは、ロボットシステムを正しくお使いいただくために必要な事項を記載したものです。

システムをご使用になる前に、本マニュアルおよび関連マニュアルをお読みいただき、正しくお使いください。お読みになった後は、いつでも取り出せる所に保管し、不明な点があったら再読してください。

当社は、厳密な試験や検査を行い、当社のロボットシステムの性能が、当社規格に満足していることを確認しております。マニュアルに記載されている使用条件を超えて、当社ロボットシステムを使用した場合は、製品の基本性能は発揮されませんのでご注意ください。

マニュアルの内容は、当社が予見する範囲の、危険やトラブルについて記載しています。当社のロボットシステムを、安全に正しくお使いいただくため、マニュアルに記載されている安全に関するご注意は、必ず守ってください。

1.2 商標

Microsoft, Windows, Windowsロゴは、米国Microsoft Corporationの米国およびその他の国における登録商標または商標です。その他の社名, ブランド名, 製品名は、各社の登録商標または商標です。

1.3 表記について

Microsoft® Windows® 10 operating system 日本語版

Microsoft® Windows® 11 operating system 日本語版

本取扱説明書では、上記オペレーティングシステムをそれぞれ、Windows 10, Windows 11と表記しています。また、Windows 10, Windows 11を総称して、Windowsと表記することがあります。

1.4 ご注意

本取扱説明書の一部、または全部を無断で複製や転載をすることはできません。

本書に記載の内容は、将来予告なく変更することがあります。

本書の内容について、誤りや、お気づきの点がありましたら、ご連絡くださいますようお願いいたします。

1.5 製造元

セイコーエプソン株式会社

1.6 お問い合わせ先

お問い合わせ先の詳細は、以下のマニュアルの"販売元"に記載しています。

ご利用の地域によって、お問い合わせ先が異なりますのでご注意ください。

"安全マニュアル - お問い合わせ先"

安全マニュアルは、以下のサイトからも閲覧できます。

URL: <https://download.epson.biz/robots/>



1.7 ご使用の前に

マニュアルのご使用の前に、知っておいていただきたいことを記載しています。

1.7.1 Epson RC+8.0のインストールフォルダーについて

Epson RC+8.0は、インストールフォルダーパスを任意の場所に変更が可能です。本マニュアルでは、C:\¥EpsonRC80にEpson RC+ 8.0がインストールされた場合を想定して説明しています。

2. GUI Builder 8.0について

2.1 GUI Builder 8.0について

GUI Builder 8.0は、GUI(グラフィカルユーザーインターフェイス)をSPEL+アプリケーションに簡単に作成できるEpson RC+オプションです。Epson RC+開発環境でSPEL+アプリケーションGUIを作成するためにより使いやすくデザインされた統合ツールで、Visual Studioなどのサードパーティー製品を使用せずに、シンプルなGUIを必要としているユーザーにとって、GUI Builderは理想的です。また、これまでにGUIを作成したことがなくても、GUI Builderを使用すれば簡単にGUIを作成することができます。

キーポイント

より高度なGUIアプリケーションには、Epson RC+ 8.0 RC+ APIオプションを使用することでVisual Studioなどのツールと連携したアプリケーションを開発できます。

2.2 特徴

GUI Builder 8.0には、次のような特徴があります。

- GUIは、Epson RC+環境に完全に統合されており、簡単デザイン、デバッグ、プログラム実行中の表示ができます。第三者ツールは必要ありません。
- Epson RC+プロジェクト内でGUIフォームの作成、デバッグが可能です。
- ボタン、ラベル、テキストボックスなどの標準的なコントロールがあります。さらに、ビジョン画像の表示や、コントローラーの状態表示、I/O状態表示も可能です。
- フォームとコントロールのイベントは、SPEL+タスクとして作成されます。これらのタスクには、Normal、NoPause、またはNoEmgAbortモードの指定が可能です。
- オペレーターモードでは、Epson RC+が起動時にユーザーのメインフォームを自動表示します。また、SPEL+プログラムからフォームを表示させることもできます。

3. インストール

3.1 GUI Builderのインストール

本章の指示にしたがって、GUI Builder 8.0をインストールしてください。

インストールの前に、すべてのWindowsアプリケーションを終了してください。

1. Epson RC+ 8.0をインストールします。GUI Builder 8.0オプションは、自動的にインストールされます。
2. GUI Builder 8.0を使用する前に、コントローラーでソフトウェアキーの設定を有効にしておきます。オプションを使用可能にする方法は、以下のマニュアルを参照してください。"Epson RC+ 8.0 ユーザーズガイド"

これで、GUI Builder 8.0のインストールは終了です。

4. ご使用の前に

4.1 ご使用の前に

GUI Builder 8.0のご使用の前に、以下の本章をお読みください。

GUI Builder 8.0 オプションが有効になっていることを確認してください。詳細は、以下を参照してください。

GUI Builderのインストール

Epson RC+ を初めてご使用になる方は、以下のマニュアルをお読みください。

"Epson RC+ 8.0 ユーザーズガイド - プロジェクトとプログラムの作成方法について"

次の項では、いくつかのコンセプトを簡単に説明するチュートリアルを示します。

4.2 GUI Builderチュートリアル

この項では、ロボットのサイクル動作を実行する、簡単なGUIアプリケーションを作成します。次の手順にしたがってください。なお、GUI Builderの用語および、解説は、4.2項以降をご覧ください。

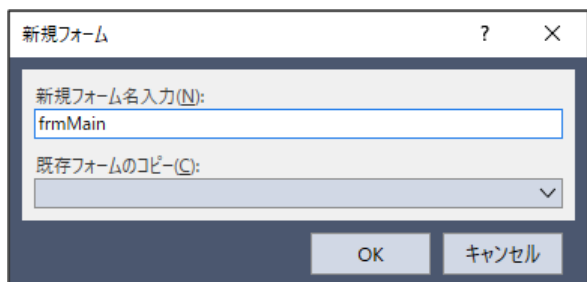
- ロボットのサイクル動作を行うファンクションを持つ新しいプロジェクトを作成します。
- ロボットのサイクル動作を開始、停止するためのボタンを含むフォームを作成します。ここでは、フォームの作成や、開始／停止イベントを実行するSPEL+のタスクに関連づけられたボタンの配置方法を示します。
- [一時停止]と[継続実行]ボタンをフォームに追加します。ここではEventTaskTypeプロパティーについて説明します。
- セットアップ用のフォームを追加します。このフォームは、ロボットの速度を変更するためのラベルと、テキストボックスを使います。
- セットアップ用のフォームを表示するためのボタンをメインフォームに配置します。ここでは、GShowDialogステートメントと、DialogResultプロパティーを説明します。

それでは、次の手順にしたがって、プログラムを作成してください。

1. “GUITest”という名前の新しいEpson RC+ 8.0プロジェクトを作成します。
2. ロボットマネージャーを使用し、P0, P1の2つの異なるポイントデータをティーチングします。
3. “Main.prg”に、“main関数”を作成します。

```
Function main  Robot 1
  Motor On
  Do      Go P0      Wait 0.5
          Go P1      Wait 0.5
  LoopFend
```

4. ツールメニューから[GUI Builder]を選択し、GUI Builderウィンドウを表示します。
5. GUI Builderウィンドウのツールバーにある[新規フォーム]ボタンをクリックし、“frmMain”という名前の新しいフォームを作成し、[OK]ボタンをクリックします。



6. GUI Builderウィンドウのツールバーにある[新規ボタン]ボタンをクリックし、フォーム上でクリックします。クリックした場所に、新しいボタンが作成されます。

7. 新しく作成されたボタンのプロパティを設定します。

プロパティのグリッドを下へスクロールして[Name]プロパティを表示します。名前を“btnStart”に変更します。キーボードの[ENTER]キーを押します。

プロパティ	設定値
Enabled	1 - True
EventTaskType	Normal
Font	Microsoft Sans Serif, 8.25pt
ForeColor	■ ControlText
Height	45
Image	
ImageAlign	MiddleCenter
Left	34
Name	btnStart
PressDelay	0
PressSound	0 - False
TabIndex	1

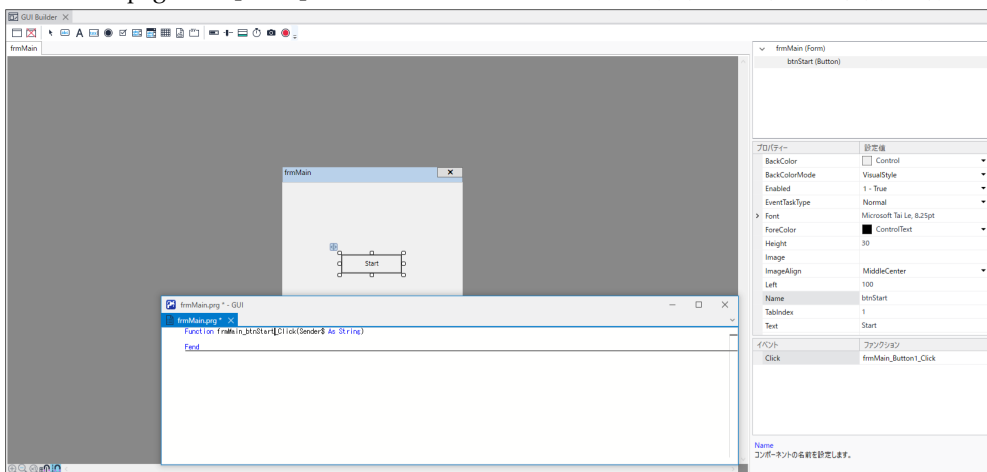
8. プロパティのグリッドを下へスクロールして、[Text]プロパティを表示します。

“Button1”を“Start”に変更します。[ENTER]キーを押します。

プロパティ	設定値
Left	34
Name	btnStart
PressDelay	0
PressSound	0 - False
TabIndex	1
Text	Start
TextAlign	MiddleCenter
ToolTipText	
Top	34
Visible	1 - True
Width	52

9. フォーム上の[Start]ボタンをダブルクリックします。

“frmMain.prg”に、[Start]ボタンをクリックしたときに実行する新しい関数を作成されます。

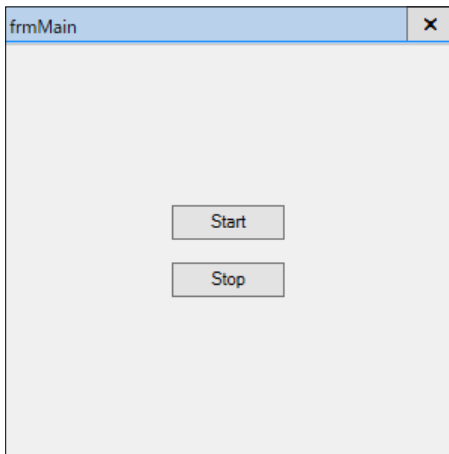


10. “main関数”を起動させるため、次のようにfrmMain_btnStart_Click関数を変更します。

これによって、[Start]ボタンが押されたときに、“main関数”がスタートします。

```
Function frmMain_btnStart_Click(Sender$ As String)
    Xqt main
End
```

11. GUI Builderウィンドウのツールバーにある[新規ボタン]ボタンをクリックし、[Start]ボタンの下の位置をクリックして、別のボタンを作成します。
12. 新しいボタンの[Name]プロパティを“btnStop”に、[Text]プロパティを“Stop”に変更します。
ここまでの手順で、次のようなフォームになっているはずです。



13. [Stop]ボタンをダブルクリックして、イベントハンドラーを作成します。
関数を次のように変更します。

```
Function frmMain_btnStop_Click(Sender$ As String)
    Quit main
End
```

14. キーボードの[F5]キーを押して、プロジェクトをビルドし、Runウィンドウを表示します。
ここでビルドエラーが発生した場合は、プログラムを修正して、再度[F5]キーを押してビルドを行ってください。
15. Runウィンドウで[フォーム]ボタンを選択します。
16. [開始]ボタンをクリックします。
17. frmMainフォームが表示されます。[frmMain]の[Start]ボタンをクリックします。
ロボットが、P0とP1の間を動作します。
18. [frmMain]フォームの[Stop]ボタンをクリックします。
ロボットが停止します。
19. 今度は、フォームの右上の[閉じる]ボタンをクリックします。
フォームが閉じます。
次に、[Pause]ボタンと[Continue]ボタンをGUIに加えてみましょう。
20. GUI Builderウィンドウのツールバーにある[新規ボタン]ボタンをクリックします。
[Start]ボタンの右側の位置をクリックし、新しいボタンを作成します。
21. 新しいボタンのNameプロパティをbtnPauseに、TextプロパティをPauseに変更します。
22. [Pause]ボタンのEventTaskTypeプロパティをNoPauseに変更します。
これにより、イベントハンドラーは他の通常タスクが一時停止しているときに一時停止せずに、Pauseステートメントを実行できます。
23. [Pause]ボタンをダブルクリックして、イベントハンドラー関数を作成します。
次のようなPauseステートメントを追加します。

```
Function frmMain_btnPause_Click(Sender$ As String)
    Pause
End
```

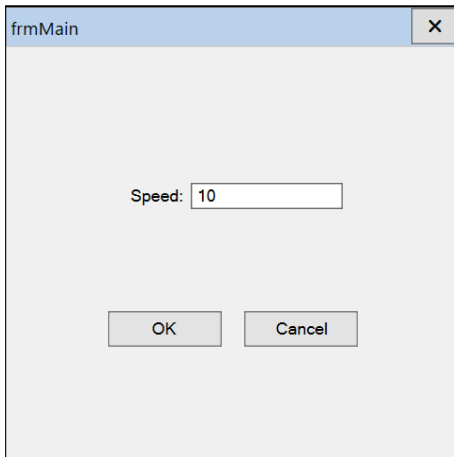
24. GUI Builderウィンドウのツールバーにある[新規ボタン]ボタンをクリックします。
[Stop]ボタンの右側の位置をクリックして、新しいボタンを作成します。
25. 新しいボタンのNameプロパティをbtnContに、TextプロパティをContinueに変更します。
26. [Continue]ボタンのEventTaskTypeプロパティをNoPauseに変更します。
これにより、イベントハンドラーは、他の通常タスクが一時停止しているときに、Contステートメントを実行できるようになります。
27. [Continue]ボタンをダブルクリックして、イベントハンドラー関数を作成します。
次に、以下に示すようにContステートメントを追加します。

```
Function frmMain_btnCont_Click(Sender$ As String)
    Cont
End
```

28. キーボードの[F5]キーを押して、プロジェクトをビルドし、Runウィンドウを表示します。
ここでビルドエラーが発生した場合は、プログラムを修正して、再度 [F5]キーを押してビルドを行ってください。
29. Runウィンドウの[開始]ボタンをクリックして、作成したフォームを表示します。
30. [Start]ボタンをクリックします。
ロボットのサイクル動作が実行されます。
31. [Pause]ボタンをクリックします。
ロボットのサイクル動作が一時停止します。
32. [Continue]ボタンをクリックします。
ロボットのサイクル動作が再開します。
33. [Stop]ボタンをクリックします。
次にフォーム右上の[閉じる]ボタンをクリックして、フォームを閉じます。
次に、セットアップフォームを追加してみましょう。
34. GUI Builderウィンドウのツールバーにある[新規フォーム]ボタンをクリックし、frmSetupという名前の新しいフォームを作成します。
[OK]ボタンを押すと、新しいタブページに新しいフォームが表示されます。
35. [新規ラベル]ボタンをクリックします。
[frmSetup]フォーム上をクリックして、新しいラベルを作成します。
36. NameプロパティをlblSpeedに変更します。
37. TextプロパティをSpeed:に変更します。
38. [新規テキストボックス]ボタンをクリックします。
[frmSetup]フォームの[Speed]ラベルの右側をクリックして、新しいテキストボックスを作成します。
39. NameプロパティをtxtSpeed、Textプロパティを10に変更します。
40. [frmSetup]フォームをダブルクリックすると、フォームのLoadイベントハンドラー関数が作成されます。
次のようにプログラムを追加してください。

```
Function frmSetup_Load(Sender$ As String)
    GSet frmSetup.txtSpeed.Text, Str$(Speed(1))
Fend
```

41. GUI Builderウィンドウのツールバーにある[新規ボタン]ボタンをクリックして、新しいボタンを作成します。
NameプロパティをbtnOK、TextプロパティをOKに変更します。
42. GUI Builderウィンドウのツールバーにある[新規ボタン]ボタンをクリックして新しいボタンを作成します。
NameプロパティをbtnCancel、TextプロパティをCancelに変更します。この時点で、セットアップフォームは以下のようになります。



43. [OK]ボタンをダブルクリックしてイベントハンドラー関数を作成し、次のようにフォームのDialogResultプロパティを設定するプログラムを追加します。

```
Function frmSetup_btnOK_Click(Sender$ As String)
    GSet frmSetup.DialogResult, DIALOGRESULT_OK
    GClose frmSetup
Fend
```

44. [Cancel]ボタンをダブルクリックしてイベントハンドラー関数を作成し、次のようにフォームのDialogResultプロパティを設定するプログラムを追加します。

```
Function frmSetup_btnCancel_Click(Sender$ As String)
    GSet frmSetup.DialogResult, DIALOGRESULT_CANCEL
    GClose frmSetup
Fend
```

45. GUI Builderウィンドウの[frmMain]タブをクリックして、[frmMain]フォームを修正します。
46. GUI Builderウィンドウのツールバーにある[新規ボタン]ボタンをクリックして新しいボタンを作成します。
NameプロパティをbtnSetup、TextプロパティをSetupに変更します。
47. [Setup]ボタンをダブルクリックしてイベントハンドラー関数を作成し、次のように[frmSetup]ダイアログを表示してロボットの動作速度を設定するプログラムを追加します。

```
Function frmMain_btnSetup_Click(Sender$ As String)
    Integer result
    String value$

    result = GShowDialog(frmSetup)
    If result = DIALOGRESULT_OK Then
        GGet frmSetup.txtSpeed.Text, value$
        Speed Val(value$)
    EndIf
Fend
```

48. [F5]キーを押してプロジェクトをビルドし、Runウィンドウを表示します。
49. Runウィンドウの[開始]ボタンをクリックし、作成したフォームを表示します。
50. [frmMain]フォームの[Start]ボタンをクリックします。
ロボットのサイクル動作が実行されます。
51. [frmMain]フォームの[Setup]ボタンをクリックすると、現在のロボットの動作速度をテキストボックスに表示した[frmSetup]ダイアログを表示します。
52. 新しい速度を入力して、[OK]ボタンをクリックします。
53. ロボットのサイクル動作が、設定した動作速度に変更されます。
[Stop]ボタンをクリックしてロボットを停止すると、動作速度はデフォルトに戻ります。
これでこのチュートリアルは終了です。

5. GUI Builder環境

5.1 概要

本章では、GUI Builderと構成要素を理解していただくために、次のコンセプトと定義について説明します。

- GUI Builderを理解するための基本コンセプト
- GUI Builderウィンドウの開き方
- GUI Builderウィンドウのそれぞれの機能
- フォームとコントロールの使い方
- GUI Builderの設定

5.2 GUI Builderを使用するための基本コンセプト

本章をより理解するために、簡単にいくつかの基本コンセプトを説明します。本章を読み進める前に確認してください。

■ GUIとは？

GUI(グイ、ジーユーアイ)は、Graphical User Interface(グラフィカルユーザーインターフェイス)の略です。SPEL+アプリケーションの実行や各設定などを、より簡単に対話的に行うことができます。GUIの基本部品は、“フォーム”と呼ばれます。

■ フォームとは？

フォームとは、コントロールを含むウィンドウ、またはダイアログボックスです。フォームが、GUIアプリケーションの基本ユニットになります。実行時にフォームが表示され、フォーム上のコントロールが有効になり、キーボードやマウス操作のイベントを受け取ります。GUIプロジェクトは、1つまたはいくつかのフォームから構成されます。

- **コントロールとは？** コントロールとは、フォームに含まれるボタンや、チェックボックス、テキストボックスなどのオブジェクトです。各コントロールに固有のプロパティ、イベントがあります。

■ イベントとは？

イベントとは、フォームやコントロールのイベントが発生したときにGUIから呼び出されるSPEL+関数です。例えばオペレーターがボタンをクリックしたとき、ボタンクリックイベントが、クリックされたときに指定されたSPEL+関数を呼び出します。

5.3 GUI Builderウィンドウの開き方

GUI Builderウィンドウは、Epson RC+の開発環境から開くことができます。

Epson RC+が起動している状態から、次の2つの方法でGUI Builderウィンドウを開きます。

■ ツールバー

ツールバーにGUI Builderアイコン  があります。GUI Builderアイコンをクリックすると、GUI Builderウィンドウが開きます。

■ ツールメニュー

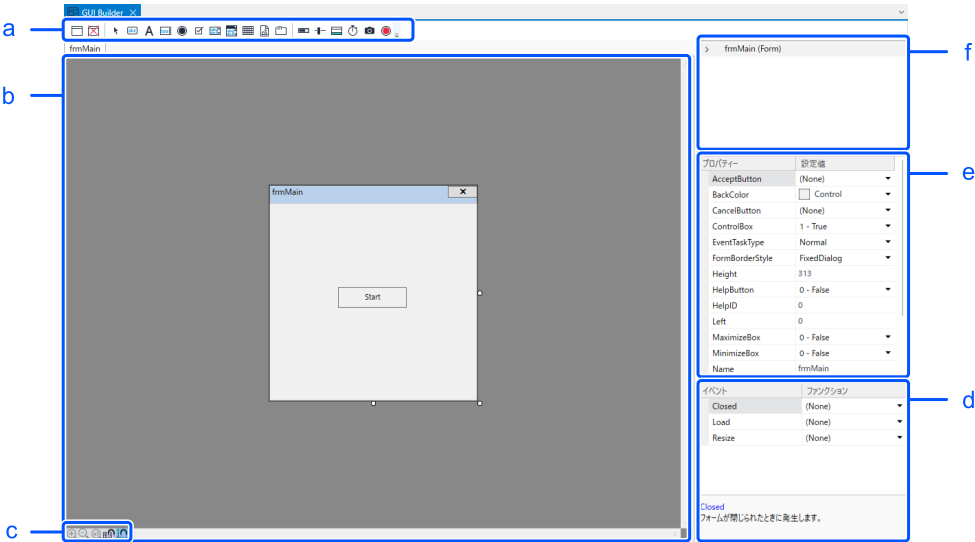
ツールメニューのGUI Builderから、GUI Builderウィンドウを開くことができます。

GUI Builderウィンドウを開くと、GUIアプリケーションが作成できるようになります。

次に、GUI Builderウィンドウの各部分について説明していきます。

5.4 GUI Builderウィンドウのそれぞれの機能

GUI Builderウィンドウを以下に示します。各部分を次の項で説明します。



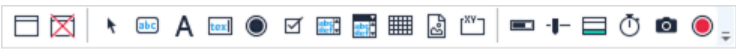
記号	項目
a	ツールバー
b	デザインエリア
c	ステータスバー
d	イベントグリッド
e	プロパティグリッド
f	フォームエクスプローラー

5.4.1 デザインエリア













設計時のフォームが表示される場所です。
タブごとにそれぞれのフォームが表示されます。タブをクリックするか、フォームエクスプローラーでフォームの名前をダブルクリックすることで、表示されるフォームを切り替えることができます。
フォームがデザインエリアより大きい場合、スクロールバーが表示され、フォームのすべてのエリアを操作することができます。








5.4.2 ツールバー

ツールバーには、フォームやコントロールの作成ボタンが含まれます。
GUI Builderのツールバーは、GUI Builderウィンドウの上部にあるタイトルバーのすぐ下に、次のように表示されます。



GUI Builderツールバーの各ボタンについて説明します。

アイコン	名称	説明
	新規フォーム	新しいフォームを作成します。フォームの名前を入力するためのダイアログが表示されます。
	フォーム削除	現在のプロジェクト内の選択しているフォームを削除します。フォームがない場合は、このボタンは無効になります。
	ポインター	新しいコントロールの追加を中断して、通常の操作に戻します。
	新規ボタン	新しいボタンコントロールを作成します。
	新規ラベル	文字を表示するための、新しいラベルコントロールを作成します。
	新規テキストボックス	文字を入力するための、新しいテキストボックスコントロールを作成します。
	新規ラジオボタン	新しいラジオボタンを作成します。
	新規チェックボックス	新しいチェックボックスコントロールを作成します。
	新規リストボックス	新しいリストボックスコントロールを作成します。
	新規コンボボックス	新しいコンボボックスコントロールを作成します。
	新規グリッド	新しいグリッドコントロールを作成します。
	新規ピクチャーボックス	新しいピクチャーボックスコントロールを作成します。

アイコン	名称	説明
	新規グループボックス	新しいグループボックスコントロールを作成します。
	新規プログレスバー	新しいプログレスバーコントロールを作成します。
	新規トラックバー	新しいトラックバーコントロールを作成します。
	新規ステータスバー	新しいステータスバーコントロールを作成します。
	新規タイマー	新しいタイマーコントロールを作成します。
	新規ビデオボックス	新しいビデオボックスコントロールを作成します。このコントロールは、Vision Guideオプションの画像を表示します。
	新規LED	新しいLEDコントロールを作成します。このコントロールは、I/Oの状態を表示します。このコントロールをダブルクリックすることで、I/Oの出力状態を変更するような使い方も可能です。表示色を変えたり、LEDの代わりにイメージ画像を表示することも可能です。

5.4.3 フォームエクスプローラー

フォームエクスプローラーは、現在のプロジェクトのフォームや関連するコントロールをツリー表示します。新しいフォームやコントロールが追加されると、このツリーに加えられます。

ツリー上のフォームをクリックすると、デザインエリアにフォームを開きます。選択されているフォームのプロパティーやイベントが、プロパティーグリッド、イベントグリッドに表示されます。フォーム上で右クリックすると、スタートアップフォームの設定、削除、閉じるのメニューが表示されます。

コントロールをクリックすると、関連するフォームが開き、選択したコントロールがフォーカスされます。コントロールのプロパティーとイベントは、プロパティーグリッド、イベントグリッドに表示されます。

5.4.4 プロパティーグリッド

プロパティーグリッドでは、フォームとコントロールのプロパティーを表示、編集します。

フォームやコントロールを選択すると、そのプロパティーがグリッド上に表示されます。プロパティーグリッドには、プロパティーと設定値の2つの列があります。

プロパティー: プロパティーの名前

設定値: プロパティーの現在値(編集可能)

5.4.5 イベントグリッド

イベントグリッドでは、フォームとコントロールのイベントを表示、編集します。各イベントは、そのイベントで呼び出されるユーザー関数を持っています。

5.4.6 ステータスバー

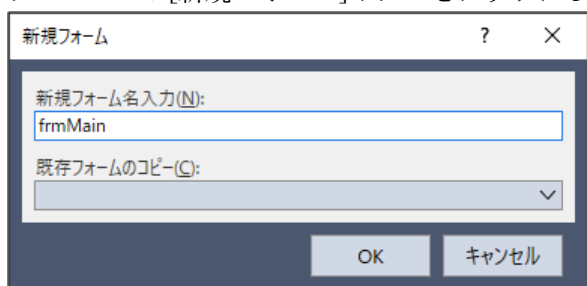
ステータスバーでは、フォーム表示の拡大縮小、レイアウトモードを設定します。

5.5 フォームとコントロールの使い方

5.5.1 フォームの作成

フォームを作成します。

1. GUI Builderウィンドウを開きます。
2. ツールバーの[新規フォーム]ボタンをクリックします。



3. フォームの名前を入力します。次の名前は使用することができません。

- 名前の先頭に数字を使用したフォームの名前
- 編集しているプロジェクト名称と同じフォームの名前

[新規フォーム]ダイアログからは、フォームリストに表示される他のフォームからコピーしてフォームを作成することも可能です。

5.5.2 フォームの削除

フォームを削除するには、フォームエクスプローラー上で右クリックし、削除メニューを選択します。確認のメッセージが表示されたら、[はい]をクリックしてフォームを削除します。

また、次の方法でもフォームを削除できます。

- ツールバーにある[フォーム削除]ボタンをクリック
- フォームのタブを右クリック - [削除]メニューを選択
- プロジェクトエクスプローラー上の対象フォームを右クリック - [削除]メニューを選択

5.5.3 フォームの開閉

フォームを開くには、次の2つの方法があります。

- プロジェクトエクスプローラからフォームをダブルクリックします。
- フォームエクスプローラーからフォームをクリックします。

フォームを閉じるには、フォームエクスプローラーを右クリックして、[閉じる]メニューを選択します。または、フォームのタブを右クリックして、[閉じる]メニューを選択します。

5.5.4 フォーム表示の拡大縮小

フォームの表示サイズを拡大縮小するには、次の2つの方法があります。

- GUI Builderウィンドウのデザインエリア上で任意の場所をクリックした後、[Ctrl]キーを押したままマウスホイールを操作します。
- GUI Builderウィンドウのステータスバー上の次のボタンをクリックし操作します。

ボタン表示	説明
+	フォームの表示を拡大します。
—	フォームの表示を縮小します。
×1	フォームの表示を標準倍率にします。

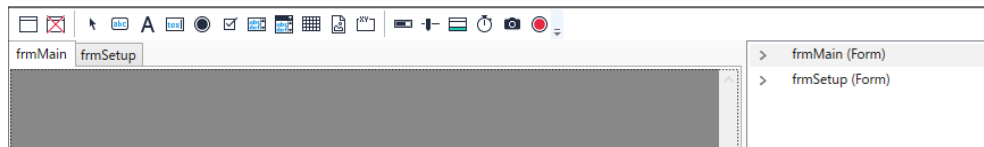
5.5.5 フォームサイズの変更

フォームのサイズを変更するには、次の2つの方法があります。

- GUI Builderウィンドウのデザインエリアのフォームに表示されているフォームの縁をマウスでつかんでドラッグします。
- プロパティグリッドのHeight(高さ)、Width(幅)プロパティの値を変更するか、SPEL+プログラムからGSetステートメントを使って値を書き換えます。

5.5.6 複数のフォームの編集

GUI Builderウィンドウは、タブインターフェースを使って複数のフォームを切り替えます。タブをクリックすることで、現在のフォームを切り替えることができます。



フォームからコントロールをコピーして、他のフォームに貼りつけることもできます。

5.5.7 コントロールの作成

コントロールを作成します。

1. コントロールを追加したいフォームを開きます。
2. GUI Builderウィンドウのツールバーにあるコントロールボタンをクリックします。
カーソルがクロスカーソルに変わります。
3. コントロールを配置したい場所でクリックすると、デフォルトのサイズでコントロールを配置します。
マウスでコントロールの外形をドラッグすると、自由にコントロールの大きさを変えられます(タイマー以外)。
4. プロパティグリッドで各プロパティを設定します。
Nameプロパティの先頭の文字は数字にできません。

5.5.8 コントロールの削除

コントロールを削除します。

1. コントロールを削除したいフォームを開きます。.
2. 削除したいコントロールをクリックします。
複数のコントロールを削除する場合、[Ctrl]キーまたは[Shift]キーを押したまま別のコントロールをクリックします。

3. キーボードの[Del]キーを押すと、選択したコントロールが削除されます。

5.5.9 コントロールのサイズ変更と移動

コントロールのサイズを変更します。


- GUI Builderウィンドウのデザインエリアのフォームに表示されているコントロールの縁をマウスでつかんでドラッグします。

注意

複数のコントロールが選択されている状態では操作できません。

- プロパティグリッドのHeight(高さ)、Width(幅)プロパティの値を変更するか、SPEL+プログラムからGSetステートメントを使って値を書き換えます。
- [Ctrl]キーを押したまま、サイズ変更したい方向の矢印[↑/↓/←/→]をクリックします。サイズ変更量を多くしたい場合、[Ctrl]キーに加え、[Shift]キーも押したまま、矢印[↑/↓/←/→]をクリックします。

コントロールを移動します。

- コントロールをクリックして、選択した状態にします。コントロール上にカーソルを移動させると、カーソルが移動カーソル  に変わります。その状態でコントロールをドラッグし、新しい場所に配置します。
- プロパティグリッドのLeft、Topプロパティの値を変更するか、SPEL+プログラムからGSetステートメントを使って値を書き換えます。

5.5.10 コントロールのコピー、切り取り、貼りつけ

コピーと切り取りのために、コントロールを選択します。

まず、コントロールをクリックします。複数のコントロールを選択するには、[Ctrl]キーまたは[Shift]キーを押したまま別のコントロールをクリックします。

選択したコントロールをコピーします。次の3つの方法があります。

- [Ctrl] + [C]キーを押す
- ツールバーの[コピー]ボタンをクリックする
- 編集メニューの[コピー]を選択する

選択したコントロールを切り取ります。次の3つの方法があります。

- [Ctrl] + [X]キーを押す
- ツールバーの[切り取り]ボタンをクリックする
- 編集メニューの[切り取り]を選択する

選択したコントロールを貼りつけます。次の3つの方法があります。


- [Ctrl] + [V]キーを押す
- ツールバーの[貼り付け]ボタンをクリックする
- 編集メニューの[貼り付け]を選択する

5.5.11 プロパティの編集

プロパティを編集するには、まずフォームかコントロールをクリックして、プロパティグリッドにプロパティを表示させます。

- テキスト入力が必要なプロパティの場合
プロパティグリッドの変更したいプロパティをクリックします。新しい値を入力したあと、[Enter]キーを押すか別のプロパティを選択すると、変更が適用されます。

- ドロップダウンリストから選択するプロパティーの場合
プロパティーグリッドの変更したいプロパティーをクリックします。値の右にある▼をクリックし、ドロップダウンリストから新しい値を選択すると変更が適用されます。
- ボタンを選択するプロパティーの場合

プロパティーグリッドの変更したいプロパティーをクリックします。値の右にある参照  のボタンを選択します。新しい値をダイアログから設定します。

5.5.12 イベントハンドラーの使い方

イベントハンドラーは、イベント固有のパラメーターをもったSPEL+関数です。

イベントハンドラーは、カレントプロジェクトのどのプログラムファイルにも記述できます。

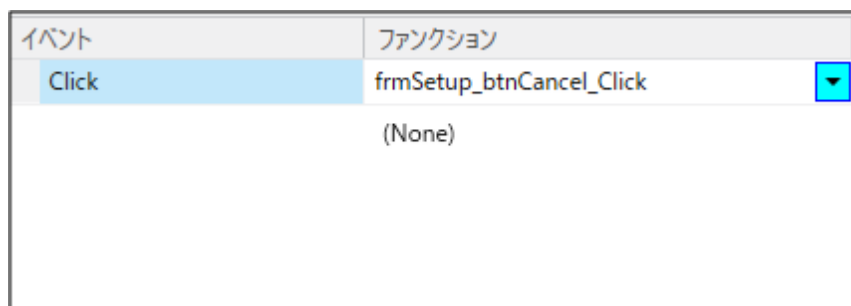
デフォルトでは、フォームを作成するためのプログラムの中に生成されます。例えば、ボタンコントロールのクリックイベントのハンドラーを作成する場合、もしプログラムファイルがなければ自動的に作成し、そのファイルの中に関数を作成します。

イベントハンドラーを作成します。次の3つの方法があります。

- フォームやコントロールをダブルクリックし、デフォルトのイベントハンドラーを作成します。例えば、ボタンコントロールをダブルクリックすると、クリックイベントのハンドラーの関数が生成されます。
- イベントグリッドで、イベント名をダブルクリックします。
- イベントグリッドで、イベントの値のドロップダウンリストから関数を選択します。正しいパラメーターで作成された関数が、ドロップダウンリストに表示されます。

イベントハンドラーを変更します。

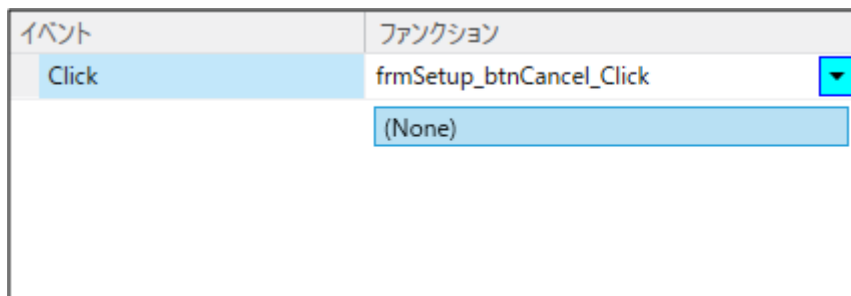
イベントのファンクションのドロップダウンリストから、イベントハンドラー関数を変更できます。



イベントハンドラーを無効にします。

イベントハンドラー作成後に、必要なくなったイベントハンドラーを無効にすることができます。

イベントハンドラーを無効にするには、[ファンクション]ドロップダウンリストから(None)を選択します。関数は削除されませんが、イベントが発生したときに呼び出されなくなります。



EventTaskTypeの使い方

イベントが発生した場合、イベントハンドラー関数がSPEL+のタスクとして起動されます。

EventTaskTypeプロパティーは、タスクの実行されるタイプを選択します。これは、一時停止状態や、非常停止状態でタスクを実行するためにはとても重要です。

例えば、PauseステートメントやContステートメントを実行するボタンのイベントハンドラーでは、

EventTaskTypeプロパティを1 - NoPauseに設定しなくてはなりません。また、非常停止中にResetステートメントを行うボタンのイベントハンドラーでは、EventTaskTypeを2 - NoEmgAbortに設定します。

5.5.13 変更の保存

GUI Builderウィンドウの変更を保存するには、次の3つの方法があります。

- [Ctrl]+[S]キーを押します。
- メニュー-[ファイル]-[ファイルの保存]を選択します。
- ツールバー-[ファイルの保存]ボタンをクリックします。

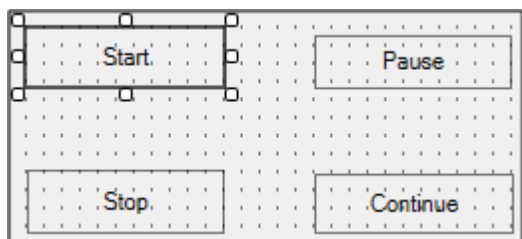
5.6 GUI Builderの設定

GUI Builderウィンドウのステータスバー上の[スナップグリッド]、[スナップライン]をクリックし操作します。GUI Builderウィンドウのフォーム上でのコントロールの配置方法を、GUI Builderの設定から変更することができます。

ステータスバーの[スナップグリッド]をクリックします。

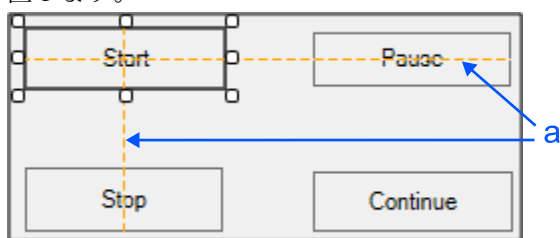


[スナップグリッド]をクリックした場合、GUI Builderは、グリッドを表示して、コントロールをグリッドの位置に配置します。



グリッドのサイズ(間隔)を変える場合は、[スナップグリッド]を右クリックし、[グリッド幅]の値を変更してください。

[スナップライン]をクリックした場合、GUI Builderは、ライン(a)を表示して、コントロールをラインに添って配置します。

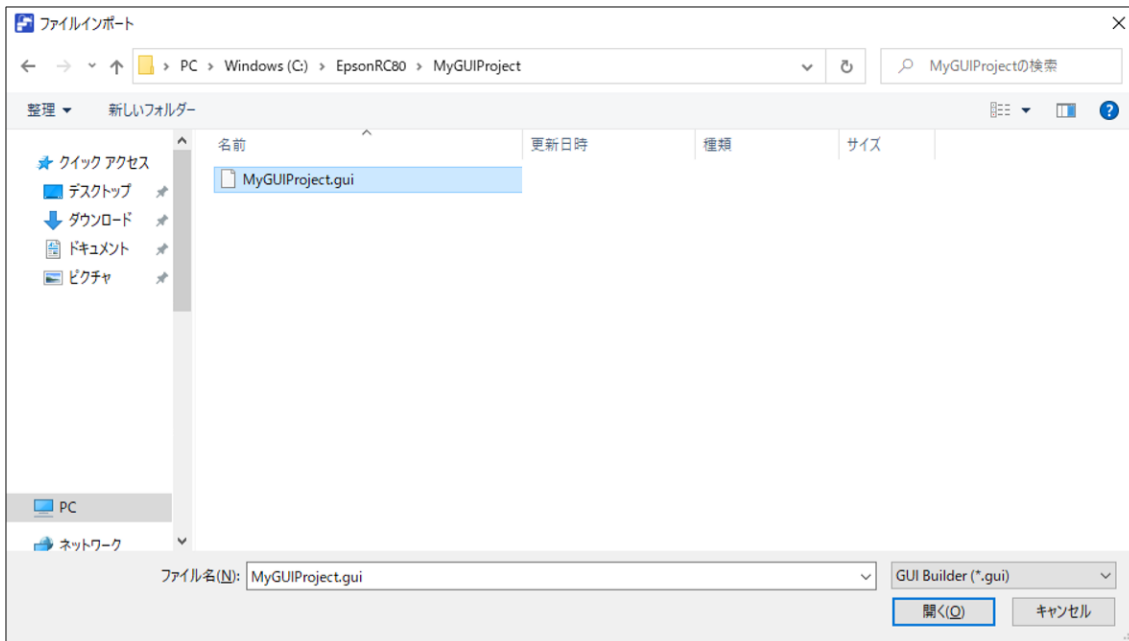


5.7 他のプロジェクトからフォームをインポートするには

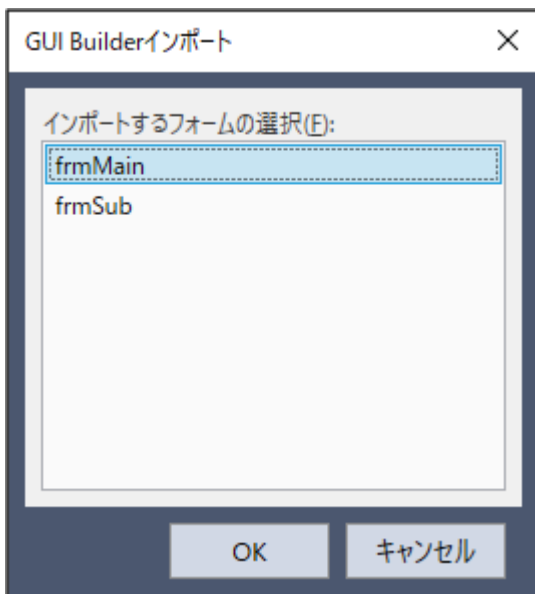
他のプロジェクトからGUIフォームをインポートする手順を以下に示します。

1. [ファイル]-[ファイルのインポート] もしくは、プロジェクトエクスプローラー上の[フォーム]を右クリックし、[フォームの追加]メニューの[既存のフォーム]を選択します。

2. インポート元のプロジェクトフォルダーを指定し、GUI Builderのファイル形式を選択します。



3. GUIのファイル名を選択し、[開く]ボタンをクリックします。
4. インポート元のプロジェクトが持っているフォームの一覧を表示するダイアログが表示されます。



5. インポートしたいフォームを選択し(複数選択可)、[OK]をクリックします。
6. カレントプロジェクトに選択したフォームが追加されます。選択したフォームがすでに追加されていた場合、フォームを上書きするかを、質問されます。

キーポイント

他プロジェクトで定義されたイベントハンドラーを使用する場合、フォームのプログラム[フォーム名.prg]をインポートする必要があります。

6. GUI Builderの構成要素

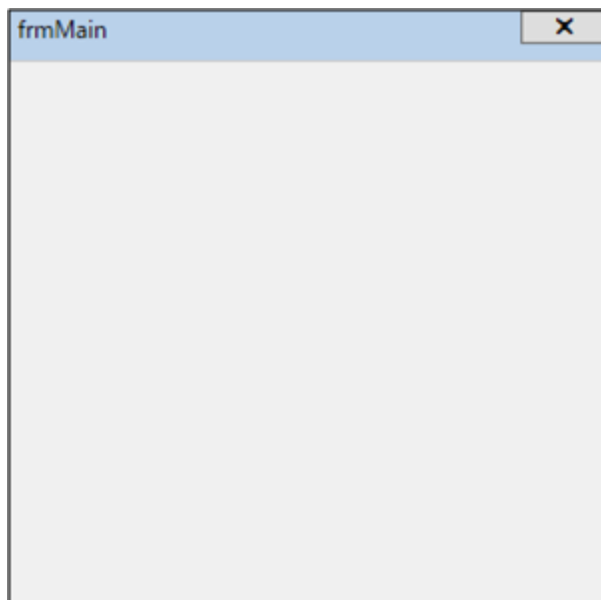
本章では、フォームと、フォームの上で使用可能なすべてのコントロールについて説明します。各コンポーネントの項では、使い方、プロパティ、イベントについて説明します。

プロパティ、イベント、ステートメントの詳細は、リファレンスの章を参照してください。

6.1 フォーム

6.1.1 解説

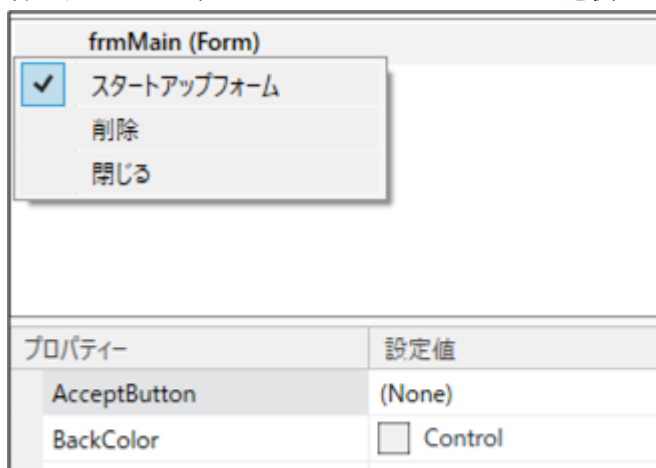
フォームはGUIアプリケーションの基本的な土台となります。オペレーター操作のために、コントロールが配置されたウィンドウやダイアログボックスを表示します。GUIアプリケーションは、1つまたは複数のフォームを持つことができます。



6.1.2 使い方

フォームを表示します。次の4つの方法でフォームを表示します。

1. フォームがスタートアップフォームに設定されている場合は、オペレーターモードでシステムを起動すると自動的に表示されます。スタートアップフォームに設定するためには、フォームエクスプローラーのフォームを右クリックして、“スタートアップフォーム”を選択してください。



スタートアップフォームは、太字で表示されます。また、メニュー-[プロジェクト]-[プロパティ]-[GUI Builder]を選択して、スタートアップフォームに設定することもできます。

2. GShowDialog関数を使用します。この関数は、フォームをダイアログボックスとして表示し、DialogResultの値を返します。

```
result = GShowDialog(frmSetSpeed)
```

通常、DialogResultの値は、フォームの上の[OK]ボタンや[キャンセル]ボタンなどで設定されます。

3. GShowステートメントを使用します。このステートメントは、フォームをウィンドウとして表示します。

```
GShow frmIODiags
```

4. Runウィンドウでドロップダウンリストから実行したいフォームを選択し、[開始]ボタンをクリックすると、選択されたフォームが表示されます。

フォームの設定

FormBorderStyleプロパティを設定します。ダイアログとして使うときは、3 - FixedDialogを設定します。フォームのタイトルバーの構成は、ControlBox、MaximizeBox、MinimizeBoxプロパティを設定します。フォームを表示するときの設定は、WindowStateプロパティに、Normal size、Maximized、Minimizedを選択してください。

ヘルプの使い方

HelpButtonプロパティをTrueに設定すると、独自のHelpファイルからヘルプトピックを表示させることができます。その時は、HelpIDプロパティに、ヘルプファイルのトピックIDを設定します。

詳細は以下を参照してください。

[ヘルプファイルの使い方](#)

6.1.3 フォームプロパティ

プロパティ	説明
AcceptButton	ボタン以外のコントロールにフォーカスがあるとき、[Enter]キーを押したときにクリックイベントを実行するボタンを設定します。 デフォルト: None
BackColor	フォームの背景色を設定します。 デフォルト: Control
CancelButton	キーボードの[Esc]キーを押したときに、クリックイベントを実行するボタンを設定します。実行後にフォームを閉じます。 デフォルト: None
ControlBox	タイトルバーのコントロールボックスを表示するかを設定します。 デフォルト: True
Controls	フォーム上のコントロール配列。
Count	コントロール配列上のコントロールの数を返します。
Dialog Result	フォームを閉じるときの戻り値を設定します。(SEPLプログラムからのみ有効)
EventTaskType	イベントで起動するタスクタイプを設定します。 デフォルト: Normal
FormBorderStyle	フォームのボーダースタイルを設定します。 デフォルト: FixedDialog
FormTemplate	フォームのサイズをテンプレートから選択、設定します。
Height	フォームの高さをピクセル数で設定します。
HelpButton	タイトルバーへ、ヘルプボタンを表示するかを設定します。 デフォルト: False

プロパティ	説明
HelpID	タイトルバーのヘルプボタンを押したときに表示するヘルプトピックIDを設定します。 デフォルト: 0
Icon	フォームのタイトルバーのアイコンに表示するイメージを設定・取得します。
Left	フォームの表示位置の左の座標をピクセル数で設定します。
MaximizeBox	タイトルバーに最大化ボタンを表示するかを設定します。 デフォルト: False
MinimizeBox	タイトルバーに最小化ボタンを表示するかを設定します。 デフォルト: False
Name	フォームの名前を設定します。
StartPosition	フォームの表示位置を設定します。 デフォルト: CenterScreen
Text	フォームの表示テキストを設定します。 デフォルト: フォームの名前
Top	フォームの表示位置の上の座標をピクセル数で設定/取得します。
Type	コントロールのタイプを返します。
Width	フォームの幅をピクセル数で設定します。
WindowState	フォームのウィンドウの状態を設定します。 デフォルト: Normal

6.1.4 フォームイベント

イベント	説明
Closed	フォームが閉じられたとき実行されます。
Closing	フォームを閉じる操作が行われたとき、フォームが閉じられる前に実行されます。
Load	フォームが読み込まれるとき実行されます。
Resize	フォームのサイズが変更されると実行されます。

6.2 ボタンコントロール

6.2.1 解説

ボタンコントロールは、オペレーターが何かを実行したいときにクリックするものです。ボタンコントロールには、テキストと画像の両方を表示させることができます。ボタンがクリックされると、ボタンが押されたように表示されます。



6.2.2 使い方

ボタンコントロールは、オペレーターが行う操作をクリック一つで実行します。

ForeColor(ボタンテキストの表示用)、BackColor、Font、TextAlign、Image、ImageAlignなどのプロパティを設定すると、ボタンの外観を変更できます。ボタンが押されたときにどのような動作を行うか、クリックイベント関数に記述してください。

6.2.3 ボタンコントロールのプロパティ

プロパティ	説明
BackColor	コントロールの背景色を設定します。 デフォルト: Control
BackColorMode	コントロールの背景色モードを設定します。 デフォルト: Visual Style
Enabled	コントロールを有効にするかを設定します。 デフォルト: True
EventTaskType	イベントで起動するタスクタイプを設定します。 デフォルト: 0 - Normal
Font	コントロールのテキストを表示するフォントを設定します。(フォント名、スタイル、サイズ) デフォルト: Microsoft Sans Serif 8.25 pt
FontBold	現在のフォントを太字にします。 デフォルト: False
FontItalic	現在のフォントをイタリックにします。 デフォルト: False
FontName	現在のフォントの名前を設定します。 デフォルト: Microsoft Sans Serif
FontSize	現在のフォントのサイズをポイントで設定します。 デフォルト: 8.25
ForeColor	コントロールのテキストの色を設定します。 デフォルト: ControlText
Height	コントロールの高さをピクセル数で設定します。
Image	コントロールに表示する画像ファイルを設定します。[Delete]キーで削除できます。 デフォルト: 空白
ImageAlign	表示する画像の配置を設定します。 デフォルト: MiddleCenter
Left	コントロールの表示位置の左の座標をピクセル数で設定します。
Name	コントロールの名前を設定します。 デフォルト: ButtonXX

プロパティ	説明
PressDelay	コントロールをクリックしイベントが実行されるまでの時間をmsecで設定します。設定した時間が経過する前にクリックを離すとイベントは実行されません。 デフォルト：即時イベント実行
PressSound	コントロールをクリックした際にビープ音を鳴動するかを設定します。 デフォルト：False - 鳴動しない
TabIndex	コントロールのタブインデックスを設定します。
Text	コントロールの表示テキストを設定します。 デフォルト：コントロールの名前
TextAlign	テキストの配置を設定します。 デフォルト:MiddleCenter
ToolTipText	コントロールのツールチップで表示されるテキストを設定します。 デフォルト：空白
Top	コントロールの表示位置の上の座標をピクセル数で設定します。
Visible	コントロールを表示するか、隠すかを設定します。 デフォルト：True
Width	コントロールの幅をピクセル数で設定します。

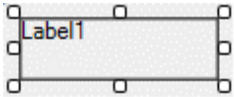
6.2.4 ボタンコントロールのイベント

イベント	説明
Click	コントロールがクリックされたときに実行されます。

6.3 ラベルコントロール

6.3.1 解説

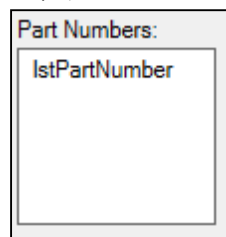
ラベルコントロールは、オペレーターが編集できないテキストやイメージを表示します。フォーム上のオブジェクトの説明や、実行時の情報表示、アプリケーションの状況などを表示します。



6.3.2 使い方

ラベルコントロールを使用して、フォームにテキストを表示します。
通常、ラベルコントロールは他のコントロールのテキストラベルに使用されます。例えば、フォームの上にリスト

ボックスコントロールを作成する場合、ラベルコントロールでリスト名をリストボックスの上に配置します。また、リストボックス名として、TextプロパティをPart Numbers:としたラベルコントロールの例を下に示します。



ラベルの外観の設定

ラベルの外観を変更するには、次のプロパティを使用します。

BorderStyle、ForeColor、BackColor、Font、Image、ImageAlign、TextAlign

ホットキーの使用(ニーモニック)

ラベルのTextプロパティで、ホットキーとなる文字を指定できます。

ラベルコントロールのTextプロパティの中で、ホットキーになる文字の前にアンパーサンド(&)を付けてください。

例えば、前の説明のラベルで“Part &Numbers:”のようにTextプロパティを設定します。

実行時に、[Alt]+[N]を押すと、フォーカスがリストボックスに移動します。

ホットキーを使用するとき、ラベルのタブインデックスが、コントロールのタブインデックスの一つ前となっていることを確認してください。

6.3.3 ラベルコントロールのプロパティ

プロパティ	説明
BackColor	コントロールの背景色を設定します。 デフォルト: Control
BorderStyle	コントロールのボーダースタイルを設定します。 デフォルト: None
Enabled	コントロールを有効にするかを設定します。 デフォルト: True
EventTaskType	イベントで起動される関数のタスクタイプを設定します。 デフォルト: Normal
Font	コントロールのテキストを表示するフォントを設定します(フォント名、スタイル、サイズ)。 デフォルト: Microsoft Sans Serif 8.25 pt
FontBold	現在のフォントをボールド表示します。 デフォルト: False
FontItalic	現在のフォントをイタリック表示します。 デフォルト: False
FontName	現在のフォントの名前を設定します。 デフォルト: Microsoft Sans Serif
FontSize	現在のフォントのサイズをポイントで設定します。 デフォルト: 8.25
ForeColor	コントロールのテキストの色を設定します。 デフォルト: ControlText

プロパティ	説明
Height	コントロールの高さをピクセル数で設定します。
Image	コントロールに表示する画像ファイルを設定します。[Delete]キーで削除できます。 デフォルト: 空白
ImageAlign	表示する画像の配置を設定します。 デフォルト: MiddleCenter
Left	コントロールの表示位置の左の座標をピクセル数で設定します。
Name	コントロールの名前を設定します。 デフォルト: LabelXX
TabIndex	コントロールのタブインデックスを設定します。
Text	コントロールの表示テキストを設定します。 デフォルト: コントロールの名前
TextAlign	テキストの配置を設定します。 デフォルト: TopLeft
ToolTipText	コントロールのツールチップで表示されるテキストを設定します。 デフォルト: 空白
Top	コントロールの表示位置の上の座標をピクセル数で設定します。
Visible	コントロールを表示するか、隠すかを設定します。 デフォルト: True
Width	コントロールの幅をピクセル数で設定します。

6.3.4 ラベルコントロールのイベント

イベント	説明
Click	コントロールがクリックされたときに実行されます。
DblClick	コントロールがダブルクリックされたときに実行されます。

6.4 テキストボックスコントロール

6.4.1 解説

テキストボックスコントロールは、オペレーターが入力したり、テキストを表示するために使われます。通常、テキストボックスは編集可能なテキストのために使われます。(書き換え不可にすることも可能です。)

テキストボックスは、複数行のテキストを表示したり、コントロールのサイズに合わせてテキストを改行します。テキストボックス内のフォーマットは1つのみ設定できます。一部の文字を太字にするなどはできません。



6.4.2 使い方

テキストボックスは、1行または複数行のモードで使用します。

1行のテキストボックス

デフォルトでは、テキストボックスは1行モードで作成されます。コントロールの幅のみ変更可能です。オペレーターは、文字を1行入力することができます。

複数行のテキストボックス

複数行モードで使うときは、MultilineプロパティをTrueに設定します。このモードでは、コントロールの高さと幅を変更できます。Scrollbarsプロパティの設定によって、スクロールバーを表示することもできます。水平スクロールバーを表示するには、WordWrapプロパティをFalseに設定します。

テキストボックスの外観の設定

テキストボックスの外観を変更するには、次のプロパティを使用します。

BorderStyle、ForeColor、BackColor、Font、TextAlign

SPEL+のグローバル変数 (配列を除く)の表示方法

Variableプロパティを設定することによって、自動的にSPEL+グローバル変数の値を表示することができます。

注意

Variable プロパティ設定でSPEL+のグローバル変数が表示されない場合、プロジェクトをビルドしてください。

6.4.3 テキストボックスコントロールのプロパティ

プロパティ	説明
AppendText	テキストボックスにテキストを追加します。(SPELプログラムからのみ有効)
BackColor	コントロールの背景色を設定します。 デフォルト: Window
BorderStyle	コントロールのボーダースタイルを設定します。 デフォルト: Fixed3D
Enabled	コントロールを有効にするかを設定します。 デフォルト: True
EventTaskType	イベントで起動するタスクタイプを設定します。 デフォルト: Normal
Font	コントロールのテキストを表示するフォントを設定します。 デフォルト: Microsoft Sans Serif 8.25 pt
FontBold	現在のフォントを太字にします。 デフォルト: False
FontItalic	現在のフォントをイタリック表示します。 デフォルト: False
FontName	現在のフォントの名前を設定します。 デフォルト: Microsoft Sans Serif
FontSize	現在のフォントのサイズをポイントで設定します。 デフォルト: 8.25

ForeColor	コントロールのテキストの色を設定します。 デフォルト: WindowText
Height	コントロールの高さをピクセル数で設定します。
Left	コントロールの表示位置の左の座標をピクセル数で設定します。
Multiline	複数行モードにするかを設定します。 デフォルト: False
Name	コントロールの名前を設定します。 デフォルト: TextBoxXX
PasswordChar	入力された文字を隠すための文字を設定します。 デフォルト: 空白
ReadOnly	オペレーターにテキストを編集させない場合に設定します。 デフォルト: False
ScrollBars	スクロールバーを表示するときに設定します。 デフォルト: None
ShowPrint	Printステートメントの出力を表示するかを設定します。 デフォルト: False
TabIndex	コントロールのタブインデックスを設定します。
Text	コントロールの表示テキストを設定します。 デフォルト: 空白
TextAlign	テキストの配置を設定します。 デフォルト: Left
ToolTipText	コントロールのツールチップで表示されるテキストを設定します。 デフォルト: 空白
Top	コントロールの表示位置の上の座標をピクセル数で設定します。
Update	テキストボックスコントロールの表示を更新します。
Variable	表示させたいSPEL+のグローバル変数 (配列を除く)を設定します。 デフォルト: None
Visible	コントロールを表示するか、隠すかを設定します。 デフォルト: True
Width	コントロールの幅をピクセル数で設定します。
WordWrap	テキストを改行して表示するかどうかを設定します。 デフォルト: True

6.4.4 テキストボックスコントロールのイベント

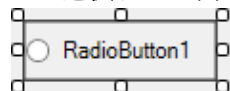
イベント	説明
KeyPress	コントロールにフォーカスがあり、さらにキーが押された場合に実行されます。

イベント	説明
Click	コントロールがクリックされたときに実行されます

6.5 ラジオボタンコントロール

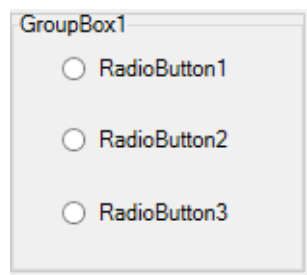
6.5.1 解説

ラジオボタンコントロールは、オペレーターが2つまたはそれ以上の選択操作を行うためのコントロールです。ラジオボタンとチェックボックスは同じような機能に見えますが、大きな違いがあります。ラジオボタンの場合、一つ選択すると同じグループの他のラジオボタンは選択できなくなります。



6.5.2 使い方

ラジオボタンコントロールは、グループボックスコントロールとともに使用します。オペレーターがラジオボタンの1つをクリックすると、他のボタンは選択が解除されます。



オペレーターがラジオボタンをクリックしたことは、クリックイベントで判断できます。ラジオボタンがチェックされているかどうかは、Checkedプロパティで判断できます。

各ラジオボタンにイベントハンドラーを作成するより、グループのラジオボタンすべてにひとつのイベントハンドラーを作成するほうが簡単な場合があります。

イベントハンドラーのSender\$パラメーターを使うと、どのラジオボタンがクリックされたかを判断することができます。Sender\$はイベントを送ったコントロールの名前が設定されています。

```
Function frmSetup_OptionsClick(Sender$ As String)
    Boolean checked
    GGet frmSetup.Sender$.Checked, checked
    If checked Then
        Select Sender$
            Case "RadioButton1"
                g_Option1 = True
            Case "RadioButton2"
                g_Option2 = True
        Send
    EndIf
End
Fend
```

ラジオボタンの外観の設定

ラジオボタンの外観を変更するには、次のプロパティを使用します。

BorderStyle、ForeColor、BackColor、Font、Image、ImageAlign、TextAlign

6.5.3 ラジオボタンコントロールのプロパティ

プロパティ	説明
BackColor	コントロールの背景色を設定します。 デフォルト: Control
Checked	ラジオボタンの選択状態の初期値を設定します。 デフォルト: False
Enabled	ラジオボタンを有効にするかどうかを設定します。 デフォルト: True
EventTaskType	イベントで起動するタスクタイプを設定します。 デフォルト: Normal
Font	コントロールのテキストを表示するフォントを設定します。 デフォルト: Microsoft Sans Serif 8.25 pt
FontBold	現在のフォントをボールド表示します。 デフォルト: False
FontItalic	現在のフォントをイタリック表示します。 デフォルト: False
FontName	現在のフォントの名前を設定します。 デフォルト: Microsoft Sans Serif
FontSize	現在のフォントのサイズをポイントで設定します。 デフォルト: 8.25
ForeColor	コントロールのテキストの色を設定します。 デフォルト: ControlText
Height	コントロールの高さをピクセル数で設定します。
Image	コントロールに表示する画像ファイルを設定します。[Delete]キーで削除できます。 デフォルト: 空白
ImageAlign	表示する画像の配置を設定します。 デフォルト: MiddleCenter
Left	コントロールの表示位置の左の座標をピクセル数で設定します。
Name	コントロールの名前を設定します。 デフォルト: RadioButtonXX
TabIndex	コントロールのタブインデックスを設定します。
Text	コントロールの表示テキストを設定します。 デフォルト: コントロールの名前
TextAlign	テキストのアライメント方法を設定します。 デフォルト: Middle Left
ToolTipText	コントロールのツールチップで表示されるテキストを設定します。 デフォルト: 空白

プロパティ	説明
Top	コントロールの表示位置の上の座標をピクセル数で設定します。
Visible	コントロールを表示するかどうかを設定します。 デフォルト: True
Width	コントロールの幅をピクセル数で設定します。

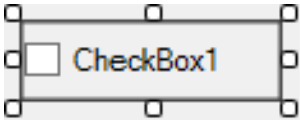
6.5.4 ラジオボタンコントロールのイベント

イベント	説明
Click	コントロールがクリックされたときに実行されます。

6.6 チェックボックスコントロール

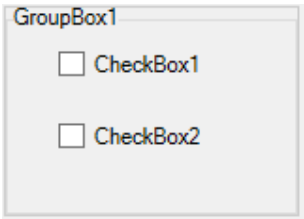
6.6.1 解説

チェックボックスコントロールは、オンかオフの状態を入力するコントロールです。主に、はい／いいえ、True／Falseを選択するのに使われます。
チェックボックスコントロールをグループ化して、1つまたは複数を選択するためのコントロールにすることもできます。
ラジオボタンコントロールと似ていますが、チェックボックスコントロールでは複数のボタンの選択が可能です。



6.6.2 使い方

チェックボックスコントロールは、各設定のオン／オフ操作を行えます。
通常、チェックボックスコントロールはグループボックスコントロールに追加していきます。
Checkedプロパティでチェックボックスがチェックされているかを確認できます。



チェックボックスの外観の設定

チェックボックスの外観を変更するには、次のプロパティを使います。
BorderStyle、ForeColor、BackColor、Font、Image、ImageAlign、TextAlign

6.6.3 チェックボックスコントロールのプロパティ

プロパティ	説明
BackColor	コントロールの背景色を設定します。 デフォルト: Control

プロパティ	説明
Checked	チェックボックスの選択状態の初期値を設定します。 デフォルト:False
Enabled	コントロールを有効にするかどうかを設定します。 デフォルト:True
EventTaskType	イベントで起動するタスクタイプを設定します。 デフォルト:Normal
Font	コントロールのテキストを表示するフォントを設定します。 デフォルト: Microsoft Sans Serif 8.25 pt
FontBold	現在のフォントを太字にします。 デフォルト:False
FontItalic	現在のフォントをイタリックにします。 デフォルト:False
FontName	現在のフォントの名前を設定します。 デフォルト: Microsoft Sans Serif
FontSize	現在のフォントのサイズをポイントで設定します。 デフォルト: 8.25
ForeColor	コントロールのテキストの色を設定します。 デフォルト: ControlText
Height	コントロールの高さをピクセル数で設定します。
Image	コントロールに表示する画像ファイルを設定します。[Delete]キーで削除できます。 デフォルト: 空白
ImageAlign	表示する画像の配置を設定します。 デフォルト:5 -MiddleCenter
Left	コントロールの表示位置の左の座標をピクセル数で設定します。
Name	コントロールの名前を設定します。 デフォルト: CheckBoxXX
TabIndex	コントロールのタブインデックスを設定します。
Text	コントロールの表示テキストを設定します。 デフォルト: コントロールの名前
TextAlign	テキストの配置を設定します。 デフォルト:MiddleLeft
ToolTipText	コントロールのツールチップで表示されるテキストを設定します。 デフォルト: 空白
Top	コントロールの表示位置の上の座標をピクセル数で設定します。
Visible	コントロールを表示するかどうかを設定します。 デフォルト:True

プロパティ	説明
Width	コントロールの幅をピクセル数で設定します。

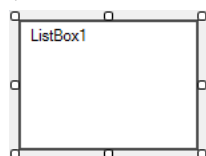
6.6.4 チェックボックスのコントロールのイベント

イベント	説明
Click	コントロールがクリックされたときに実行されます。

6.7 リストボックスコントロール

6.7.1 説明

リストボックスコントロールは、オペレーターが選択する1つまたは複数のアイテムを表示するコントロールです。



6.7.2 使い方

リストボックスコントロールを使って、オペレーターが選択できるアイテムのリストを作成します。

リストボックスにアイテムを追加

AddItemプロパティを使用して、リストボックスにアイテムを加えます。通常は、フォームのロードイベントで行います。

```
GSet frmSetup.lstModels.AddItem, "Model1"
GSet frmSetup.lstModels.AddItem, "Model2"
GSet frmSetup.lstModels.AddItem, "Model3"
```

SortedプロパティをTrueに設定すると、アイテムをソート(整列)することができます。

選択されたアイテムの確認

SelectedIndexプロパティを使用して、オペレーターが選択したアイテムを確認できます。どのアイテムも選択されていないければ、SelectedIndexプロパティの値は-1です。

```
Integer index
GGet frmSetup.lstModels.SelectedIndex, index
```

リスト配列

List Arrayプロパティを使用すると、リストの中のアイテムのすべてにアクセスが可能です。

```
Integer i, count
String item$
GGet frmSetup.lstModels.ListCount, count
For i = 0 To count- 1
    GGet frmSetup.lstModels.List(i), item$
Next i
```

Listプロパティに空の文字列を設定することで、アイテムを削除できます。

```
GSet frmSetup.lstModels.List(0), ""
```

リストボックスの外観の設定

リストボックスの概観を変更するには、次のプロパティを使います。BorderStyle、ForeColor、BackColor、Font

6.7.3 リストボックスコントロールのプロパティ

プロパティ	説明
AddItem	リストボックスにアイテムを追加します。(SPELプログラムからのみ有効)
BackColor	コントロールの背景色を設定します。 デフォルト: Window
BorderStyle	コントロールのボーダーのスタイルを設定します。 デフォルト: Fixed3D
Enabled	コントロールを有効にするかを設定します。 デフォルト: True
EventTaskType	イベントで起動するタスクタイプを設定します。 デフォルト: Normal
Font	コントロールのテキストを表示するフォントを設定します。() デフォルト: Microsoft Sans Serif 8.25 pt
FontBold	現在のフォントを太字にします。 デフォルト: False
FontItalic	現在のフォントをイタリックにします。 デフォルト: False
FontName	現在のフォントの名前を設定します。 デフォルト: Microsoft Sans Serif
FontSize	現在のフォントのサイズをポイントで設定します。 デフォルト: 8.25
ForeColor	コントロールのテキストの色を設定します。 デフォルト: WindowText
Height	コントロールの高さをピクセル数で設定します。
Left	コントロールの表示位置の左の座標をピクセル数で設定します。
List	リストボックスのアイテムにアクセスします。(SPELプログラムからのみ有効)
ListCount	アイテムの数を取得します。
Name	コントロールの名前を設定します。 デフォルト: ListBoxXX
SelectedIndex	選択されたアイテムの番号を取得します。(SPELプログラムからのみ有効)
Sorted	リストのアイテムをソートするかを設定します。 デフォルト: False
TabIndex	コントロールのタブインデックスを設定します。

プロパティ	説明
ToolTipText	コントロールのツールチップで表示されるテキストを設定します。 デフォルト: 空白
Top	コントロールの表示位置の上の座標をピクセル数で設定します。
Visible	コントロールを表示するか、隠すかを設定します。 デフォルト: True
Width	コントロールの幅をピクセル数で設定します。

6.7.4 リストボックスコントロールのイベント

イベント	説明
Click	コントロールがクリックされたときに実行されます。
DblClick	コントロールがダブルクリックされたときに実行されます。

6.8 コンボボックスコントロール

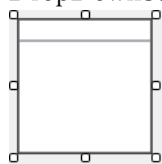
6.8.1 解説

コンボボックスコントロールは、データをドロップダウンコンボボックスに表示します。デフォルトでは2つの部分に分かれます。上部分はテキストボックスで、リストのアイテムを入力できます。下部分はリストボックスで、オペレーターは表示されているアイテムから1つを選択できます。

6.8.2 使い方

DropDownStyleプロパティの値によって、コンボボックスコントロールは3種類のモードを持ちます。

DropDownStyle = Simple



Simpleモードでは、リストボックスの上にテキストボックスがつながっています。オペレーターは、テキストボックスに入力するか、リストから選択します。

DropDownStyle = DropDown



DropDownモードでは、テキストボックスの右側の▼をクリックするとリストボックスが表示されます。テキストボックスで、選択されたテキストを編集することができます。

DropDownStyle = DropDownList



DropDownListモードでは、テキストボックス内のテキスト編集ができません。リストにあるアイテムのみ選択可能です。

コンボボックスコントロールのリストボックス部分にアイテムを追加する方法は、リストボックスのAddItem、List、ListCount、SelectedIndexプロパティの説明を参照してください。

コンボボックスの外観の設定

ForeColor、BackColor、Fontプロパティを使って、コンボボックスの外観を変更します。

6.8.3 コンボボックスコントロールのプロパティ

プロパティ	説明
AddItem	コンボボックスにアイテムを追加します。(SPELプログラムからのみ有効)
BackColor	コントロールの背景色を設定します。 デフォルト: Window
DropDownStyle	コンボボックスのドロップダウンのモードを設定します。 デフォルト: DropDown
Enabled	コントロールを有効にするかを設定します。 デフォルト: True
EventTaskType	イベントで起動するタスクタイプを設定します。 デフォルト: Normal
Font	コントロールのテキストを表示するフォントを設定します。() デフォルト: Microsoft Sans Serif 8.25 pt
FontBold	現在のフォントを太字にします。 デフォルト: False
FontItalic	現在のフォントをイタリックにします。 デフォルト: False
FontName	現在のフォントの名前を設定します。 デフォルト: Microsoft Sans Serif
FontSize	現在のフォントのサイズをポイントで設定します。 デフォルト: 8.25
ForeColor	コントロールのテキストの色を設定します。 デフォルト: WindowText
Height	コントロールの高さをピクセル数で設定します。
Left	コントロールの表示位置の左の座標をピクセル数で設定します。
List	リストボックスのアイテムにアクセスします。(SPELプログラムからのみ有効)
ListCount	アイテムの数を取得します。(SPELプログラムからのみ有効)
Name	コントロールの名前を設定します。 デフォルト: ComboBoxXX
SelectedIndex	選択されたアイテムの番号を取得します。(SPELプログラムからのみ有効)
Sorted	リストのアイテムをソートするかを設定します。 デフォルト: False

TabIndex	コントロールのタブインデックスを設定します。
Text	コントロールの表示テキストを設定します。 デフォルト: 空白
ToolTipText	コントロールのツールチップで表示されるテキストを設定します。 デフォルト: 空白
Top	コントロールの表示位置の上の座標をピクセル数で設定します。
Visible	コントロールを表示するか、隠すかを設定します。 デフォルト: True
Width	コントロールの幅をピクセル数で設定します。

6.8.4 コンボボックスコントロールのイベント

イベント	説明
Click	コントロールがクリックされたときに実行されるイベントです。

6.9 ピクチャーボックスコントロール

6.9.1 解説

ピクチャーボックスコントロールは、ウィンドウズビットマップ、GIF、JPEG、PNG、アイコンを表示します。



6.9.2 使い方

ピクチャーボックスコントロールを使って、選択したイメージファイルを表示できます。
イメージを選択するのは、設計時または実行時どちらでも可能です。

イメージの設定

設計時にはImageプロパティを使用し、イメージリスト画面でイメージファイルを追加します。ImageIndexプロパティを使用し、追加したイメージファイルから表示したいイメージファイルの番号を選択することができます。
実行時にイメージファイルを設定するにはImageプロパティにイメージファイルのフルパスの指定が必要です。
またRC800コントローラーに接続して実行している場合は、ImageIndexプロパティの値を変更することで表示するイメージを切り替えることができます。

イメージのサイズ変更

SizeModeプロパティを使ってイメージのサイズを変更することが可能です。

ピクチャーボックスの外観の設定

BackColor、BorderStyle、SizeModeを使ってピクチャーボックスの外観を変更します。

6.9.3 ピクチャーボックスコントロールのプロパティ

プロパティ	説明
BackColor	コントロールの背景色を設定します。 デフォルト: Control

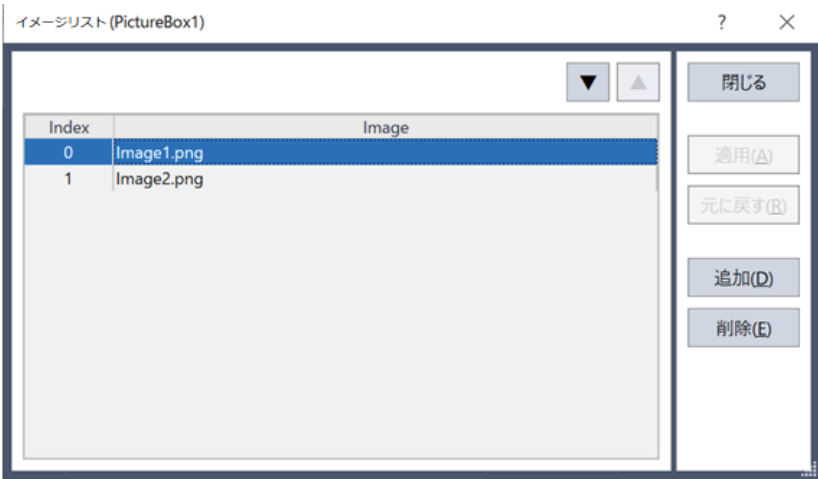
プロパティ	説明
BorderStyle	コントロールのボーダーのスタイルを設定します。 デフォルト:None
Enabled	コントロールを有効にするかを設定します。 デフォルト: True
EventTaskType	イベントで起動するタスクタイプを設定します。 デフォルト:Normal
Height	コントロールの高さをピクセル数で設定します。
Image	コントロールに表示する画像ファイルを設定します。 設計時にイメージリスト画面を表示して、画像ファイルの追加、削除、並び替えができます。
ImageIndex	イメージリスト画面で追加した画像ファイルからコントロールに表示する画像ファイルを選択します。
Left	コントロールの表示位置の左の座標をピクセル数で設定します。
Name	コントロールの名前を設定します。 デフォルト: PictureBoxXX
SizeMode	イメージのサイズの変更方法を設定します。 デフォルト:Normal
TabIndex	コントロールのタブインデックスを設定します。
ToolTipText	コントロールのツールチップで表示されるテキストを設定します。 デフォルト: 空白
Top	コントロールの表示位置の上の座標をピクセル数で設定します。
Visible	コントロールを表示するか、隠すかを設定します。 デフォルト: True
Width	コントロールの幅をピクセル数で設定します。

6.9.4 ピクチャーボックスコントロールのイベント

イベント	説明
Click	コントロールがクリックされたときに実行されます。
DbClick	コントロールがダブルクリックされたときに実行されます。

6.9.5 イメージリスト画面

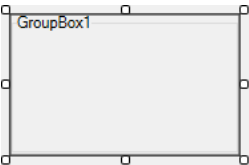
イメージリスト画面では、ユーザーがピクチャーボックスに表示したいイメージファイルの追加や削除および並び替えを行うことができます。イメージリスト画面はImageプロパティから表示することができます。



6.10 グループボックスコントロール

6.10.1 解説

グループボックスコントロールは、アイテムをグループ化して他のコントロールと分けます。コントロールを機能ごとに分類する場合などに使用されます。グループボックスでコントロールを分類すると、視覚的に分かりやすくなります。



6.10.2 使い方

グループボックスコントロールを使って、選択項目をラジオボタンやチェックボックスごとに分類してください。他のコントロールとともに分類することもできます。
まず、フォーム上にグループボックスを配置します。
次に、グループボックス上に新しいコントロールを作成するか、既存のコントロールをグループボックスの上にドラッグします。
グループボックスのTextプロパティにグループの名前を設定します。

グループボックスの外観の設定

BackColor、ForeColor、Fontプロパティを使い、グループボックスの外観を変更します。

6.10.3 グループボックスコントロールのプロパティ

プロパティ	説明
BackColor	コントロールの背景色を設定します。 デフォルト: Control
Enabled	コントロールを有効にするかを設定します。 デフォルト: True
EventTaskType	イベントで起動するタスクタイプを設定します。 デフォルト: 0 - Normal

プロパティ	説明
Font	コントロールのテキストを表示するフォントを設定します。() デフォルト: Microsoft Sans Serif 8.25 pt
FontBold	現在のフォントを太字にします。 デフォルト: False
FontItalic	現在のフォントをイタリックにします。 デフォルト: False
FontName	現在のフォントの名前を設定します。
FontSize	現在のフォントのサイズをポイントで設定します。 デフォルト: 8.25
ForeColor	コントロールのテキストの色を設定します。 デフォルト: ControlText
Height	コントロールの高さをピクセル数で設定します。
Left	コントロールの表示位置の左の座標をピクセル数で設定します。
Name	コントロールの名前を設定します。 デフォルト: GroupBoxXX
TabIndex	コントロールのタブインデックスを設定します。
Text	コントロールの表示テキストを設定します。 デフォルト: コントロールの名前
ToolTipText	コントロールのツールチップで表示されるテキストを設定します。 デフォルト: 空白
Top	コントロールの表示位置の上の座標をピクセル数で設定します。
Visible	コントロールを表示するか、隠すかを設定します。 デフォルト: True
Width	コントロールの幅をピクセル数で設定します。

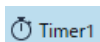
6.10.4 グループボックスコントロールのイベント

イベント	説明
Click	コントロールがクリックされたときに実行されます。

6.11 タイマーコントロール

6.11.1 解説

タイマーコントロールは、一定間隔でイベントを実行します。非常停止や他の通常タスクが一時停止しているときも実行します。



6.11.2 使い方

タイマーコントロールを使って、定期的にSPEL+プログラムを実行できます。
例えば、タイマーコントロールを使って2秒毎にステータスラベルを更新できます。

タイマーコントロールの使い方

1. フォーム上にタイマーコントロールを追加します。このコントロールは、実行時には見えません。コントロールはデザインエリアのフォームの下に表示されます。
2. 実行する時間の間隔を、Intervalプロパティにミリ秒単位で設定します。
3. タイマーイベントを常に実行する場合は、EnabledプロパティをTrueに設定します。または、実行時に、EnabledプロパティをTrueに設定します。
4. Tickイベントにイベントハンドラーを追加して、SPEL+プログラムを実行します。

キーポイント

Tickイベントハンドラーの実行中は、同じタイマーコントロールからのTickイベントは無視されます。

6.11.3 タイマーコントロールのプロパティ

プロパティ	説明
Enabled	コントロールを有効にするかを設定します。 デフォルト: False
Interval	イベントを実行する間隔を、ミリ秒単位で設定します。 デフォルト: 100
Name	コントロールの名前を設定します。 デフォルト: TimerXX

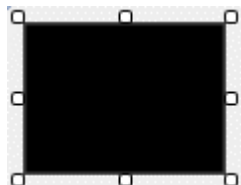
6.11.4 タイマーコントロールのイベント

イベント	説明
Tick	設定された時間間隔に達したときにイベントが実行されます。

6.12 ビデオボックスコントロール

6.12.1 解説

ビデオボックスコントロールは、フォームにVisionGuideオプションの画像を表示します。



6.12.2 使い方

フォームの上にビデオボックスコントロールを使って、簡単にアプリケーションにビデオウィンドウを表示できます。ビジョンシーケンスを実行する場合に、シーケンス結果をウィンドウに表示できます。

以下の手順で、ビジョンウィンドウを作成します。

1. フォーム上の表示したい場所に、ビデオボックスコントロールを配置します。
コントロールのサイズは、フルサイズまで変更が可能です。
2. VideoEnabledプロパティをTrueに設定します。
3. 画像処理の経過を表示する場合は、GraphicsEnabledプロパティをTrueに設定します。
4. Cameraプロパティのデフォルト値は0です。
画像処理シーケンス実行時に、カメラ画像を表示します。Cameraプロパティにカメラの番号を設定した場合、そのカメラの処理結果を表示します。

カメラ画像は、ビデオボックスコントロールのサイズに合わせて、自動的にスケーリングされます。ビデオボックスコントロールの高さや幅を変更するとき、その縦横比は一定に維持されます。

ビデオボックスコントロールの外観の設定

BorderStyleプロパティを使って、ビデオボックスコントロールの外観を変更します。

6.12.3 ビデオボックスコントロールのプロパティ

プロパティ	説明
BorderStyle	コントロールのボーダーのスタイルを設定します。 デフォルト:None
Camera	画像を表示するカメラの番号を設定します。 デフォルト:0
Enabled	コントロールを有効にするかを設定します。 デフォルト:True
EventTaskType	イベントで起動するタスクタイプを設定します。 デフォルト:Normal
GraphicsEnabled	画像処理結果を表示するかを設定します。 デフォルト:False
Height	コントロールの高さをピクセル数で設定します。
Left	コントロールの表示位置の左の座標をピクセル数で設定します。
Name	コントロールの名前を設定します。 デフォルト: VideoBoxXX
TabIndex	コントロールのタブインデックスを設定します。
ToolTipText	コントロールのツールチップで表示されるテキストを設定します。
Top	コントロールの表示位置の上の座標をピクセル数で設定します。
VideoEnabled	画像を表示させるかを設定します。 デフォルト: False

プロパティ	説明
Visible	コントロールを表示するか、隠すかを設定します。 デフォルト: True
Width	コントロールの幅をピクセル数で設定します。

6.12.4 ビデオボックスコントロールのイベント

イベント	説明
Click	コントロールがクリックされたときに実行されます。
DblClick	コントロールがダブルクリックされたときに実行されます。

6.13 LEDコントロール

6.13.1 解説

LEDコントロールは、I/Oステータス表示に使用します。



6.13.2 使い方

LEDコントロールを使用します。

1. Textプロパティにステータスの名前を設定します。
2. IOTypeプロパティを設定します。
input、output、memoryI/O(入力、出力、メモリーI/O)から選択できます。モニターするI/Oの入出力タイプを設定します。
3. IOBitプロパティを設定します。ステータスを表示するビット番号を設定します。

出力の場合に、AllowStateChangeプロパティをTrueに設定すると、コントロールをダブルクリックしたときに出力を反転させることができます。

LEDコントロールの外観の設定

LEDコントロールの外観を変更するには、次のプロパティを使います。

BackColor、BorderStyle、ForeColor、Font、ImageAlign、TextAlign

また、ImageOn、ImageOffプロパティを使って、LED表示が組み込まれたカラーイメージから選択したり、ユーザー独自のイメージを表示することができます。

6.13.3 LEDコントロールのプロパティ

プロパティ	説明
AllowStateChange	LEDをダブルクリックしたとき、出力を反転させる動作を許可するかを設定します。 デフォルト: False
BackColor	コントロールの背景色を設定します。 デフォルト: Control

プロパティ	説明
BorderStyle	コントロールのボーダーのスタイルを設定します。 デフォルト:None
Enabled	コントロールを有効にするかを設定します。 デフォルト: True
EventTaskType	イベントで起動するタスクタイプを設定します。 デフォルト:Normal
Font	コントロールのテキストを表示するフォントを設定します。() デフォルト: Microsoft Sans Serif 8.25 pt
FontBold	現在のフォントを太字にします。 デフォルト: False
FontItalic	現在のフォントをイタリック表示します。 デフォルト: False
FontName	現在のフォントの名前を設定します。 デフォルト: Microsoft Sans Serif
FontSize	現在のフォントのサイズをポイントで設定します。 デフォルト: 8.25
ForeColor	コントロールのテキストの色を設定します。 デフォルト: ControlText
Height	コントロールの高さをピクセル数で設定します。
ImageAlign	コントロールに表示する画像の配置を設定します。 デフォルト: MiddleLeft
ImageOff	I/Oがオフのときの表示を設定します。 デフォルト: LedOff.ico
ImageOn	I/Oがオンのときの表示を設定します。 デフォルト: LedRed.ico
IOBit	範囲: 0~9999 デフォルト: 0
IOType	モニターするI/Oの入出力タイプを設定します。 デフォルト: Input
Left	コントロールの表示位置の左の座標をピクセル数で設定します。
Name	コントロールの名前を設定します。 デフォルト: LedXX
TabIndex	コントロールのタブインデックスを設定します。
Text	コントロールの表示テキストを設定します。 デフォルト: コントロールの名前
TextAlign	テキストの配置を設定します。 デフォルト: MiddleRight

プロパティ	説明
ToolTipText	コントロールのツールチップで表示されるテキストを設定します。 デフォルト: 空白
Top	コントロールの表示位置の上の座標をピクセル数で設定します。
Visible	コントロールを表示するか、隠すかを設定します。 デフォルト: True
Width	コントロールの幅をピクセル数で設定します。

6.13.4 LEDコントロールのイベント

イベント	説明
DblClick	コントロールがダブルクリックされたときに実行されます。

6.14 ステータスバーコントロール

6.14.1 解説

ステータスバーコントロールは、テキストパネルの表示や、オプションパネル(日時, 非常停止状態, 安全扉状態, ロボット情報)の表示に使用します。



6.14.2 使い方

ステータスバーコントロールを使用して、任意のテキストや、その他の組み込みステータスを表示します。テキストパネルは常に表示されます。

Textプロパティを使用して、テキストパネルに表示する文字列を設定します。他のステータスパネル(日時, 非常停止状態, ロボット情報, 安全扉状態)はデフォルトでは表示されません。

ShowDateTime, ShowEStop, ShowRobot, およびShowSafeguardプロパティを使用して、これらのパネルを表示または非表示にします。これらのパネルは、システムによって自動的に更新されます。ステータスバーは、1つのフォームに対して1つのみ使用することができます。すでにフォーム内で1つのステータスバーが使用されている場合、ツールバー上の[新規ステータスバー]ボタンはグレイアウトされ、クリックできなくなります。

6.14.3 ステータスバーコントロールのプロパティ

プロパティ	説明
BackColor	コントロールの背景色を設定します。 デフォルト: Control
BorderStyle	コントロールのボーダーのスタイルを設定します。 デフォルト: None
Enabled	コントロールを有効にするかを設定します。 デフォルト: True

プロパティ	説明
EventTaskType	イベントで起動するタスクタイプを設定します。 デフォルト: Normal
Font	コントロールのテキストを表示するフォントを設定します。() デフォルト: Microsoft Sans Serif 8.25 pt
ForeColor	コントロールのテキストの色を設定します。 デフォルト: ControlText
Height	コントロールの高さをピクセル数で設定します。
Left	コントロールの表示位置の左の座標をピクセル数で設定します。
Name	コントロールの名前を設定します。 デフォルト: StatusBarXX
RobotNumber	ShowRobotに使用するロボット番号を設定します。
ShowDateTime	現在日時表示の有無を設定します。 デフォルト: False
ShowEStop	EStop状態表示の有無を設定します。 デフォルト: False
ShowRobot	RobotNumberで指定したロボットの表示有無を設定します。 デフォルト: False
ShowSafeguard	安全扉状態表示の有無を設定します。 デフォルト: False
TabIndex	コントロールのタブインデックスを設定します。
Text	コントロールの表示テキストを設定します。 デフォルト: コントロールの名前
ToolTipText	コントロールのツールチップで表示されるテキストを設定します。 デフォルト: 空白
Top	コントロールの表示位置の上の座標をピクセル数で設定します。
Visible	コントロールを表示するか、隠すかを設定します。 デフォルト: True
Width	コントロールの幅をピクセル数で設定します。

6.14.4 ステータスバーコントロールのイベント

イベント	説明
Click	コントロールがクリックされたときに実行されます。

6.15 プログレスバーコントロール

6.15.1 解説

プログレスバーコントロールは、処理の進行状況表示に使用します。



6.15.2 使い方

プログレスバーコントロールを使用して、実行時間の長い操作のステータスや変数の値を表示します。

SPEL+ のグローバル変数 (配列を除く) の表示方法

Variableプロパティを設定することによって、自動的にSPEL+グローバル変数の値を表示することができます。

注意

Variableプロパティ設定でSPEL+のグローバル変数が表示されない場合、プロジェクトをビルドしてください。

6.15.3 プログレスバーコントロールのプロパティ

プロパティ	説明
BackColor	コントロールの背景色を設定します。 デフォルト: Control
BorderStyle	コントロールのボーダーのスタイルを設定します。 デフォルト: None
Enabled	コントロールを有効にするかを設定します。 デフォルト: True
EventTaskType	イベントで起動するタスクタイプを設定します。 デフォルト: Normal
ForeColor	コントロールのテキストの色を設定します。 デフォルト: ControlText
Height	コントロールの高さをピクセル数で設定します。
Left	コントロールの表示位置の左の座標をピクセル数で設定します。
Maximum	コントロールが処理している範囲の上限を設定します。 デフォルト: 100
Minimum	コントロールが処理している範囲の下限を設定します。 デフォルト: 0
Name	コントロールの名前を設定します。 デフォルト: ProgressBarXX
Orientation	コントロールの向きを設定します。 デフォルト: Horizontal
ProgressBarStyle	コントロールのスタイルを設定します。 デフォルト: Continuous
TabIndex	コントロールのタブインデックスを設定します。

ToolTipText	コントロールのツールチップで表示されるテキストを設定します。 デフォルト: 空白
Top	コントロールの表示位置の上の座標をピクセル数で設定します。
Value	コントロールの現在位置を表す値を設定します。 デフォルト: 50
Variable	表示させたいSPEL+のグローバル変数 (配列を除く)を設定します。 デフォルト: None
Visible	コントロールを表示するか、隠すかを設定します。 デフォルト: True
Width	コントロールの幅をピクセル数で設定します。

6.15.4 プログレスバーコントロールのイベント

イベント	説明
Click	コントロールがクリックされたときに実行されます。
DbClick	コントロールがダブルクリックされたときに実行されます。

6.16 トラックバーコントロール

6.16.1 解説

トラックバーコントロールは、スライダーをドラッグして値を設定して、その設定値を表示するために使用します。



6.16.2 使い方

値をグラフィカルに変更したい場合に、トラックバーコントロールを使用します。

SPEL+のグローバル変数 (配列を除く)の表示方法

Variableプロパティを使用すると、トラックバーの値に従って、自動でSPEL+グローバル変数の値を設定できます。

注意

Variable プロパティ設定で、SPEL+のグローバル変数が更新されない場合、プロジェクトをビルドしてください。

6.16.3 トラックバーコントロールのプロパティ

プロパティ	説明
BackColor	コントロールの背景色を設定します。 デフォルト: Control

プロパティ	説明
Enabled	コントロールを有効にするかを設定します。 デフォルト: True
EventTaskType	イベントで起動するタスクタイプを設定します。 デフォルト: Normal
Height	コントロールの高さをピクセル数で設定します。
LargeChange	マウスのクリックや、Page UpおよびPage Downキーを押す時、スライダーが移動するポジションの数を設定します。 デフォルト: 5
Left	コントロールの表示位置の左の座標をピクセル数で設定します。
Maximum	コントロールが処理している範囲の上限を設定します。 デフォルト: 10
Minimum	コントロールが処理している範囲の下限を設定します。 デフォルト: 0
Name	コントロールの名前を設定します。 デフォルト: TrackBarXX
Orientation	コントロールの向きを設定します。 デフォルト: Horizontal
SmallChange	キーボードの入力 (方向キー) に対して、スライダーが移動するポジションの数を設定します。 デフォルト: 1
TabIndex	コントロールのタブインデックスを設定します。
TickFrequency	目盛りマーク間の位置の数を設定します。 デフォルト: 1
TickStyle	目盛りをトラックバーのどこに表示するかを設定します。 デフォルト: BottomRight
ToolTipText	コントロールのツールチップで表示されるテキストを設定します。 デフォルト: 空白
Top	コントロールの表示位置の上の座標をピクセル数で設定します。
Value	コントロールの現在位置を表す値を設定します。 デフォルト: 0
Variable	Valueプロパティに反映するSPEL変数を設定します。 デフォルト: None
Visible	コントロールを表示するか、隠すかを設定します。 デフォルト: True
Width	コントロールの幅をピクセル数で設定します。

6.16.4 トラックバーコントロールのイベント

イベント	説明
Scroll	トラックバーのスライダーが移動したときに実行されます。

6.17 グリッドコントロール

6.17.1 解説

グリッドコントロールは、データをスプレッドシート方式で表示・編集するために使用します。

Column 0	Column 1

6.17.2 使い方

グリッドコントロールには、行と列のデータを含むセルがあります。オペレーターは、行と列を選択でき、指定した範囲のセルを自由に編集できます。

デザイン時:

1. GridEditorプロパティをクリックし、[GridEditor]ダイアログを開きます。
2. 適切な値にプロパティを設定し、グリッドを設計します。
3. ダイアログを閉じます。

実行時:

- CellTexプロパティを使って、セルの文字列を読み書きできます。
- AddRowプロパティと、RemoveRowプロパティを使って、行の追加や削除ができます。
- CellForeColorプロパティと、CellBackColorプロパティを使って、セルのForeColorとBackColorを変更できます。

6.17.3 グリッドコントロールのプロパティ

プロパティ	説明
AddRow	実行時に行を追加します。
BorderStyle	コントロールの背景色を設定します。 デフォルト: Control
CellBackColor	セルの背景色を設定・取得します。
CellForeColor	セルのForeColorを設定・取得します。
CellText	セル内の文字列を設定・取得します。
Enabled	コントロールを有効にするかを設定します。 デフォルト: True

プロパティ	説明
EventTaskType	イベントで起動するタスクタイプを設定します。 デフォルト: Normal
GridEditor	[GridEditor]ダイアログを開いて、グリッドコントロールを設定します
Height	コントロールの高さをピクセル数で設定します。
Left	コントロールの表示位置の左の座標をピクセル数で設定します。
Name	コントロールの名前を設定します。 デフォルト: GridXX
RemoveRow	行を削除します。
RowCount	行数を設定・取得します。
ScrollBars	スクロールバーを表示するときに設定します。 デフォルト: None
SelectedIndex	選択した行を設定・取得します。(SPELプログラムからのみ有効)
Top	コントロールの表示位置の上の座標をピクセル数で設定します。
Visible	コントロールの表示・非表示を設定します。 デフォルト: True
Width	コントロールの幅をピクセル数で設定します。

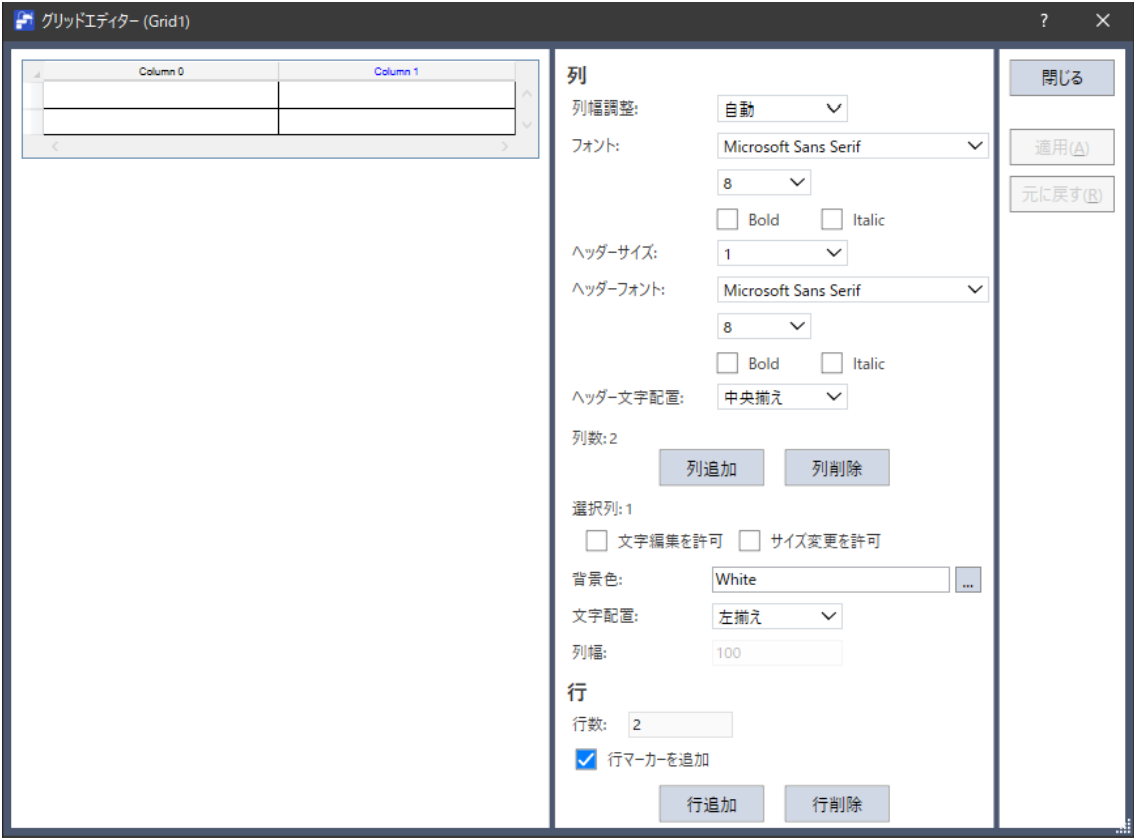
6.17.4 グリッドコントロールのイベント

イベント	説明
CellChanged	変更したセルを離れたときに実行されます。
Click	コントロールがクリックされたときに実行されます。
DbClick	コントロールがダブルクリックされたときに実行されます。

6.17.5 グリッドエディター

グリッドエディターは、ユーザーがグリッドコントロールを修正・設定できるダイアログです。デザイン時のみ、使用できます。

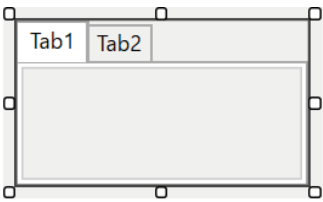
最適な設定にするために、多くのプロパティがあります。グリッドエディターの詳細は、GridEditorプロパティを参照してください。



6.18 タブコントロール

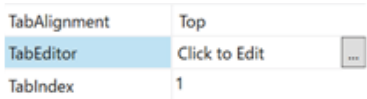
6.18.1 解説

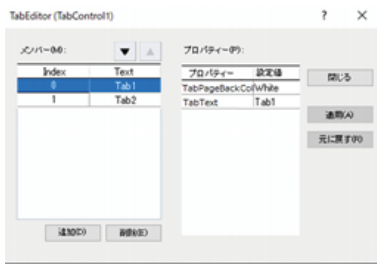
タブコントロールは、タブ切り替え可能なページを作成できます。設定や状態確認といった目的ごとに、表示するページを切り替えられる画面の作成が可能となります。



6.18.2 使い方

タブコントロールを使って、目的ごとのページを作成します。タブコントロールを配置したあと、TabEditorプロパティのボタンを押下してタブエディターを表示します。





タブエディターにてタブページの追加や削除を行います。またそれぞれのタブのテキストやフォント、タブページの背景色の設定を行った後、タブエディターを閉じます。タブコントロールのサイズや位置を変更した後、追加したいページに対応するタブを選択します。タブページが切り替わるので、タブページ上に表示したいコントロールを配置します。

6.18.3 タブコントロールのプロパティ

プロパティ	説明
Enabled	コントロールを有効にするかを設定します。 デフォルト: True
EventTaskType	イベントで起動するタスクタイプを設定します。 デフォルト: Normal
Height	コントロールの高さをピクセル数で設定します。
Left	コントロールの表示位置の左の座標をピクセル数で設定します。
Name	コントロールの名前を設定します。 デフォルト: PictureBoxXX
SelectedIndex	選択されたアイテムの番号を取得します。(SPELプログラムからのみ有効)
TabAlignment	コントロールに対するタブの配置場所を設定します。 デフォルト: Top
TabEditor	タブの追加や削除および並び替えを行うためのタブエディター画面を表示します。
TabHeaderSize	タブヘッダーのサイズを設定します。(自動で調整するかどうか、幅、高さ)
TabHeaderTextAlign	タブヘッダーのテキストの配置を設定します。 デフォルト: Center
TabHeaderAutoFit	タブヘッダーのサイズをテキストのサイズに合わせて自動で調整するかどうかを設定します。 デフォルト: True
TabHeaderHeight	タブヘッダーの高さをピクセル数で設定します。TabHeaderAutoFitがFalseの時に値が反映されます。
TabHeaderWidth	タブヘッダーの幅をピクセル数で設定します。TabHeaderAutoFitがFalseの時に値が反映されます。
TabIndex	コントロールのタブインデックスを設定します。
ToolTipText	コントロールのツールチップで表示されるテキストを設定します。 デフォルト: 空白

プロパティ	説明
Top	コントロールの表示位置の上の座標をピクセル数で設定します。
Visible	コントロールを表示するか、隠すかを設定します。 デフォルト:True
Width	コントロールの幅をピクセル数で設定します。

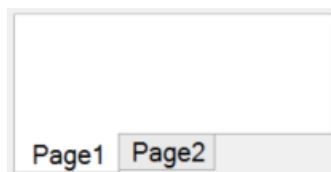
6.18.4 タブコントロールのイベント

イベント	説明
Click	コントロールがクリックされたときに実行されます。
SelectionChanged	タブページの選択を変更した時に実行されます。

6.19 ウィザードコントロール

6.19.1 解説

ウィザードコントロールは、ウィザード形式の画面を作成するために使用します。StandardエディションおよびPremiumエディションで使用できます。

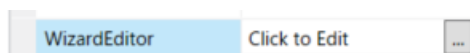


6.19.2 使い方

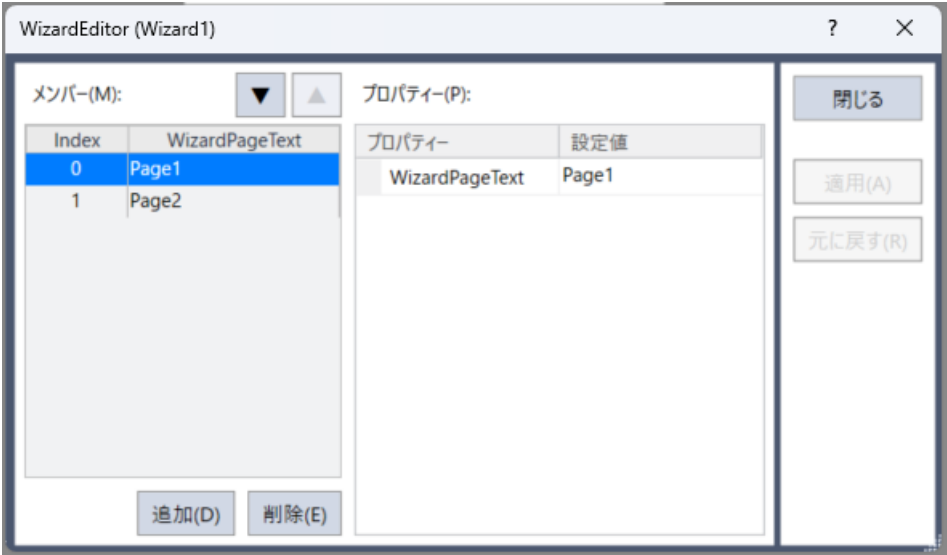
ウィザードコントロールには、複数のページを追加することができます。

デザイン時:

1. WizardEditorプロパティのボタンをクリックし、[WizardEditor]ダイアログを開きます。

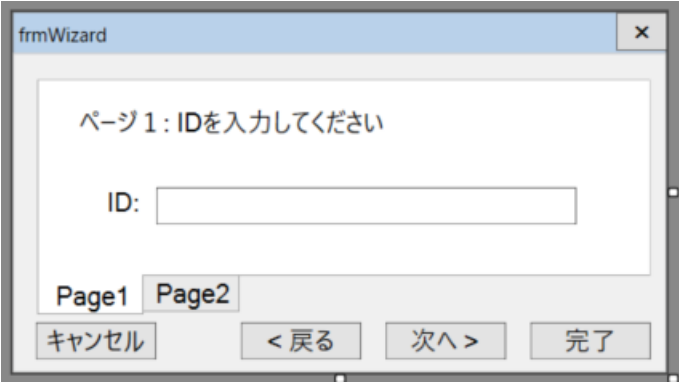


2. 必要なページを追加し、適切な値にプロパティを設定します。
3. ダイアログを閉じます。
4. 各ページに任意のコントロールを配置し、適切な値にプロパティを設定します。



実行時:

- SelectedIndexプロパティを使って、表示するページの切替ができます。
例:「戻る」、「次へ」のボタンコントロールを配置し、ClickイベントでSelectedIndexプロパティを設定する。



```
'戻るボタンのClickイベント関数
Function frmWizard_btnBack_Click(Sender$ As String)
    '前のページを表示する
    Integer index
    GGet frmWizard.Wizard1.SelectedIndex, index
    index = index - 1
    GSet frmWizard.Wizard1.SelectedIndex, index
Fend

'次へボタンのClickイベント関数
Function frmWizard_btnNext_Click(Sender$ As String)
    '次のページを表示する
    Integer index
    GGet frmWizard.Wizard1.SelectedIndex, index
    index = index + 1
    GSet frmWizard.Wizard1.SelectedIndex, index
Fend
```

6.19.3 ウィザードコントロールのプロパティ

プロパティ	説明
BackColor	コントロールの背景色を設定します。 デフォルト: Window

プロパティ	説明
Enabled	コントロールを有効にするかを設定します。 デフォルト: True
EventTaskType	イベントで起動するタスクタイプを設定します。 デフォルト: Normal
Height	コントロールの高さをピクセル数で設定します。
Left	コントロールの表示位置の左の座標をピクセル数で設定します。
Name	コントロールの名前を設定します。 デフォルト: WizardXX
SelectedIndex	選択されたアイテムの番号を取得、設定します。(SPELプログラムからのみ有効)
ShowNavigators	ページを切り替えるためのヘッダーを表示するかを設定します。(デザインエリアでの表示のみに反映されます。)
TabIndex	コントロールのタブインデックスを設定します。
ToolTipText	コントロールのツールチップで表示されるテキストを設定します。 デフォルト: 空白
Top	コントロールの表示位置の上の座標をピクセル数で設定します。
Visible	コントロールを表示するか、隠すかを設定します。 デフォルト: True
Width	コントロールの幅をピクセル数で設定します。
WizardEditor	ページの追加や削除および並び替えを行うためのウィザードエディター画面を表示します。

6.19.4 ウィザードコントロールのイベント

イベント	説明
Click	コントロールがクリックされたときに実行されます。
SelectionChanged	タブページの選択を変更した時に実行されます。

6.20 ツリービューコントロール

6.20.1 解説

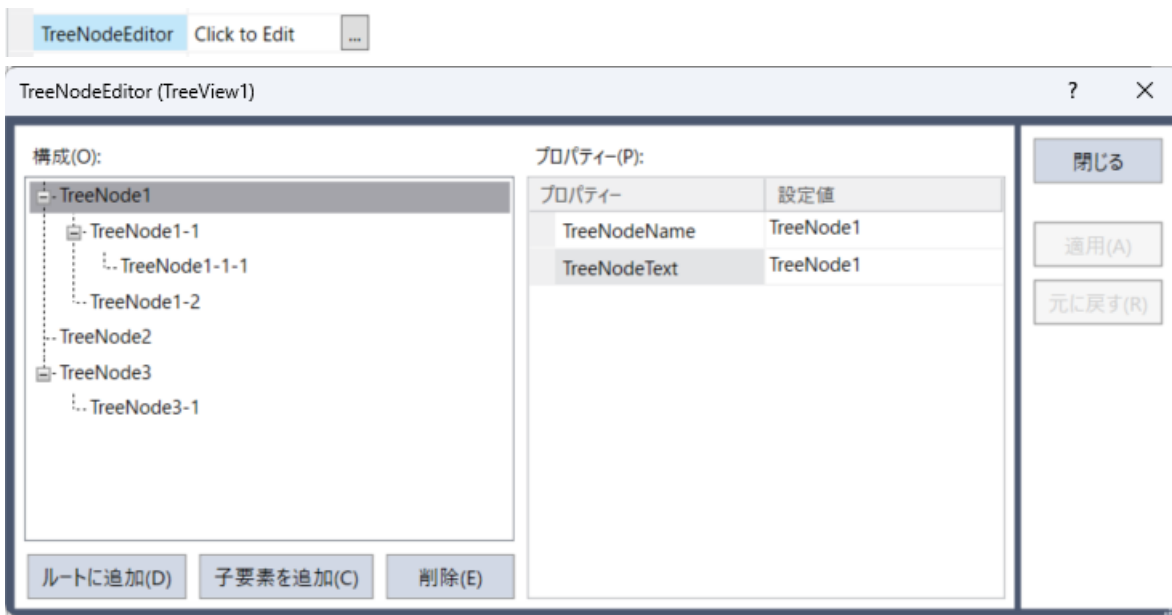
ツリービューコントロールは、折りたたみ可能なノードを使用して階層構造に情報を表示するコントロールです。StandardエディションおよびPremiumエディションで使用できます。

6.20.2 使い方

ツリービューコントロールを使って、階層構造に情報を表示します。

デザイン時:

1. TreeNodeEditorプロパティのボタンをクリックし、[TreeNodeEditor]ダイアログを開きます。



2. [TreeNodeEditor]ダイアログにてノードの追加や削除、並べ替えを行います。
3. それぞれのノードの名前やテキストの設定を行った後、[TreeNodeEditor]ダイアログを閉じます。

[TreeNodeEditor]ダイアログの詳細は、以下を参照してください。

TreeNodeEditor プロパティ

実行時:

以下のプロパティを使って、ノードの追加, 削除, 並べ替え, テキスト変更ができます。

- AddNodeプロパティ
- RemoveNodeプロパティ
- Parentプロパティ
- Textプロパティ

```
Function AddNode
    String parentNodeName$, childNodeName$
    parentNodeName$ = "Node1"
    childNodeName$ = "Node2"

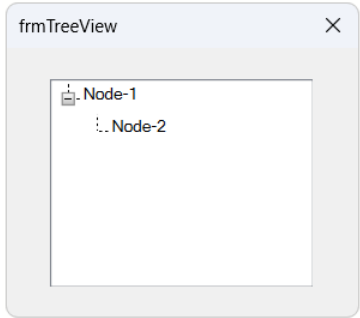
    'ノードを追加します。
    GSet frmMain.TreeView1.AddNode, parentNodeName$
    GSet frmMain.TreeView1.AddNode, childNodeName$

    'ノードに表示する文字列を設定します。
    GSet frmMain.TreeView1.Text (parentNodeName$), "Node-1"
    GSet frmMain.TreeView1.Text (childNodeName$), "Node-2"

    'ノードの親子関係を設定します。(childNodeName$の親ノードをparentNodeName$に設定します。)
    GSet frmMain.TreeView1.Parent (childNodeName$), parentNodeName$
End

Function RemoveNode (nodeName$ As String)
    'ノードを削除します。(親ノードを削除すると、子ノードも削除されます。)
    GSet frmMain.TreeView1.RemoveNode, nodeName$
End
```

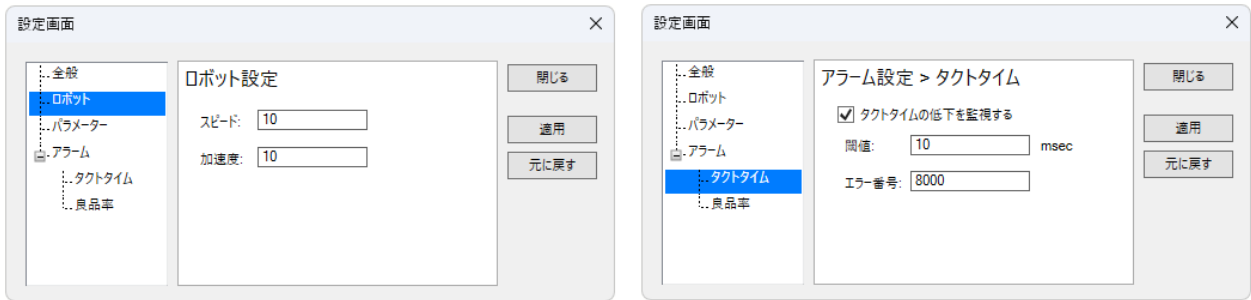
AddNodeファンクションを実行すると、以下の画面が表示されます。



TreeViewの応用例 (複数カテゴリの設定画面):

他のコントロールと組み合わせることで、複数のカテゴリに分かれた項目を設定する画面に応用できます。

1. 以下のようにコントロールを配置します。
- 左側：ツリービューコントロール
 - 中央：ウィザードコントロール
 - 右側：ボタンコントロール



2. SelectedNodeChangedイベントを使用し、ツリービューコントロールで選択されているノードが変更されたら、ウィザードコントロールのページを切り替えます。

```
Function frmMain_TreeView1_SelectedNodeChanged(Sender$ As String, NodeName$ As String)
    Integer wizardIndex

    ' 選択されたノードに応じて、ウィザードコントロールの表示ページを切り替える
    Select NodeName$
        Case "TreeNode1"; wizardIndex = 0
        Case "TreeNode2"; wizardIndex = 1
        Case "TreeNode3"; wizardIndex = 2
        Case "TreeNode4"; wizardIndex = 3
        Case "TreeNode5"; wizardIndex = 4
        Case "TreeNode6"; wizardIndex = 5
    Send

    GSet frmMain.Wizard1.SelectedIndex, wizardIndex
End
```

6.20.3 ツリービューのプロパティー

プロパティー	説明
AddNode	指定したノードを追加します。(SPELプログラムからのみ有効)
BackColor	コントロールの背景色を設定します。 デフォルト: Window

プロパティ	説明
BorderStyle	コントロールのボーダーのスタイルを設定します。 デフォルト:Fixed3D
Enabled	コントロールを有効にするかを設定します。 デフォルト: True
EventTaskType	イベントで起動するタスクタイプを設定します。 デフォルト:Normal
Font	コントロールのテキストを表示するフォントを設定します。 デフォルト: Microsoft Sans Serif 8.25 pt
FontBold	現在のフォントを太字にします。 デフォルト: False
FontItalic	現在のフォントをイタリックにします。 デフォルト: False
FontName	現在のフォントの名前を設定します。 デフォルト: Microsoft Sans Serif
FontSize	現在のフォントのサイズをポイントで設定します。 デフォルト: 8.25
ForeColor	コントロールのテキストの色を設定します。 デフォルト: WindowText
Height	コントロールの高さをピクセル数で設定します。
Left	コントロールの表示位置の左の座標をピクセル数で設定します。
Name	コントロールの名前を設定します。 デフォルト:TreeViewXX
Parent	指定したノードの親を取得・設定します。(SPELプログラムからのみ有効)
RemoveNode	指定したノードを削除します。(SPELプログラムからのみ有効)
TabIndex	コントロールのタブインデックスを設定します。
Text	コントロールの表示テキストを設定します。デフォルト: 空白
ToolTipText	コントロールのツールチップで表示されるテキストを設定します。 デフォルト: 空白
Top	コントロールの表示位置の上の座標をピクセル数で設定します。指定したノードを削除します。(SPELプログラムからのみ有効)
TreeNodeEditor	ノードの追加や削除および並び替えを行うためのツリーノードエディター画面を表示します。

プロパティ	説明
Visible	コントロールを表示するか、隠すかを設定します。 デフォルト: True
Width	コントロールの幅をピクセル数で設定します。

6.20.4 ツリービューのイベント

イベント	説明
Click	コントロールがクリックされたときに実行されます。
SelectedNodeChanged	ノードの選択を変更した時に実行されます。

7. 操作

7.1 概要

本章では、以下の項目を説明します:

- プログラムモードでGUI開発
- オートモードでGUIをスタートアップに登録する方法
- 一時停止、継続実行の操作
- 非常停止の操作
- ヘルプファイルの使い方

7.2 プログラムモードでGUI開発

GUIアプリケーションの開発とデバッグはプログラムモードで行います。

まず、プログラムモードでEpson RC+ 8.0を起動してください。

7.2.1 GUI設計

GUIアプリケーションを設計するために、簡単なガイドラインを示します。

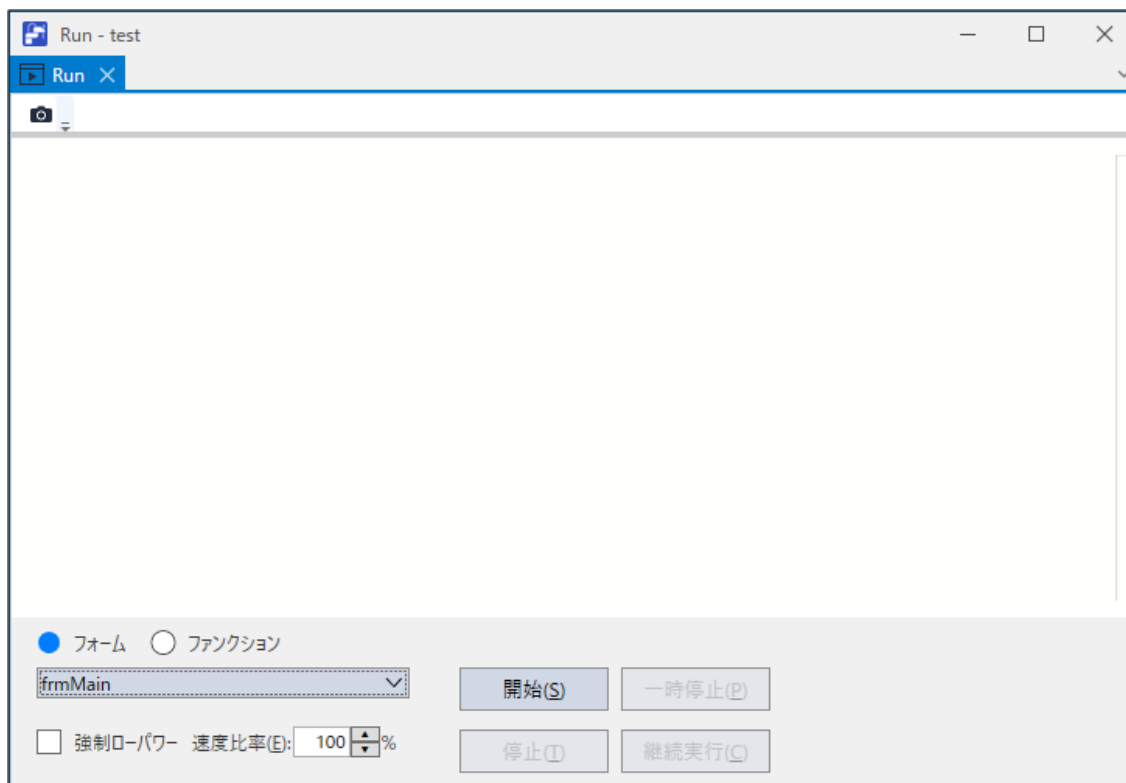
1. アプリケーションのために1つ以上のフォームを作成してください。通常、そのうちの1つをメインフォームとして設計します。
2. GUIをどのように起動するかを決めてください。スタートアップフォームとして、ひとつのフォームを指定することができます。
このフォームは、Epson RC+をオペレーターモードで起動すると自動的に表示されます。
起動時にスタートアップフォームを表示したくないときは、オペレーターウィンドウが表示された後、SPEL+プログラムからGShowステートメントまたはGShowDialog関数を使ってフォームを表示することができます。
3. フォームにコントロールを追加します。
4. コントロールに必要なイベントを追加します。
5. フォームとコントロールの使い方については、以下を参照してください。

[フォームとコントロールの使い方](#)

7.2.2 デバッグ

Runウィンドウから、プロジェクト内のフォームの実行が可能です。

1. Runウィンドウを開きます。



2. ドロップダウンリストから実行したいフォームを選択します。
3. [開始]ボタンをクリックすると、選択されたフォームが表示され、コントロールを使うことができます。
4. 停止するには、フォームを閉じるか、Runウィンドウの[停止]ボタンをクリックします。

SPEL+タスクで実行されるGUIイベントハンドラーは、イベントハンドラー中にブレイクポイントの設定や、プログラムのステップ実行、変数の値の確認ができます。

7.3 オペレーターモード

アプリケーションをオペレーターモードで実行するには、Epson RC+のセットアップが必要です。コントローラー起動時に、GUIフォームの1つを表示するか、オペレーターウィンドウを表示するか選択した後、プログラムからGUIフォームを表示するかを選択できます。

Epson RC+をオペレーターモードで起動する

1. セットアップメニューから[システム設定]を選択します。
2. [スタートモード]画面で、[オペレーターモード]ラジオボタンをチェックし、[適用]、[閉じる]ボタンをクリックします。

起動時に表示するフォームの設定

1. GUI Builderウィンドウを開きます。
2. フォームエクスプローラーから、起動時に表示したいフォームを右クリックし、[スタートアップフォーム]を有効にします。
3. ツールバーの[プロジェクトの保存]ボタンをクリックします。

7.4 一時停止、継続実行の操作

一時停止や継続実行を、GUIアプリケーションから操作することもできます。

通常、フォーム上に一時停止ボタンや継続実行ボタンを配置します。ボタンが有効に動作するように、

EventTaskTypeプロパティを1 - NoPauseに設定してください。

これは、コントローラの一時停止状態の中で、イベントハンドルのプログラムを実行する必要があるためです。

```
Function frmMain_btnPause_Click(Sender$ As String)
    Pause
Fend
Function frmMain_btnCont_Click(Sender$ As String)
    Cont
Fend
```

7.5 非常停止の操作

非常停止が入力された後に、ロボットを再度動作させるためには、非常停止状態をリセットする必要があります。そのためには、リセットボタンのEventTaskTypeプロパティを2 - NoEmgAbortに設定してください。

```
Function frmMain_btnReset_Click(Sender$ As String)
    Reset
    If EstopOn Then
        MsgBox "EStop could not be reset"
    EndIf
Fend
```

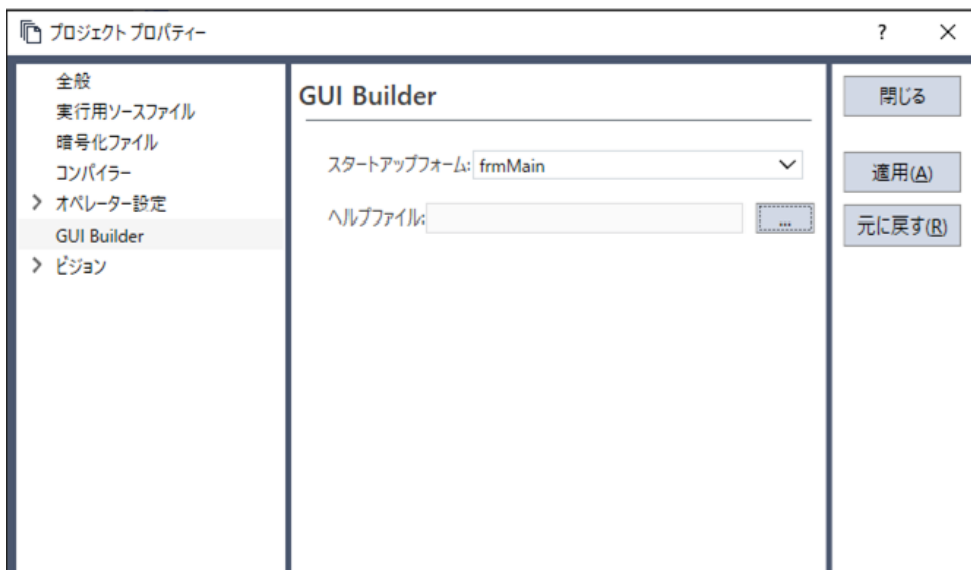
7.6 ヘルプファイルの使い方

GUI Builderは分かりやすいヘルプ機能をサポートしています。オペレーターがヘルプボタンをクリックしたときに表示するヘルプファイル、トピックを設定します。

ヘルプファイルは、マイクロソフトHTMLヘルプフォーマットで作成してください。

ヘルプファイルを使用するために

1. プロジェクトメニュー-[プロパティ]-[GUI Builder]を選択します。
2. ヘルプファイルを設定します。



3. 各フォームからヘルプを表示するには、HelpButtonプロパティをTrueにして、HelpIDプロパティに表示したいトピックのIDを設定します。

8. GUI Builderリファレンス

8.1 概要

本章では、GUI Builderのフォーム、コントロールのプロパティとイベントについて説明します。また、GUI Builderに関係するすべてのSPEL+コマンドについても説明します。

GUI Builderの使用についての詳しい情報は、GUI Builderマニュアルを参照してください。

8.2 GUI Builderプロパティとイベント書式

GUI Builderのプロパティとイベントを以降のページで挙げています。

プロパティやリザルトリファレンスに関する説明は次のとおりです。

項目	説明
適用	プロパティやイベントを適用できるGUI Builderオブジェクトを表示します。 (例. Button、Label、CheckBox…) プロパティやイベントがフォームで使用される場合は、Formを表示します。
解説	各プロパティやイベントについての簡潔な説明です。
用法	SPEL+言語からプロパティやイベントにアクセスする方法です。
値	プロパティに設定できる値、戻り値の範囲が記載されています。デフォルト値がある場合、その値も記載されます。
詳細説明	解説の内容より詳しい説明です。それぞれのプロパティ特有の注意事項や特記事項が記載されていますので、そのプロパティをご使用になる前に必ずこの項目をお読みください。
参照	関連するプロパティ、リザルト、オブジェクト、トピックをあげています。
プログラム実行時のみ	プロパティまたはリザルトがプログラム実行中のみに適用しているとき、プロパティ名かリザルト名の下に記載されています。プログラム実行中のみのプロパティとリザルトは、GUI Builderからはアクセスできません。SPEL+プログラムからのみアクセスすることができます。

8.3 A

8.3.1 AcceptButton プロパティ

適用

Form

説明

Enterキーが押されたときにクリックイベントを実行するボタンを設定・取得します。

使い方

```
GGet Form.AcceptButton, var  
GSet Form.AcceptButton, value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

var

プロパティの値を取得する文字列変数を指定します。

value

プロパティに設定する文字列式を指定します。

値

フォーム上のボタンの名前

デフォルト: None

注記

設計したGUI上で、Enterキーが押されたときに実行するデフォルトの動作を指定できます。カレントフォーム上、またはカレントフォーム上のコンテナにあるボタンを指定します。

このプロパティを指定することで、ユーザーがマウス操作で操作を終了するかわりに、Enterキーを押すことで操作することを可能にします。

コントロールによっては、Enterキー入力を別の操作が優先する場合があります。例えば複数行入力のテキストボックスは、Enterキーが押された場合、新しい行を追加します。

参照

[フォーム](#), [ボタンコントロール](#), [CancelButton](#) プロパティ

使用例

```
GSet frmMain.AcceptButton, "btnOK"
```

8.3.2 AddItem プロパティ

適用

ListBox, ComboBox

説明

リストボックスやコンボボックスコントロールに、アイテムを追加します。

使い方

```
GSet Form.Control.AddItem, value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

Control

コントロールの名前、またはコントロールの名前を含む文字列を指定します。そのコントロールは、指定されたフォーム上に存在しなくてはなりません。

value

プロパティに設定する文字列式を指定します。

値

リストに追加する文字列です。

参照

[コンボボックスコントロール](#), [リストボックスコントロール](#), [List プロパティ](#), [ListCount プロパティ](#), [Sorted プロパティ](#)

使用例

```
GSet frmMain.lstModels.AddItem, "Model1"
```

8.3.3 AddNode プロパティ

適用

TreeView

説明

ツリービューに項目(ノード)を追加します。第一階層に追加されます。他の項目の子としてぶら下げたい場合、続けてParentプロパティを設定します。

使い方

```
GSet Form.Control.AddNode, value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

Control

コントロールの名前、またはコントロールの名前を含む文字列を指定します。そのコントロールは、指定されたフォーム上に存在しなくてはなりません。

value

プロパティに設定する文字列式を指定します。

値

ツリービューに追加するノードを示す文字列です。表示名とは異なります。

参照

[ツリービューコントロール](#), [Parent プロパティ](#), [RemoveNode プロパティ](#), [Text プロパティ](#)

使用例

```
GSet frmMain.lstModels.AddNode, "Model1"  
GSet frmMain.lstModels.AddNode, "Model1-1"  
GSet frmMain.lstModels.Parent("Model1-1"), "Model1"
```

8.3.4 AddRow プロパティ

適用

Grid

説明

グリッドコントロールに、行を追加します。

使い方

```
GSet Form.Control.AddRow, value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

Control

コントロールの名前、またはコントロールの名前を含む文字列を指定します。そのコントロールは、指定されたフォーム上に存在しなくてはなりません。

value

新しく追加した行のインデックスを表す整数値です。“-1”を指定すると、新しい行が最終行の次に追加されます。それ以外の場合、新しい行は、指定した行のインデックスに追加されます。

参照

[グリッドコントロール](#), [RemoveRow プロパティ](#), [RowCount プロパティ](#)

使用例

```
GSet frmMain.Grid01.AddRow, 0
```

8.3.5 AppendText プロパティ

適用

TextBox

説明

テキストボックスコントロールに文字列を追加します。

使い方

```
GSet Form.Control.AppendText, value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

Control

コントロールの名前、またはコントロールの名前を含む文字列を指定します。そのコントロールは、指定されたフォーム上に存在しなくてはなりません。

value

プロパティに設定する文字列式を指定します。

値

コントロールの最後に追加される文字列です。

参照

[テキストボックスコントロール](#)

使用例

```
GSet frmMain.txtStatus.AppendText, "Cycle Complete"
```

8.3.6 AllowStateChange プロパティ

適用

LED

説明

LEDコントロールをダブルクリックしたとき、出力を反転させる動作を許可するかどうかを設定、取得します。

使い方

```
GGet Form.Control.AllowStateChange, var  
GSet Form.Control.AllowStateChange, value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

Control

コントロールの名前、またはコントロールの名前を含む文字列を指定します。そのコントロールは、指定されたフォーム上に存在しなくてはなりません。

var

プロパティの値を取得する論理変数を指定します。

value

プロパティに設定する論理式を指定します。

値

False

True

デフォルト: False

注記

IOTypeプロパティが0 - Inputに設定されている場合は、AllowStateChangeプロパティをTrueに設定しても、出力を反転できません。出力を反転させるには、IOTypeプロパティを1 - Outputまたは2 - Memoryに設定してください。

参照

[LEDコントロール, IOType プロパティ](#)

使用例

```
GSet frmMain.Led1.AllowStateChange, True
```

8.4 B

8.4.1 BackColor プロパティ

適用

Form, Button, Label, TextBox, RadioButton, CheckBox, ListBox, ComboBox, PictureBox, GroupBox, LED, ProgressBar, TrackBar, StatusBar, Wizard, TreeView

説明

フォーム、またはコントロールの背景色を設定・取得します。

使い方

```
GGet Form.Control.BackColor, var  
GSet Form.Control.BackColor, value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

Control

コントロールの名前、またはコントロールの名前を含む文字列を指定します。

var

プロパティの値を取得する文字列変数を指定します。

value

プロパティに設定する文字列式を指定します。

値

フォーム、またはコントロールの背景色に指定する色の名前です。

デフォルト:

- Control (Form, Button, Label, RadioButton, CheckBox, PictureBox, GroupBox, LED)
- Window (TextBox, ListBox, ComboBox)

参照

[BackColorMode プロパティ](#), [フォーム](#), [ボタンコントロール](#), [ラベルコントロール](#), [テキストボックスコントロール](#), [ラジオボタンコントロール](#), [チェックボックスコントロール](#), [リストボックスコントロール](#), [コンボボックスコントロール](#), [ピクチャーボックスコントロール](#), [グループボックスコントロール](#), [LEDコントロール](#), [プログレスバーコントロール](#), [トラックバーコントロール](#), [ステータスバーコントロール](#), [ウィザードコントロール](#), [ツリービューコントロール](#)

使用例

```
GSet frmMain.lblStatus.BackColor, "Red"
```

8.4.2 BackColorMode プロパティ

適用

Button

説明

ボタンコントロールの背景色モードを設定・取得します。

使い方

```
GGet Form.Control.BackColorMode, var  
GSet Form.Control.BackColorMode, value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

Control

コントロールの名前、またはコントロールの名前を含む文字列を指定します。

var

プロパティの値を取得する整数変数を指定します。

value

プロパティに設定する整数式を指定します。

値

0 -VisualStyle: システムで設定されている背景色を使います。

1 - User: ユーザーが指定する背景色を使います。

デフォルト:VisualStyle

注記

BackColorModeが0のとき、Windowsの画面のプロパティで設定された背景色になります。

BackColorModeが1のとき、背景色はBackColorプロパティで指定された色になります。

参照

[BackColor プロパティ, ボタンコントロール](#)

使用例

```
GSet frmMain.btnOK.BackColorMode, BACKCOLORMODE_USER
```

8.4.3 BorderStyle プロパティ

適用

Label, TextBox, ListBox, PictureBox, VideoBox, LED, ProgressBar, StatusBar, Grid, TreeView

説明

コントロールのボーダースタイルを設定・取得します。

使い方

```
GGet Form.Control.BorderStyle, var  
GSet Form.Control.BorderStyle, value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

Control

コントロールの名前、またはコントロールの名前を含む文字列を指定します。

var

プロパティの値を取得する整数変数を指定します。

value

プロパティに設定する整数式を指定します。

値

0 - None

1 - FixedSingle

2 - Fixed3D

デフォルト:

- None (Label, PictureBox, VideoBox, LED)
- Fixed3D (TextBox, ListBox)

注記

コントロールのボーダースタイルを設定するのに使います。例えば、アプリケーションの状態を表示するラベルを、他のラベルと異なるボーダースタイルで表示することができます。

参照

[ラベルコントロール](#), [テキストボックスコントロール](#), [リストボックスコントロール](#),
[ピクチャーボックスコントロール](#), [ビデオボックスコントロール](#), [LEDコントロール](#),
[プログレスバーコントロール](#), [ステータスバーコントロール](#), [ツリービューコントロール](#)

使用例

```
GSet frmMain.lblStatus.BorderStyle, BORDERSTYLE_NONE
```

8.5 C

8.5.1 Camera プロパティ

適用

VideoBox

説明

ビデオボックスコントロールのカメラ番号を設定・取得します。

使い方

```
GGet Form.Control.Camera, var  
GSet Form.Control.Camera, value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

Control

コントロールの名前、またはコントロールの名前を含む文字列を指定します。

var

プロパティの値を取得する整数変数を指定します。

value

プロパティに設定する整数式を指定します。

値

ビデオボックスに表示するカメラ番号です。値が“0”の時は、VRunコマンドが実行されたときのカメラの画像が表示されます。画像を表示するときは、VideoEnabledプロパティを“True”に設定してください。

参照

[ビデオボックスコントロール](#), [VideoEnabled プロパティ](#)

使用例

```
GSet frmMain.VideoBox1.Camera, 1
```

8.5.2 CancelButton プロパティ

適用

Form

説明

ESCキーが押されたときにクリックイベントを実行するボタンを設定・取得します。

使い方

```
GGet Form.CancelButton, var  
GSet Form.CancelButton, value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

var

プロパティの値を取得する文字列変数を指定します。

value

プロパティに設定する文字列式を指定します。

値

フォーム上のボタンの名前

デフォルト: None

注記

キャンセルボタンは、ユーザーがEscキーを押すときにクリックされるボタンコントロールです。カレントフォーム上、またはカレントフォーム上のコンテナにあるボタンを指定します。

設計したGUI上でEscキーが押されたときに実行するデフォルトの動作を指定できます。このプロパティを指定することで、ユーザーがマウス操作で操作を終了するかわりに、Escキーを押すことで操作することを可能にします。

コントロールによっては、Escキー入力を別の操作が優先する場合があります。例えばコンボボックスのドロップダウンリストが開いている場合は、Escキーが押された場合、コンボボックスのドロップダウンリストを閉じます。キャンセルボタンに割り当てたボタンを可視状態にしなければ、Escキーの入力は無視されます。

参照

[フォーム](#), [ボタンコントロール](#), [AcceptButton プロパティ](#)

使用例

```
GSet frmMain.CancelButton, "btnCancel"
```

8.5.3 CellBackColor プロパティ

適用

Grid

説明

グリッドセルの背景色を取得・設定します。

使い方

```
GGet Form.Control.CellBackColor (row, column), var  
GSet Form.Control.CellBackColor (row, column), value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

Control

コントロールの名前、またはコントロールの名前を含む文字列を指定します。

row

セル行の整数式を指定します。最初の行のインデックスは“0”です。

column

セル列の整数式を指定します。最初の列のインデックスは“0”です。

var

プロパティの値を取得する文字列変数を指定します。

value

プロパティに設定する文字列式を指定します。

値

指定したグリッドセルの、背景色の名称を含む文字列です。

参照

[グリッドコントロール](#), [CellForeColor プロパティ](#), [CellText プロパティ](#)

使用例

```
GSet frmMain.gridMyData.CellBackColor(1, 1), "Red"
```

8.5.4 CellChanged イベント

適用

Grid

説明

変更したセルを離れたときに、発生するイベントです。

使い方

```
Form_Control_CellChanged  
(sender$ As String, CellText$ As String, RowNumber As Integer, ColumnNumber As Integer)
```

Sender\$

イベントを発生させたコントロールの名前が設定されます。

CellText\$

変更したセルのテキストが設定されます。

RowNumber

変更したセルの行番号が設定されます。

ColumnNumber

変更したセルの列番号が設定されます。

注記

CellChangedイベントは、ユーザーが変更したセルを離れたときに、応答するために使われます。Sender\$パラメーターを使用し、イベントを送信したコントロールを判別します。CellText\$パラメーターを使用し、セルの現在値を確認します。RowNumberパラメーターとColumnNumberパラメーターを使用して、イベントのトリガーであるセルを判別します。これを使うことで、特定のセルや任意のセルが、いつ変更されたかを確認するときに役に立ちます。

参照

[グリッドコントロール](#)

使用例

```
Function frmMain_Grid1_CellChanged(Sender$ As String, CellText$ As  
String, RowNumber As Integer, ColumnNumber As Integer)  
  
Xqt main  
Fend
```

8.5.5 CellForeColor プロパティ

適用

Grid

説明

グリッドセルのテキストの色を設定・取得します。

使い方

```
GGet Form.Control.CellForeColor (row, column), var  
GSet Form.Control.CellForeColor (row, column), value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

Control

コントロールの名前、またはコントロールの名前を含む文字列を指定します。

row

セル行の整数式を指定します。最初の行のインデックスは0です。

column

セル列の整数式を指定します。最初の列のインデックスは0です。

var

プロパティの値を取得する文字列変数を指定します。

value

プロパティに設定する文字列式を指定します。

値

指定したグリッドセルの、テキストの色の名称を含む文字列です。

参照

[グリッドコントロール](#), [CellBackColor プロパティ](#), [CellText プロパティ](#)

使用例

```
GSet frmMain.gridMyData.CellForeColor(1, 1), "Red"
```

8.5.6 CellText プロパティ

適用

Grid

説明

グリッドセルのテキストを取得・設定します。

使い方

```
GGet Form.Control.CellText (row, column), var  
GSet Form.Control.CellText (row, column), value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

Control

コントロールの名前、またはコントロールの名前を含む文字列を指定します。

row

セル行の整数式を指定します。最初の行のインデックスは0です。

column

セル列の整数式を指定します。最初の列のインデックスは0です。

var

プロパティの値を取得する文字列変数を指定します。

value

プロパティに設定する文字列式を指定します。

値

特定のグリッドセルのテキストを含む文字列です。

参照

[グリッドコントロール](#), [CellForeColor プロパティ](#), [CellBackColor プロパティ](#)

使用例

```
GSet frmMain.gridMyData.CellText(1, 1), "Value1"
```

8.5.7 Checked プロパティ

適用

RadioButton, Checkbox

説明

チェックボックスか、ラジオボタンコントロールがチェックされているかどうか設定・取得します。

使い方

```
GGet Form.Control.Checked, var  
GSet Form.Control.Checked, value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

Control

コントロールの名前、またはコントロールの名前を含む文字列を指定します。

var

プロパティの値を取得する論理変数を指定します。

value

プロパティに設定する論理式を指定します。

値

False

True

デフォルト: False

注記

このプロパティは、チェックボックスやラジオボタンコントロールがチェックされているか判断するために使います。チェックボックスやラジオボタンコントロールを使ってON/OFFの状態を表示するために使うこともできます。

参照

[ラジオボタンコントロール, チェックボックスコントロール](#)

使用例

```
GSet frmMain.chkHiPower.Checked, False
```

8.5.8 Click イベント

適用

Button, Label, RadioButton, Checkbox, ListBox, ComboBox, PictureBox, GroupBox, VideoBox, TextBox, ProgressBar, StatusBar, Grid, TabControl, Wizard, TreeView

説明

コントロールがクリックされたときに実行されるイベントです。

使い方

```
Form_Control_Click (Sender$ As String)
```

Sender\$

イベントを発生させたコントロールの名前が設定されます。

注記

クリックイベントは、ユーザーがコントロールをクリックしたときの応答をするために使われます。

Sender\$パラメーターにより、どのコントロールがイベントを発生させたか判断できます。これを使うことで、ラジオボタンやチェックボックスのように、複数のコントロールを1つのファンクション内で使用したいときに役に立ちます。

ハンドリングを、1つの関数で作成することが容易になります。

参照

[ボタンコントロール](#), [ラベルコントロール](#), [ラジオボタンコントロール](#), [チェックボックスコントロール](#), [リストボックスコントロール](#), [コンボボックスコントロール](#), [ピクチャーボックスコントロール](#), [グループボックスコントロール](#), [ビデオボックスコントロール](#), [プログレスバーコントロール](#), [ステータスバーコントロール](#), [ウィザードコントロール](#), [ツリービューコントロール](#)

使用例

```
Function frmMain_btnStart_Click(Sender$ As String) Xqt mainFend
```

8.5.9 Closed イベント

適用

Form

説明

フォームが閉じられたときに実行されるイベントです。

使い方

```
Form_Closed (Sender$As String)
```

Sender\$

イベントを発生させたフォームの名前が設定されます。

注記

このイベントは、フォームが閉じられたあとタスクを実行するときに使用します。

参照

[フォーム](#), [Loadイベント](#), [Resizeイベント](#)

使用例

```
Function frmMain_Closed(Sender$ As String)
    Print "frmMain was closed"
End
```

8.5.10 Closing イベント

適用

Form

説明

フォームを閉じる操作が行われたとき、フォームが閉じられる前に実行されるイベントです。

使い方

```
Form_Closing(Sender$ As String, ByRef Cancel As Boolean)
```

Sender\$

イベントを発生させたフォームの名前が設定されます。

Cancel

フォームを閉じる操作をキャンセルするかどうかを設定します。

Sender\$

イベントを発生させたフォームの名前が設定されます。

注記

このイベントは、フォームが閉じられる前にタスクを実行するときに使用します。

また、CancelをTrueに設定するとフォームが閉じられるのをキャンセルすることができます。

参照

[フォーム](#)

使用例

```
Global Boolean g_CanClose

Function Form_Closing(Sender$ As String, ByRef Cancel As Boolean)
    If g_CanClose = False Then
        'フォームが閉じられるのをキャンセルします
        Cancel = True
    EndIf
End
```

8.5.11 ControlBox プロパティ

適用

Form

説明

フォームのキャプションバーに、コントロールボックスを表示するかどうかを設定・取得します。

使い方

```
GGet Form.ControlBox, var  
GSet Form.ControlBox, value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

var

プロパティの値を取得する論理変数を指定します。

value

プロパティに設定する論理式を指定します。

値

False

True

デフォルト: True

注記

ControlBoxプロパティがTrueに設定された場合、コントロールボックスがフォームのタイトルバーの左上に表示されます。コントロールボックスは、フォームを閉じるのに使うことができます。

参照

フォーム, [MaximizeBox プロパティ](#), [MinimizeBox プロパティ](#)

使用例

```
GSet frmMain.ControlBox, False
```

8.5.12 Controls プロパティ

適用

Form

説明

フォーム上のコントロールの配列。インデックスを使用してコントロールのプロパティにアクセスする場合に使用します。

使い方

```
GGet Form.Controls(Index As Integer).Property, var  
GSet Form.Controls(Index As Integer).Property, value
```

Form

フォームの名前を指定します。

Index

フォーム上のコントロールのインデックスを含む整数式を指定します。

Property

アクセスするコントロールのプロパティの名前を指定します。

var

プロパティの値を取得する変数を指定します。データ型は指定するプロパティによって異なります。

value

プロパティに設定する式を指定します。データ型は指定するプロパティによって異なります。

注記

Controlsプロパティは、インデックスを使用してフォーム上のコントロールにアクセスします。これにより、コントロールの集合を順に処理し、共通するプロパティを設定または取得することが出来ます。

参照

[Countプロパティ](#), [Typeプロパティ](#)

使用例

```
Integer i, count  
String type$  
  
GGet frmMain.Controls.Count, count  
For i = 1 To count  
    GGet frmMain.Controls(i).Type, type$  
    If type$ = "Button" Then  
        GSet frmMain.Controls(i).Enabled, False  
    EndIf  
Next i
```

8.5.13 Countプロパティ

適用

Form.Controls

説明

フォーム上のコントロールの数を取得します。

使い方

```
GGet Form.Controls.Count, var
```

Form

フォームの名前を指定します。

var

プロパティの値を取得する変数を指定します。

値

フォーム上のコントロールの数を返します。

注記

Countプロパティは、フォーム上のコントロールの数を特定するために使用します。これにより、Controlsプロパティを使用して、すべてのコントロールを順に処理することが出来ます。

参照

[Controls プロパティ](#), [Typeプロパティ](#)

使用例

```
Integer count

GSet frmMain.Controls.Count, count
For i = 1 To count
    GGet frmMain.Controls(i).Type, type$
    If type$ = "Button" Then
        GSet frmMain.Controls(i).Enabled, False
    EndIf
Next i
```

8.6 D

8.6.1 DblClick イベント

適用

Label, ListBox, PictureBox, VideoBox, LED, ProgressBar, Grid

説明

コントロールがダブルクリックされたときに実行されるイベントです。

使い方

```
Form_Control_DblClick (Sender$ As String)
```

Sender\$

イベントを発生させたコントロールの名前が設定されます。

注記

クリックイベントは、ユーザーがコントロールをクリックしたときの応答をするために使われます。
Sender\$パラメーターにより、どのコントロールがイベントを発生させたか判断できます。

参照

[ラベルコントロール](#), [リストボックスコントロール](#), [ピクチャーボックスコントロール](#),
[ビデオボックスコントロール](#), [LEDコントロール](#), [プログレスバーコントロール](#)

使用例

```
Function frmMain_lstModels_DblClick(Sender$ As String) Integer index  
    GGet frmMain.lstModels.SelectedIndex, index  
    GGet frmMain.lstModels.List(index), g_CurrModel$  
End
```

8.6.2 DialogResult プロパティ

適用

Form

説明

フォームのダイアログ結果を設定・取得します。

使い方

```
GGet Form.DialogResult, var  
GSet Form.DialogResult, value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

var

プロパティの値を取得する整数変数を指定します。

value

プロパティに設定する整数式を指定します。

値

0 - None

1 - OK

2 - Cancel

デフォルト:Cancel

注記

DialogResultプロパティは、ユーザーがフォームを閉じるときに、了解して閉じたか、キャンセルして閉じたかを判断するために使います。

通常、ダイアログは、OKボタン、またはキャンセルボタンを持ちます。ユーザーがOKボタンを押したとき、ボタンクリックイベントの中で、DialogResultプロパティに1 - OKを設定します。[キャンセル]ボタンが押されたときには、2 - Cancelを設定します。

参照

[フォーム](#)

使用例

```
Function frmSetSpeed_btnOK_Click(Sender$ As String)  
    GSet frmSetSpeed.DialogResult, DIALOGRESULT_OK  
    GClose frmSetSpeed  
Fend  
Function frmMain_btnSetSpeed_Click(Sender$ As String)  
    Integer result  
    String speed$  
    result = GShowDialog(frmSetSpeed)  
    If result = DIALOGRESULT_OK Then  
        GGet frmSetSpeed.txtSpeed.Text, speed$  
        g_RobotSpeed = Val(speed$)  
    End If  
Fend
```

8.6.3 DropDownStyle プロパティ

適用

ComboBox

説明

コンボボックスのドロップダウンスタイルを設定・取得します。

使い方

```
GGet Form.Control.DropDownStyle, var  
GSet Form.Control.DropDownStyle, value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

Control

コントロールの名前、またはコントロールの名前を含む文字列を指定します。

var

プロパティの値を取得する整数変数を指定します。

value

プロパティに設定する整数式を指定します。

値

0 - Simple

1 - DropDown

2 - DropDownList

デフォルト:DropDown

注記

DropDownStyleプロパティは、ユーザーへのインターフェイスをコントロールします。

次の中から選択します。

- 常にリストを表示し選択できる
- 右側のドロップダウンボタンで表示したリストから選択でき、テキストボックス内のテキストを編集可能（デフォルト）
- 右側のドロップダウンボタンで表示したリストから選択できるが、テキストボックス内のテキストは編集不可

常にリストが表示されて、テキストの編集を許可しないのであれば、ListBoxコントロールを使ってください。

参照

[コンボボックスコントロール](#)

使用例

```
GSet frmMain.cmbPartNames.DropDownStyle, DROPDOWNSTYLE_SIMPLE
```

8.7 E

8.7.1 Enabled プロパティ

適用

Button, Label, TextBox, RadioButton, CheckBox, ListBox, ComboBox, PictureBox, GroupBox, Timer, VideoBox, LED, ProgressBar, TrackBar, StatusBar, Grid, TabControl, Wizard, TreeView

説明

コントロールが実行時に有効かどうか、設定・取得します。

使い方

```
GGet Form.Control.Enabled, var  
GSet Form.Control.Enabled, value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

Control

コントロールの名前、またはコントロールの名前を含む文字列を指定します。

var

プロパティの値を取得する論理変数を指定します。

value

プロパティに設定する論理式を指定します。

値

False

True

デフォルト: True (Timer以外) False (Timer)

注記

あるケースでは、ボタンのようなコントロールをオペレーターにクリックさせたくない場合があります。例えば、SPEL+のタスクが実行されているときに、[Pause]ボタンを押させたくないときは、EnabledプロパティをFalseに設定してください。

参照

[ボタンコントロール](#), [ラベルコントロール](#), [テキストボックスコントロール](#), [ラジオボタンコントロール](#), [チェックボックスコントロール](#), [リストボックスコントロール](#), [コンボボックスコントロール](#), [ピクチャーボックスコントロール](#), [グループボックスコントロール](#), [タイマーコントロール](#), [ビデオボックスコントロール](#), [LEDコントロール](#), [プログレスバーコントロール](#), [トラックバーコントロール](#), [ステータスバーコントロール](#), [タブコントロール](#), [ウィザードコントロール](#), [ツリービューコントロール](#)

使用例

```
GSet frmMain.btnPause.Enabled, False
```

8.7.2 EventTaskType プロパティ

適用

Form, Button, Label, TextBox, RadioButton, CheckBox, ListBox, ComboBox, PictureBox, GroupBox, VideoBox, LED, ProgressBar, TrackBar, StatusBar, Grid, TabControl, Wizard, TreeView

説明

イベントで起動するタスクタイプを取得します。

使い方

```
GGet Form.[Control].EventTaskType, var
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

Control

コントロールの名前、またはコントロールの名前を含む文字列を指定します。

var

プロパティの値を取得する整数変数を指定します。

値

0 - Normal

1 - NoPause

2 - NoEmgAbort

デフォルト:Normal

注記

EventTaskTypeプロパティを1 - NoPauseに設定すると、他の通常タスクが一時停止しているときに、イベントハンドラーを実行できるようになります。2 - NoEmgAbortに設定すると、他の通常タスクが一時停止しているときや非常停止状態でも、イベントハンドラーを実行できるようになります。

参照

フォーム, ボタンコントロール, ラベルコントロール, テキストボックスコントロール, ラジオボタンコントロール, チェックボックスコントロール, リストボックスコントロール, コンボボックスコントロール, ピクチャーボックスコントロール, グループボックスコントロール, ビデオボックスコントロール, LEDコントロール, プログレスバーコントロール, トラックバーコントロール, ステータスバーコントロール, タブコントロール, ウィザードコントロール, ツリービューコントロール

使用例

```
Integer Ttype  
GGet frmMain.btnPause.EventTaskType, Ttype
```

8.8 F

8.8.1 Font プロパティ

設計GUI上のみの機能です。

適用

Button, Label, TextBox, RadioButton, CheckBox, ListBox, ComboBox, GroupBox, LED, StatusBar, TreeView

説明

フォーム設計時に、コントロールのフォントを変更することができます。(フォント名、スタイル、サイズ)

使い方

GUI Builderのプロパティグリッド内で表示されるFontプロパティの値を変更することで表示文字のフォントを変更することができます。

プロパティ	設定値
Enabled	1 - True ▼
EventTaskType	Normal ▼
▼ Font	Microsoft Sans Serif, 8.25pt
FontName	Microsoft Sans Serif ▼
FontSize	8.25
FontBold	False ▼
FontItalic	False ▼
ForeColor	■ ControlText ▼
Height	20

フォントの設定を変更しプロジェクトを保存した後、設定が保存されます。

参照

ボタンコントロール, ラベルコントロール, テキストボックスコントロール, ラジオボタンコントロール, チェックボックスコントロール, リストボックスコントロール, コンボボックスコントロール, グループボックスコントロール, LEDコントロール, ステータスバーコントロール, ツリービューコントロール

8.8.2 FontBold プロパティ

適用

Button, Label, TextBox, RadioButton, CheckBox, ListBox, ComboBox, GroupBox, LED, StatusBar, TreeView

説明

フォーム設計時に、フォントが“標準”または“太字”かを設定・取得します。

使い方

```
GGet Form.Control.FontBold, var  
GSet Form.Control.FontBold, value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

Control

コントロールの名前、またはコントロールの名前を含む文字列を指定します。

var

プロパティの値を取得する論理変数を指定します。

value

プロパティに設定する論理式を指定します。

値

False

True

デフォルト: False

参照

[ボタンコントロール](#), [ラベルコントロール](#), [テキストボックスコントロール](#), [ラジオボタンコントロール](#), [チェックボックスコントロール](#), [リストボックスコントロール](#), [コンボボックスコントロール](#), [グループボックスコントロール](#), [LEDコントロール](#), [ステータスバーコントロール](#), [ツリービューコントロール](#), [FontName プロパティ](#), [FontItalic プロパティ](#), [FontSize プロパティ](#)

使用例

```
GSet frmMain.btnOK.FontBold, True
```

8.8.3 FontItalic プロパティ

適用

Button, Label, TextBox, RadioButton, CheckBox, ListBox, ComboBox, GroupBox, LED, StatusBar, TreeView

説明

フォーム設計時に、フォントが“標準”または“イタリック(斜体)”かを設定・取得します。

使い方

```
GGet Form.Control.FontItalic, var  
GSet Form.Control.FontItalic, value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

Control

コントロールの名前、またはコントロールの名前を含む文字列を指定します。

var

プロパティの値を取得する論理変数を指定します。

value

プロパティに設定する論理式を指定します。

値

False

True

デフォルト: False

参照

[ボタンコントロール](#), [ラベルコントロール](#), [テキストボックスコントロール](#), [ラジオボタンコントロール](#), [チェックボックスコントロール](#), [リストボックスコントロール](#), [コンボボックスコントロール](#), [グループボックスコントロール](#), [LEDコントロール](#), [ステータスバーコントロール](#), [ツリービューコントロール](#), [FontName プロパティ](#), [FontBold プロパティ](#), [FontSize プロパティ](#)

使用例

```
GSet frmMain.btnOK.FontItalic, True
```

8.8.4 FontName プロパティ

適用

Button, Label, TextBox, RadioButton, CheckBox, ListBox, ComboBox, GroupBox, LED, StatusBar, TreeView

説明

コントロールに使われているフォントの名称を設定・取得します。

使い方

```
GGet Form.Control.FontName, var  
GSet Form.Control.FontName, value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

Control

コントロールの名前、またはコントロールの名前を含む文字列を指定します。

var

プロパティの値を取得する文字列変数を指定します。

value

プロパティに設定する文字列式を指定します。

値

フォントの名称の文字列です。

デフォルト: Microsoft Sans Serif

参照

[ボタンコントロール](#), [ラベルコントロール](#), [テキストボックスコントロール](#), [ラジオボタンコントロール](#),
[チェックボックスコントロール](#), [リストボックスコントロール](#), [コンボボックスコントロール](#),
[グループボックスコントロール](#), [LEDコントロール](#), [ステータスバーコントロール](#),
[ツリービューコントロール](#), [FontSize プロパティ](#), [FontItalic プロパティ](#), [FontBold プロパティ](#)

使用例

```
GSet frmMain.txtStatus.FontName, "Courier New"
```

8.8.5 FontSize プロパティ

適用

Button, Label, TextBox, RadioButton, CheckBox, ListBox, ComboBox, GroupBox, LED, StatusBar, TreeView

説明

コントロールに使われるフォントのサイズを、設定・取得します。

使い方

```
GGet Form.Control.FontSize, var  
GSet Form.Control.FontSize, value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

Control

コントロールの名か、またはコントロールの名前を含む文字列を指定します。

var

プロパティの値を取得する実数変数を指定します。

value

プロパティに設定する実数式を指定します。

値

フォントのサイズのポイント値です。

デフォルト: 8.25

参照

[ボタンコントロール](#), [ラベルコントロール](#), [テキストボックスコントロール](#), [ラジオボタンコントロール](#),
[チェックボックスコントロール](#), [リストボックスコントロール](#), [コンボボックスコントロール](#),
[グループボックスコントロール](#), [LEDコントロール](#), [ステータスバーコントロール](#),
[ツリービューコントロール](#), [FontName プロパティ](#), [FontItalic プロパティ](#), [FontBold プロパティ](#)

使用例

```
GSet frmMain.btnOK.FontSize, 16
```

8.8.6 ForeColor プロパティ

適用

Button, Label, TextBox, RadioButton, CheckBox, ListBox, ComboBox, GroupBox, LED, ProgressBar, StatusBar, TreeView

説明

コントロールのテキストの色を設定・取得します。

使い方

```
GGet Form.Control.ForeColor, var  
GSet Form.Control.ForeColor, value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

Control

コントロールの名前、またはコントロールの名前を含む文字列を指定します。

var

プロパティの値を取得する文字列変数を指定します。

value

プロパティに設定する文字列式を指定します。

値

コントロールのテキストの色の名称の文字列です。

デフォルト: Control Text

参照

ボタンコントロール, ラベルコントロール, テキストボックスコントロール, ラジオボタンコントロール, チェックボックスコントロール, リストボックスコントロール, コンボボックスコントロール, グループボックスコントロール, LEDコントロール, プログレスバーコントロール, ステータスバーコントロール, ツリービューコントロール

使用例

```
GSet frmMain.btnOK.ForeColor, "Blue"
```

8.8.7 FormBorderStyle プロパティ

適用

Form

説明

フォームのボーダースタイルを設定・取得します。

使い方

```
GGet Form.FormBorderStyle, var  
GSet Form.FormBorderStyle, value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

var

プロパティの値を取得する整数変数を指定します。

value

プロパティに設定する整数式を指定します。

値

0 - None

1 - Fixed Single

2 - Fixed3D

3 - FixedDialog

4 - Sizable

デフォルト:FixedDialog

参照

[フォーム](#)

使用例

```
GSet frmMain.FormBorderStyle, FORMBORDERSTYLE_NONE  
GClose
```

8.8.8 FormTemplate プロパティ

適用

Form

説明

フォームのサイズをテンプレートから選択、設定します。

フォームのサイズをティーチペンダントTP4の画面サイズに合わせたい場合に使用します。

使い方

本プロパティはSPEL+プログラムからの設定には対応していません。

対象のフォームのプロパティグリッドに値を指定し、設定してください。

値

0 - Standard

1 - TP4Vertical

2 - TP4Horizontal

デフォルト:Standard

注記

値を1もしくは2に設定すると、以下のプロパティが固定されます。

- FormBorderStyle
- Height
- Left
- Width
- Top
- StartPosition 任意に設定したい場合は値を0にしてください。

参照

フォーム, [FormBorderStyle プロパティ](#), [Height プロパティ](#), [Left プロパティ](#), [Width プロパティ](#)

8.9 G

8.9.1 GClose ステートメント

適用

Form

説明

フォームを閉じます。

使い方

```
GClose Form
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。また、フォームIDを指定して、以下のシステムウィンドウを閉じることも出来ます。

WIN_IOMON

I/O モニターウィンドウを閉じます。

WIN_TASKMGR

タスクマネージャーウィンドウを閉じます。

WIN_FORCEMON

フォースモニターウィンドウを閉じます。

WIN_SIMULATOR

シミュレーターウィンドウを閉じます。

注記

表示していないフォームに対してはGCloseステートメントは使用しないでください。

参照

[GShow ステートメント](#), [GShowDialog ファンクション](#)

使用例

```
GClose frmSetup
```

```
GClose WIN_TASKMGR
```

8.9.2 GGet ステートメント

説明

SPEL+プログラムからフォームやコントロールのプロパティを取得するためのステートメントです。

使い方

```
GGet Form .Property, var  
GGet Form .Control.Property, var
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

Control

コントロールの名前、またはコントロールの名前を含む文字列を指定します。そのコントロールは、指定されたフォーム上に存在しなくてはなりません。

Property

取得するプロパティの名前を指定します。

var

プロパティの値を取得する変数を指定します。

注記

GGetステートメントは、実行時にプロパティの値を取得するために使います。プロパティに256文字以上の文字列が入力されている場合、先頭255文字が変数varに格納されます。

参照

[GSet ステートメント](#)

8.9.3 GraphicsEnabled プロパティ

適用

VideoBox

説明

VideoBoxコントロールに、画像処理のグラフィックスを表示するかを、設定・取得します。

使い方

```
GGet Form.Control.GraphicsEnabled, var  
GSet Form.Control.GraphicsEnabled, value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

Control

コントロールの名前、またはコントロールの名前を含む文字列を指定します。

var

プロパティの値を取得する論理変数を指定します。

value

プロパティに設定する論理式を指定します。

値

False

True

デフォルト: False

参照

[ビデオボックスコントロール, VideoEnabled プロパティ](#)

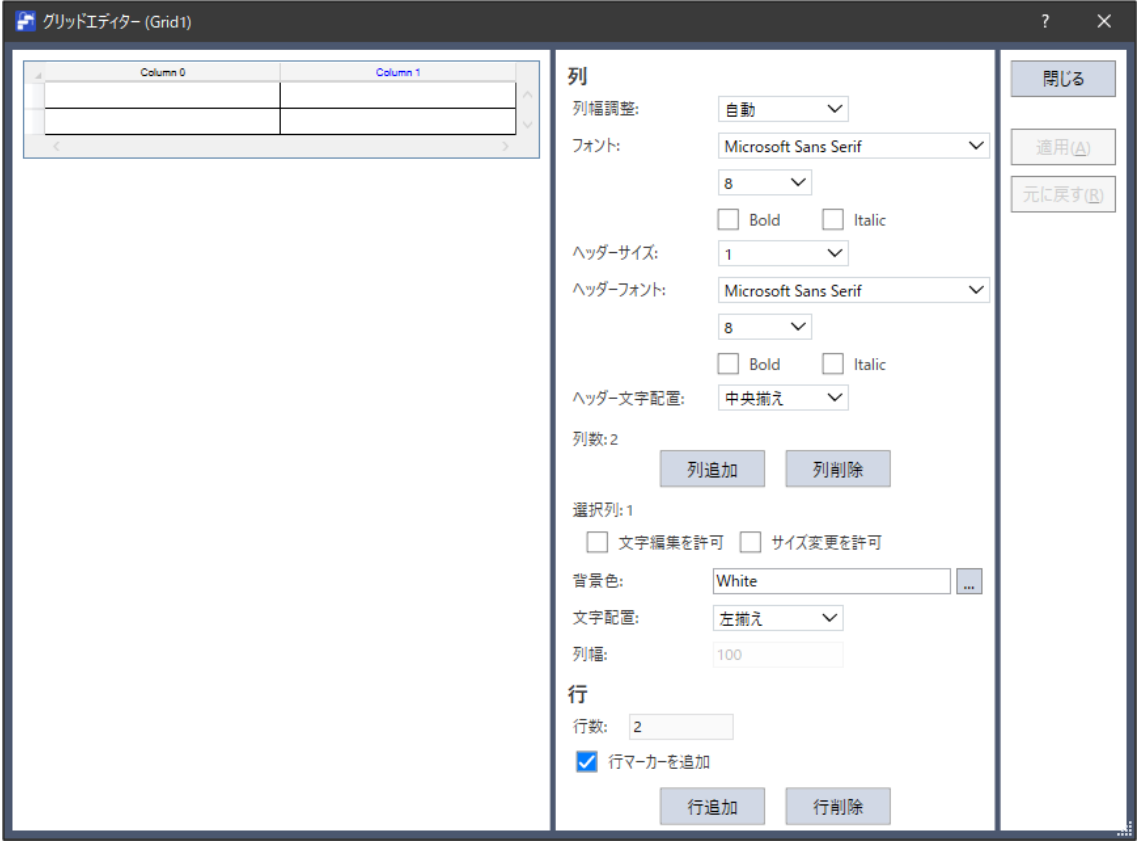
使用例

```
GSet frmMain.VideoBox1.GraphicsEnabled, True
```

8.9.4 GridEditor プロパティ

適用
Grid

説明
グリッドコントロールをセットアップするため、[GridEditor]ダイアログを開きます。



使い方
GridEditorプロパティは、デザイン時のみの使用できます。ユーザーの要望に合わせ、グリッドを設定できるようにします。

設定	説明
列追加	グリッドの最後に、列を追加します。 最小値: 1 最大値: 25
行追加	グリッドの下部に、行を追加します。 最小値: 0 最大値: 10000
文字編集を許可	ユーザーが、選択した列のセルを編集できるかを設定します。
サイズ変更を許可	ユーザーが、選択した列のサイズを変更できるかを設定します。
背景色	選択した列の背景色を設定します。

設定	説明
文字配置	セルテキストの配置を指定します。 左揃え: セルテキストを列の左側に揃えます。 中央揃え: セルテキストを列の中央に揃えます。 右揃え: セルテキストを列の右側に揃えます。
フォント	列のヘッダーのフォントを設定します。
ヘッダー文字配置	列のヘッダーの配置を設定します。 左揃え: 列名を列の左側に揃えます。 中央揃え: 列名を列の中央に揃えます。 右揃え: 列名を列の右側に揃えます。
ヘッダーサイズ	列のヘッダーで、利用できる行数を設定します。 最小値: 1 最大値: 3
列削除	選択した列を削除します。
行削除	選択した行を削除します。
列幅調整	列の追加、削除、サイズ変更するとき、グリッド内の列のサイズ変更方法を決定します。 自動: グリッドコントロールに合わせて、自動で列のサイズを変更します。 手動: 各列をそれぞれの列幅に変更します。 同期: 列のサイズ変更をするとき、すべての列を同じサイズに変更します。
行数	グリッドの行数を設定・表示します。手動で値を変更するには、該当の行に入力して、[適用]ボタンをクリックします。
行マーカを表示	グリッドの左側にある、行選択を表示するかを設定します。
列幅	選択した列の幅を設定・表示します。手動で値を変更するには、列幅調整を自動にしないでください。

注記

[GridEditor]ダイアログを使って、グリッドと初回の表示方法を設定します。設計時にセルにテキストを追加したり、データの追加ができます。

列は、ドラッグ&ドロップで移動したい位置に動かせます。列のヘッダーにある、ドロップダウン矢印を使って、他の列と交換することもできます。

参照

[グリッドコントロール](#)

8.9.5 GSet ステートメント

説明

SPEL+プログラムからフォームやコントロールのプロパティを設定するためのステートメントです。

使い方

```
GSet Form .Property, value  
GSet Form .Control.Property, value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

Control

コントロールの名前、またはコントロールの名前を含む文字列を指定します。そのコントロールは、指定されたフォーム上に存在しなくてはなりません。

Property

設定するプロパティの名前を指定します。

value

プロパティの値を設定する式を指定します。

注記

GSetステートメントは、実行時にプロパティの値を設定するために使います。

参照

[GGet ステートメント](#)

8.9.6 GShow ステートメント

適用

Form

説明

フォームをモードレス ウィンドウとして表示します。

使い方

```
GShow Form
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。また、フォームIDを指定して、以下のシステムウィンドウを表示することも出来ます。

WIN_IOMON

I/O モニターウィンドウを開きます。

WIN_TASKMGR

タスクマネージャウィンドウを開きます。

WIN_FORCEMON

フォースモニターウィンドウを開きます。

WIN_SIMULATOR

シミュレーターウィンドウを開きます。

参照

[GClose ステートメント](#), [GShowDialog ファンクション](#)

使用例

```
GShow frmIODiags
```

```
GShow WIN_TASKMGR
```

8.9.7 GShowDialog ファンクション

適用

Form

説明

フォームをモーダル ダイアログとして表示し、DialogResultの値を返します。

使い方

```
GShowDialog(Form)
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

戻り値

DialogResultプロパティの値を返します。

参照

[GShow ステートメント](#), [GShowDialog ステートメント](#)

使用例

```
result = GShowDialog(frmSetup)
If result = DialogResult_OK Then
    Call SaveSettings
EndIf
```

8.9.8 GShowDialog ステートメント

適用

Form

説明

DialogResultの値を返さずに、フォームをモーダル ダイアログとして表示します。

使い方

```
GShowDialog Form
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

参照

[GShow ステートメント](#), [GShowDialog ファンクション](#)

使用例

```
GShowDialog frmInfoDisplay
```

8.10 H

8.10.1 Height プロパティ

適用

Form, Button, Label, TextBox, RadioButton, CheckBox, ListBox, ComboBox, PictureBox, GroupBox, VideoBox, LED, ProgressBar, TrackBar, StatusBar, Grid, TabControl, Wizard, TreeView

説明

フォームやコントロールの高さをピクセル値で設定・取得します。

使い方

```
GGet Form.[Control].Height, var  
GSet Form.[Control].Height, value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

Control

コントロールの名前、またはコントロールの名前を含む文字列を指定します。

var

プロパティの値を取得する整数変数を指定します。

value

プロパティに設定する整数式を指定します。

値

高さを表すピクセル整数値です。

参照

フォーム, ボタンコントロール, ラベルコントロール, テキストボックスコントロール, ラジオボタンコントロール, チェックボックスコントロール, リストボックスコントロール, コンボボックスコントロール, ピクチャーボックスコントロール, グループボックスコントロール, ビデオボックスコントロール, LEDコントロール, プログレスバーコントロール, トラックバーコントロール, ステータスバーコントロール, タブコントロール, ウィザードコントロール, ツリービューコントロール, Left プロパティ, Top プロパティ, Width プロパティ

使用例

```
GSet frmMain.btnOK.Height, 48
```

8.10.2 HelpButton プロパティ

適用

Form

説明

フォームのタイトルバーに[Help]ボタンを表示するかを設定・取得します。

使い方

```
GGet Form.HelpButton, var  
GSet Form.HelpButton, value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

var

プロパティの値を取得する論理変数を指定します。

value

プロパティに設定する論理式を指定します。

値

False

True

デフォルト: False

注記

このプロパティを Trueに設定すると、クエスションマークの小さなボタンが、タイトルバーの[閉じる]ボタンの左に表示されます (MaximizeBox, MinimizeBoxプロパティがFalseのとき)。このボタンを使って、ヘルプを表示することができます。

参照

[フォーム](#), [HelpID プロパティ](#)

使用例

```
GSet frmMain.HelpButton, True
```

8.10.3 HelpID プロパティ

適用

Form

説明

ヘルプファイルのトピックのHelpIDを設定・取得します。

使い方

```
GGet Form.HelpID, var  
GSet Form.HelpID, value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

var

プロパティの値を取得する整数変数を指定します。

value

プロパティに設定する整数式を指定します。

値

ヘルプファイルのトピックIDの整数値です。(0 ~ 999999)

デフォルト: 0

注記

HelpIDは、文脈依存のヘルプを表示させるために使います。ヘルプファイルは、Epson RC+メニュー-[プロジェクト]-[プロパティ]-[GUI Builder]から設定します。

HelpIDに0以外が設定されている場合、F1キーを押すか、フォームのヘルプボタンをクリックすると (HelpButtonプロパティがTrueに設定されているとき)、ヘルプトピックを表示できます。

参照

[フォーム](#), [HelpButton プロパティ](#)

使用例

```
GSet frmMain.HelpID, 50
```

8.11 I

8.11.1 Iconプロパティ

適用

Form

説明

フォームのタイトルバーのアイコンに表示するイメージを設定・取得します。

使い方

```
GGet Form.Icon, var  
GSet Form.Icon, value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

var

プロパティの値を取得する文字列変数を指定します。

value

プロパティに設定する文字列式を指定します。

値

イメージファイルのフルパスの文字列です。

デフォルト: 空白

参照

[フォーム](#)

使用例

```
GSet frmMain.Icon, "C:\EpsonRC80\GUI\Icons\LedGreen.ico"
```

8.11.2 Image プロパティ

適用

Button, Label, RadioButton, CheckBox, PictureBox

説明

コントロールに表示するイメージを設定・取得します。

使い方

```
GGet Form.Control.Image, var  
GSet Form.Control.Image, value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

Control

コントロールの名前、またはコントロールの名前を含む文字列を指定します。

var

プロパティの値を取得する文字列変数を指定します。

value

プロパティに設定する文字列式を指定します。

値

イメージファイルのフルパスの文字列です。

デフォルト: 空白

参照

[ボタンコントロール](#), [ラベルコントロール](#), [ラジオボタンコントロール](#), [チェックボックスコントロール](#), [ピクチャーボックスコントロール](#), [ImageAlign プロパティ](#), [ImageIndex プロパティ](#)

使用例

```
GSet frmMain.btnTools.Image, "c:\Images\Tools.bmp"
```

8.11.3 ImageAlign プロパティ

適用

Button, Label, RadioButton, CheckBox, LED

説明

コントロールに表示するイメージの配置を設定・取得します。

使い方

```
GGet Form.Control.ImageAlign, var  
GSet Form.Control.ImageAlign, value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

Control

コントロールの名前、またはコントロールの名前を含む文字列を指定します。

var

プロパティの値を取得する整数変数を指定します。

value

プロパティに設定する整数式を指定します。

値

- 1 - TopLeft
- 2 - TopCenter
- 3 - TopRight
- 4 - MiddleLeft
- 5 - MiddleCenter
- 6 - MiddleRight
- 7 - BottomLeft
- 8 - BottomCenter
- 9 - BottomRight

デフォルト:

- 5 - MiddleCenter (Button, Label, RadioButton, CheckBox)
- 4 - MiddleLeft (LED)

参照

[ボタンコントロール](#), [ラベルコントロール](#), [ラジオボタンコントロール](#), [チェックボックスコントロール](#), [LEDコントロール](#), [Image プロパティ](#), [TextAlign プロパティ](#)

使用例

```
GSet frmMain.btnTools.ImageAlign, IMAGEALIGN_MIDDLECENTER
```

8.11.4 ImageOff プロパティ

適用

LED

説明

I/Oがオフのときに表示するイメージを設定・取得します。

使い方

```
GGet Form.Control.ImageOff, var  
GSet Form.Control.ImageOff, value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

Control

コントロールの名前、またはコントロールの名前を含む文字列を指定します。

var

プロパティの値を取得する文字列変数を指定します。

value

プロパティに設定する文字列式を指定します。

値

イメージファイルのフルパスの文字列です。拡張子が、bmp、ico、jpeg、gif、png のファイルを指定できます。

デフォルト: LedOff.ico

参照

[LEDコントロール, ImageOn プロパティ](#)

使用例

```
GSet frmMain.Led1.ImageOff, "c:\EpsonRC80\GUI\Icons\LedOff.ico"
```

8.11.5 ImageOn プロパティ

適用

LED

説明

I/Oがオンのときに表示するイメージを設定・取得します。

使い方

```
GGet Form.Control.ImageOn, var  
GSet Form.Control.ImageOn, value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

Control

コントロールの名前、またはコントロールの名前を含む文字列を指定します。

var

プロパティの値を取得する文字列変数を指定します。

value

プロパティに設定する文字列式を指定します。

値

イメージファイルのフルパスの文字列です。拡張子が、bmp、ico、jpeg、gif、png のファイルを指定できます。

デフォルト: LedRed.ico

参照

[LEDコントロール, ImageOff プロパティ](#)

使用例

```
GSet frmMain.Led1.ImageOn, "c:\EpsonRC80\GUI\Icons\LedRed.ico"
```

8.11.6 ImageIndex プロパティ

適用

PictureBox

説明

コントロールに表示するイメージのインデックス番号を設定・取得します。

使い方

```
GGet Form.Control.ImageIndex, var  
GSet Form.Control.ImageIndex, value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

Control

コントロールの名前、またはコントロールの名前を含む文字列を指定します。

var

プロパティの値を取得する整数変数を指定します。

value

プロパティに設定する整数式を指定します。

本プロパティは、RC800コントローラー接続時のみにGGet, GSetにて使用可能です。

値

コントロールに追加されているイメージファイルのインデックス番号です。デフォルト: -1

参照

[ピクチャーボックスコントロール, Image プロパティ](#)

使用例

```
' 現在設定されているイメージファイルのインデックス番号を取得  
GGet frmMain.pictureBox1.ImageIndex, index  
' イメージファイルのインデックス番号を2に設定。  
GSet frmMain.pictureBox1.ImageIndex, 2
```

8.11.7 Interval プロパティ

適用

Timer

説明

タイマー間隔をmsec (ミリ秒)単位で設定・取得します。

使い方

```
GGet Form.Control.Interval, var  
GSet Form.Control.Interval, value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

Control

コントロールの名前、またはコントロールの名前を含む文字列を指定します。

var

プロパティの値を取得する整数変数を指定します。

value

プロパティに設定する整数式を指定します。

値

タイマー間隔をmsec (ミリ秒)で指定します。1~9999の整数を設定します。

デフォルト: 100

注記

タイマー間隔を秒で取得したいときは、1000で割ってください。

参照

[タイマーコントロール, Enabled プロパティ](#)

使用例

```
GSet frmMain.tmrMain.Interval, 500
```

8.11.8 IOBit プロパティ

適用

LED

説明

状態を表示したいI/Oビットを設定・取得します。

使い方

```
GGet Form.Control.IOBit, var  
GSet Form.Control.IOBit, value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

Control

コントロールの名前、またはコントロールの名前を含む文字列を指定します。

var

プロパティの値を取得する整数変数を指定します。

value

プロパティに設定する整数式を指定します。

値

I/Oビットを表す整数値です。(範囲: 0~9999)

デフォルト: 0

参照

[LEDコントロール](#), [IOType プロパティ](#)

使用例

```
GSet frmMain.Led1.IOBit, 10
```

8.11.9 IOType プロパティ

適用

LED

説明

状態を表示したいI/Oの入出力タイプを設定・取得します。

使い方

```
GGet Form.Control.IOType, var  
GSet Form.Control.IOType, value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

Control

コントロールの名前、またはコントロールの名前を含む文字列を指定します。

var

プロパティの値を取得する整数変数を指定します。

value

プロパティに設定する整数式を指定します。

値

0 - Input

1 - Output

2 - Memory

デフォルト:Input

参照

[LEDコントロール, IOBit プロパティ](#)

使用例

```
GSet frmMain.Led1.IOType, IOTYPE_OUTPUT
```

8.12 K

8.12.1 KeyPressイベント

適用

TextBox

説明

キーが押されたときに実行されるイベントです。

使い方

```
Form_Control_KeyPress (Sender$ As String, ByRef Key$ As String)
```

Sender\$

イベントを発生させたコントロールの名前が設定されます。

Key\$

押されたキーの文字列です。

注記

KeyPressイベントは、ユーザーのキー入力を直接読みたいときに使います。

また、このイベントをフィルターとして使うことも可能です。Key\$に他の文字を代入すると、入力されたキー文字列を変更することができます。Key\$に空の文字列を代入すると、キー入力をキャンセルすることができます。

参照

[テキストボックスコントロール](#)

使用例

```
Function frmMain_txtSpeed_KeyPress(Sender$ As String, ByRef Key$ As String)
    ' 数値のみ入力可能にする
    If Instr("0123456789" Key$) < 0 Then
        Key$ = ""
    EndIf
End
Fend
```

8.13 L

8.13.1 LargeChange プロパティ

適用

TrackBar

説明

マウスのクリックや、Page UpおよびPage Downキーを押す時、スライダーが移動するポジションの数を設定・取得します。

使い方

```
GGet Form.Control.LargeChange, var  
GSet Form.Control.LargeChange, value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

Control

コントロールの名前、またはコントロールの名前を含む文字列を指定します。

var

プロパティの値を取得する整数変数を指定します。

value

プロパティに設定する整数式を指定します。

値

スライダーが移動するポジションの数を表す整数値です。

参照

[トラックバーコントロール, SmallChange プロパティ](#)

使用例

```
GSet frmMain.TrackBar1.LargeChange, 10
```

8.13.2 Left プロパティ

適用

Form, Button, Label, TextBox, RadioButton, CheckBox, ListBox, ComboBox, PictureBox, GroupBox, VideoBox, LED, ProgressBar, TrackBar, StatusBar, Grid, TabControl, Wizard, TreeView

説明

フォームやコントロールの左側の座標をピクセル値で設定・取得します。

使い方

```
GGet Form.[Control].Left, var  
GSet Form.[Control].Left, value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

Control

コントロールの名前、またはコントロールの名前を含む文字列を指定します。

var

プロパティの値を取得する整数変数を指定します。

value

プロパティに設定する整数式を指定します。

値

左側の座標を表すピクセル整数値です。

参照

フォーム, ボタンコントロール, ラベルコントロール, テキストボックスコントロール,
ラジオボタンコントロール, チェックボックスコントロール, リストボックスコントロール,
コンボボックスコントロール, ピクチャーボックスコントロール, グループボックスコントロール,
ビデオボックスコントロール, LEDコントロール, プログレスバーコントロール,
トラックバーコントロール, ステータスバーコントロール, タブコントロール, ウィザードコントロール,
ツリービューコントロール, Top プロパティ, Height プロパティ, Width プロパティ

使用例

```
GSet frmMain.btnOK.Left, 200
```

8.13.3 List プロパティ

適用

ListBox, ComboBox

説明

指定されたコントロールのリストアイテムの値を文字列で設定・取得します。

使い方

```
GGet Form.Control.List(index), var  
GSet Form.Control.List(index), value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

Control

コントロールの名前、またはコントロールの名前を含む文字列を指定します。そのコントロールは、指定されたフォーム上に存在しなくてはなりません。

index

0から始まるリストのインデックス番号を、整数式で指定します。

var

プロパティの値を取得する 文字列変数 を指定します。

value

プロパティに設定する文字列式を指定します。

値

リストアイテムの文字列です。

注記

リストボックスやコンボボックスにAddItemプロパティでアイテムを追加したあとに、アイテムの値を取得したり変更したりすることができます。

空の文字列を設定することで、アイテムを削除することができます。

参照

[コンボボックスコントロール](#), [リストボックスコントロール](#), [ListCount プロパティ](#), [AddItem プロパティ](#)

使用例

```
String part$GGet frmMain.lstPartNames.List(0), part$
```

8.13.4 ListCount プロパティ

適用

ListBox, ComboBox

説明

リストボックスやコンボボックスのアイテム数を設定・取得します。

使い方

```
GGet Form.Control.ListCount, var  
GSet Form.Control.ListCount, value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

Control

コントロールの名前、またはコントロールの名前を含む文字列を指定します。そのコントロールは、指定されたフォーム上に存在しなくてはなりません。

var

プロパティの値を取得する整数変数を指定します。

value

プロパティに設定する 整数式 を指定します。

値

リストのアイテム数です。

注記

ListCountを“0”に設定することでListBoxやComboBoxのアイテムを消去できます。ListCountにリストのアイテム数を超える値を設定すると、追加分のアイテムに空のテキストが追加されます。

参照

[コンボボックスコントロール](#), [リストボックスコントロール](#), [AddItem プロパティ](#), [List プロパティ](#)

使用例

```
Integer count  
・ リスト内のアイテム数を取得  
GGet frmMain.lstPartNames.ListCount, count  
  
・ リスト内のアイテムを消去  
GSet frmMain.lstPartName.ListCount, 0
```

8.13.5 Loadイベント

適用

Form

説明

フォームが表示される前に発生するイベントです。

使い方

```
Form_Load (Sender$ As String)
```

Sender\$

イベントが発生したフォームの名前です。

注記

フォームが表示される前に、SPEL+プログラムを実行するときに使います。

参照

[フォーム](#), [Resizeイベント](#), [Closed イベント](#)

使用例

```
Function frmMain_Load(Sender$ As String)
    GSet frmMain.txtSpeed.Text, Str$(g_RobotSpeed)
End
```

8.14 M

8.14.1 MaximizeBox プロパティ

適用

Form

説明

フォームのタイトルバーに[最大化]ボタンを表示するかを、設定・取得します。

使い方

```
GGet Form.MaximizeBox, var  
GSet Form.MaximizeBox, value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

var

プロパティの値を取得する論理変数を指定します。

value

プロパティに設定する論理式を指定します。

値

False

True

デフォルト: False

注記

フォームを画面いっぱいに最大化するときに使います。

参照

フォーム, [MinimizeBox プロパティ](#)

使用例

```
GSet frmMain.MaximizeBox, True
```

8.14.2 Maximum プロパティ

適用

ProgressBar, TrackBar

説明

コントロールの処理範囲の最大値を設定・取得します。

使い方

```
GGet Form.Control.Maximum, var  
GSet Form.Control.Maximum, value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

Control

コントロールの名前、またはコントロールの名前を含む文字列を指定します。そのコントロールは、指定されたフォーム上に存在しなくてはなりません。

var

プロパティの値を取得する整数変数を指定します。

value

プロパティに設定する 整数式 を指定します。

値

処理範囲の最大値を表す整数値です。

参照

[プログレスバーコントロール](#), [トラックバーコントロール](#), [Minimum プロパティ](#), [Value プロパティ](#)

使用例

```
GSet frmMain.TrackBar1.Maximum, 60
```

8.14.3 MinimizeBox プロパティ

適用

Form

説明

フォームのタイトルバーに[最小化]ボタンを表示するかを、設定・取得します。

使い方

```
GGet Form.MinimizeBox, var  
GSet Form.MinimizeBox, value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

var

プロパティの値を取得する論理変数を指定します。

value

プロパティに設定する論理式を指定します。

値

False

True

デフォルト: False

注記

フォームを最小化するときに使います。

参照

フォーム, [MaximizeBox プロパティ](#)

使用例

```
GSet frmMain.MinimizeBox, True
```

8.14.4 Minimum プロパティ

適用

ProgressBar, TrackBar

説明

コントロールの処理範囲の最小値を設定・取得します。

使い方

```
GGet Form.Control.Minimum, var  
GSet Form.Control.Minimum, value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

Control

コントロールの名前、またはコントロールの名前を含む文字列を指定します。そのコントロールは、指定されたフォーム上に存在しなくてはなりません。.

var

プロパティの値を取得する整数変数を指定します。

value

プロパティに設定する 整数式 を指定します。

値

処理範囲の最小値を表す整数値です。

参照

[プログレスバーコントロール](#), [トラックバーコントロール](#), [Maximum プロパティ](#), [Value プロパティ](#)

使用例

```
GSet frmMain.TrackBar1.Minimum, 10
```

8.14.5 MultiLine プロパティ

適用

TextBox

説明

テキストボックスコントロールを複数行にするかを設定・取得します。

使い方

```
GGet Form.Control.MultiLine, var  
GSet Form.Control.MultiLine, value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

Control

コントロールの名前、またはコントロールの名前を含む文字列を指定します。そのコントロールは、指定されたフォーム上に存在しなくてはなりません。

var

プロパティの値を取得する論理変数を指定します。

value

プロパティに設定する論理式を指定します。

値

False

True

デフォルト: False

注記

テキストボックスコントロールを複数行で使うときに、MultiLineプロパティをTrueに設定します。

参照

[テキストボックスコントロール, ScrollBars プロパティ](#)

使用例

```
GSet frmMain.txttStatus.MultiLine, True
```

8.15 N

8.15.1 Name プロパティ

適用

Form, Button, Label, TextBox, RadioButton, CheckBox, ListBox, ComboBox, PictureBox, GroupBox, Timer, VideoBox, LED, ProgressBar, TrackBar, StatusBar, Grid, TabControl, Wizard, TreeView

説明

設計時は、フォームやコントロールの名前を設定します。実行時は、名前を取得します。

使い方

```
GGet Form.[Control].Name, var
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

Control

コントロールの名前、またはコントロールの名前を含む文字列を指定します。

var

プロパティの値を取得する文字列変数を指定します。

値

フォームやコントロールの名前を含んだ文字列です。次の名前は使用することができません。

- 名前の先頭に数字を使用したフォーム、またはコントロールの名前
- 編集しているプロジェクト名称と同じフォームの名前

参照

フォーム, ボタンコントロール, ラベルコントロール, テキストボックスコントロール,
ラジオボタンコントロール, チェックボックスコントロール, リストボックスコントロール,
コンボボックスコントロール, ピクチャーボックスコントロール, グループボックスコントロール,
タイマーコントロール, ビデオボックスコントロール, LEDコントロール, プログレスバーコントロール,
トラックバーコントロール, ステータスバーコントロール, タブコントロール, ウィザードコントロール,
ツリービューコントロール

使用例

```
String name$  
GGet frmMain.btnOK.Name, name$
```

8.16 O

8.16.1 Orientation プロパティ

適用

ProgressBar, TrackBar

説明

コントロールの向きを設定・取得します。

使い方

```
GGet Form.Control.Orientation, var  
GSet Form.Control.Orientation, value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

Control

コントロールの名前、またはコントロールの名前を含む文字列を指定します。

var

プロパティの値を取得する 整数変数 を指定します。

value

プロパティに設定する 整数式 を指定します。

値

0 - Horizontal

1 - Vertical

デフォルト:Horizontal

参照

[プログレスバーコントロール](#), [トラックバーコントロール](#)

使用例

```
GSet frmMain.ProgressBar1.Orientation, ORIENT_VERTICAL
```

8.17 P

8.17.1 Parent プロパティ

適用

TreeView

説明

ツリービューの追加済みの項目(ノード)について親ノードを取得・設定します。

使い方

```
GGet Form.Control.Parent(name), var  
GSet Form.Control.Parent(name), value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

Control

コントロールの名前、またはコントロールの名前を含む文字列を指定します。そのコントロールは、指定されたフォーム上に存在しなくてはなりません。

name

親ノードを設定する子ノードを示す文字列式を指定します。

var

プロパティの値を取得する文字列変数を指定します。

value

プロパティに設定する文字列式を指定します。

値

親ノードを示す文字列式を指定します。

参照

[ツリービューコントロール](#), [AddNode プロパティ](#), [RemoveNode プロパティ](#), [Text プロパティ](#)

使用例

```
GSet frmMain.lstModels.AddNode, "Model1"  
GSet frmMain.lstModels.AddNode, "Model1-1"  
GSet frmMain.lstModels.Parent("Model1-1"), "Model1"
```

8.17.2 PasswordChar プロパティ

適用

TextBox

説明

1行のテキストボックスで、パスワード入力の際に表示する文字を設定・取得します。

使い方

```
GGet Form.Control.PasswordChar, var  
GSet Form.Control.PasswordChar, value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

Control

コントロールの名前、またはコントロールの名前を含む文字列を指定します。そのコントロールは、指定されたフォーム上に存在しなくてはなりません。

var

プロパティの値を取得する文字列変数を指定します。

value

プロパティに設定する文字列式を指定します。

値

マスクに使う文字を含む文字列です。

注記

パスワード入力時に入力した文字を隠すための文字です。パスワード入力時には、この文字が表示されます。MultiLineプロパティは、Falseに設定してください。

参照

[テキストボックスコントロール](#)

使用例

```
GSet frmMain.txtPassword.PasswordChar, "*"
```

8.17.3 PressDelay プロパティ

適用

Button

説明

ボタンコントロールをクリックイベントが実行されるまでの時間を設定します。

使い方

本プロパティはSPEL+プログラムからの設定には対応していません。

対象のボタンコントロールのプロパティグリッドに値を指定し、設定してください。

値

クリック開始からイベントを発行するまでの時間をmsec (ミリ秒)で指定します。0～9999の整数を設定します。

デフォルト:即時イベント実行

注記

設定した時間が経過する前にクリックを離すとイベントは実行されません。

参照

[ボタンコントロール](#)

8.17.4 PressSound プロパティ

適用

Button

説明

ボタンコントロールをクリックした際にビープ音を鳴動するかを設定します。

使い方

本プロパティはSPEL+プログラムからの設定には対応していません。
対象のボタンコントロールのプロパティグリッドから設定してください。

値

False

True

デフォルト : False

参照

[ボタンコントロール](#)

8.17.5 ProgressBarStyle プロパティ

適用

ProgressBar

説明

プログレスバーのスタイルを設定・取得します。

使い方

```
GGet Form.Control.ProgressBarStyle, var  
GSet Form.Control.ProgressBarStyle, value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

Control

コントロールの名前、またはコントロールの名前を含む文字列を指定します。

var

プロパティの値を取得する 整数変数 を指定します。

value

プロパティに設定する 整数式 を指定します。

値

1 - Continuous

2 - Marquee

デフォルト: Continuous

参照

[プログレスバーコントロール](#)

使用例

```
GSet frmMain.ProgressBar1.ProgressBarStyle, PROGRESSBAR_STYLE_MARQUEE
```

8.18 R

8.18.1 ReadOnly プロパティ

適用

TextBox

説明

テキストボックスを編集不可にするかどうかを設定・取得します。

使い方

```
GGet Form.Control.ReadOnly, var  
GSet Form.Control.ReadOnly, value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

Control

コントロールの名前、またはコントロールの名前を含む文字列を指定します。そのコントロールは、指定されたフォーム上に存在しなくてはなりません。

var

プロパティの値を取得する論理変数を指定します。

value

プロパティに設定する論理式を指定します。

値

False

True

デフォルト: False

注記

オペレーターにテキストボックスを編集させたくない場合に使用します。

参照

[テキストボックスコントロール](#)

使用例

```
GSet frmMain.txtSpeed.ReadOnly, True
```

8.18.2 RemoveNode プロパティ

適用

TreeView

説明

ツリービューの項目(ノード)から項目を削除します。指定された項目の子ノードも削除されます。

使い方

```
GSet Form.Control.RemoveNode, value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

Control

コントロールの名前、またはコントロールの名前を含む文字列を指定します。そのコントロールは、指定されたフォーム上に存在しなくてはなりません。

value

削除される項目(ノード)を示す文字列式を指定します。

値

削除する項目(ノード)を示す文字列式を指定します。

参照

ツリービューコントロール, [AddNode プロパティ](#), [Parent プロパティ](#), [Text プロパティ](#)

使用例

```
GSet frmMain.lstModels.RemoveNode, "Model1"
```

8.18.3 RemoveRow プロパティ

適用

Grid

説明

グリッドコントロールから、行を削除します。

使い方

```
GSet Form.Control.RemoveRow, value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

Control

コントロールの名前、またはコントロールの名前を含む文字列を指定します。そのコントロールは、指定されたフォーム上に存在しなくてはなりません。

value

削除される行のインデックスを表す整数値です。

値

削除される行のインデックスを表す整数値です。

参照

[AddRow プロパティ](#), [グリッドコントロール](#), [RowCount プロパティ](#)

使用例

```
GSet frmMain.Grid01.RemoveRow, 0
```

8.18.4 Resizeイベント

適用

Form

説明

フォームのサイズが変更されたときに発生するイベントです。

使い方

```
Form_Resize (Sender$ As String)
```

Sender\$

イベントを発生させたフォームの名称が入ります。

参照

[フォーム](#), [Loadイベント](#), [Closed イベント](#)

使用例

```
Function frmMain_Resize(Sender$ As String)
    Integer width
    GGet frmMain.Width, width
    GSet frmMain.btnOK.Left, width / 2 - 100
    GSet frmMain.btnCancel.Left, width / 2 + 10
Fend
```

8.18.5 RobotNumber プロパティ

適用

StatusBar

説明

ShowRobotプロパティが有効の場合、表示したいロボット情報のロボット番号を設定・取得します。

使い方

```
GGet Form.Control.RobotNumber, var  
GSet Form.Control.RobotNumber, value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

Control

コントロールの名前、またはコントロールの名前を含む文字列を指定します。そのコントロールは、指定されたフォーム上に存在しなくてはなりません。

var

プロパティの値を取得する整数変数を指定します。

value

プロパティに設定する整数式を指定します。

値

ロボット番号を表す整数値です。

デフォルト: 1

参照

[ステータスバーコントロール, ShowRobot プロパティ](#)

使用例

```
GSet frmMain.StatusBar1.RobotNumber, 2
```

8.18.6 RowCount プロパティ

適用

Grid

説明

グリッドコントロールの行数を設定・取得します。
現在の行数より大きな値を設定した場合、新しい行が末尾に追加されます。
現在の行数より小さな値を設定した場合、末尾から行が削除されます。

使い方

```
GGet Form.Control.RowCount, var  
GSet Form.Control.RowCount, value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

Control

コントロールの名前、またはコントロールの名前を含む文字列を指定します。そのコントロールは、指定されたフォーム上に存在しなくてはなりません。

var

コントロールの行数を含む整数変数を指定します。

value

コントロールの行の総数を表す整数式を指定します。

値

グリッドの行数を指定する整数値です。

参照

[AddRow プロパティ](#)、[グリッドコントロール](#)、[RemoveRow プロパティ](#)、[RowCount プロパティ](#)

使用例

```
Integer count  
GSet frmMain.Grid01.RowCount, 50  
GGet frmMain.Grid01.RowCount, count
```

8.19 S

8.19.1 Scrollイベント

適用

TrackBar

説明

スライダーが移動したときに発生するイベントです。

使い方

```
Form_Control_Scroll (Sender$ As String)
```

Sender\$

イベントが発生させたコントロールの名前が設定されます。

参照

[トラックバーコントロール](#)

使用例

```
Function frmMain_TrackBar1_Scroll(Sender$ As String, Value As Integer)
    Print "The trackbar value after scroll is", Value
End
```

8.19.2 ScrollBars プロパティ

適用

TextBox, Grid

説明

複数行のテキストボックスに、スクロールバーを表示させるかを設定・取得します。

使い方

```
GGet Form.Control.ScrollBars, var  
GSet Form.Control.ScrollBars, value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

Control

コントロールの名前、またはコントロールの名前を含む文字列を指定します。そのコントロールは、指定されたフォーム上に存在しなくてはなりません。

var

プロパティの値を取得する整数変数を指定します。

value

プロパティに設定する整数式を指定します。

値

0 - None

1 - Horizontal

2 - Vertical

3 - Both

デフォルト:None

注記

スクロールバーを使うときは、MultiLineプロパティをTrueに設定してください。

水平スクロールバーを表示するときは、WordWrapプロパティをFalseに設定してください。

参照

[テキストボックスコントロール](#), [MultiLine プロパティ](#), [WordWrap プロパティ](#)

使用例

```
GSet frmMain.txtStatus.Scrollbars, SCROLLBARS_VERT
```

8.19.3 SelectedNodeChanged イベント

適用

TreeView

説明

選択されているノードが変更された時に実行されるイベントです。

使い方

```
Form_Control_SelectedNodeChanged(Sender$ As String, NodeName$ As String)
```

Sender\$

イベントを発生させたコントロールの名前が設定されます。

NodeName\$

選択されたノードの名前です。

参照

[ツリービューコントロール](#)

使用例

```
Function frmMain_TreeView1_SelectedNodeChanged(Sender$ As String, NodeName$ As String)
    Print NodeName$, " was selected."

    '選択されたノードに応じて、ウィザードコントロールの表示ページを切り替える
    Integer wizardIndex
    Select NodeName$
        Case "TreeNode1"; wizardIndex = 0
        Case "TreeNode2"; wizardIndex = 1
        Case "TreeNode3"; wizardIndex = 2
    Send

    GSet frmMain.Wizard1.SelectedIndex, wizardIndex
End
```

8.19.4 SelectedIndex プロパティ

適用

ListBox, ComboBox, Grid, TabControl, Wizard

説明

現在選択されているアイテムのインデックス番号を設定・取得します。

使い方

```
GGet Form.Control.SelectedIndex, var  
GSet Form.Control.SelectedIndex, value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

Control

コントロールの名前、またはコントロールの名前を含む文字列を指定します。そのコントロールは、指定されたフォーム上に存在しなくてはなりません。

var

プロパティの値を取得する整数変数を指定します。

value

プロパティに設定する整数式を指定します。

値

選択されたアイテムの番号を0から始まる整数で返します。アイテムが選択されていないときは -1 を返します。

注記

SelectedIndexプロパティは、リストボックスやコンボボックスの選択されているアイテムを判断するときに使います。多くは、コントロールのクリックイベント処理の中で使います。

参照

[AddItem プロパティ](#), [コンボボックスコントロール](#), [List プロパティ](#), [リストボックスコントロール](#), [グリッドコントロール](#), [ウィザードコントロール](#)

使用例

```
Integer index  
GGet frmMain.lstParts.SelectedIndex, index
```

8.19.5 SelectionChanged イベント

適用

TabControl, Wizard

説明

選択されているページが変更された時に実行されるイベントです。

使い方

```
Form_Control_SelectionChanged(Sender$ As String, SelectedIndex As Integer,  
PreviousIndex As Integer)
```

Sender\$

イベントを発生させたコントロールの名前が設定されます。

SelectedIndex

選択されたページのインデックス番号です。

PreviousIndex

前に選択されていたページのインデックス番号です。

値

選択されたアイテムの番号を0から始まる整数で返します。アイテムが選択されていないときは -1 を返します。

注記

SelectedIndexプロパティは、リストボックスやコンボボックスの選択されているアイテムを判断するときに使います。多くは、コントロールのクリックイベント処理の中で使います。

参照

[タブコントロール, ウィザードコントロール](#)

使用例

```
Function frmMain_TabControl1_SelectionChanged(Sender$ As String, SelectedIndex  
As Integer, PreviousIndex As Integer)  
    Print "The TabControl selection was changed to ", Str$(SelectedIndex),  
    " from ", Str$(PreviousIndex)  
End
```

8.19.6 ShowDateTime プロパティ

適用

StatusBar

説明

現在日時表示の有無を設定・取得します。

使い方

```
GGet Form.Control.ShowDateTime, var  
GSet Form.Control.ShowDateTime, value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

Control

コントロールの名前、またはコントロールの名前を含む文字列を指定します。そのコントロールは、指定されたフォーム上に存在しなくてはなりません。

var

プロパティの値を取得する論理変数を指定します。

value

プロパティに設定する論理式を指定します。

値

False

True

デフォルト: False

参照

[ステータスバーコントロール](#), [ShowEStop プロパティ](#), [ShowRobot プロパティ](#), [ShowSafeguard プロパティ](#)

使用例

```
GSet frmMain.StatusBar1.ShowDateTime, True
```

8.19.7 ShowEStop プロパティ

適用

StatusBar

説明

Estop状態表示の有無を設定・取得します。

使い方

```
GGet Form.Control.ShowEStop, var  
GSet Form.Control.ShowEStop, value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

Control

コントロールの名前、またはコントロールの名前を含む文字列を指定します。そのコントロールは、指定されたフォーム上に存在しなくてはなりません。

var

プロパティの値を取得する論理変数を指定します。

value

プロパティに設定する論理式を指定します。

値

False

True

デフォルト: False

参照

[ステータスバーコントロール](#), [ShowDateTime プロパティ](#), [ShowRobot プロパティ](#), [ShowSafeguard プロパティ](#)

使用例

```
GSet frmMain.StatusBar1.ShowEStop, True
```

8.19.8 ShowNavigators プロパティ

適用

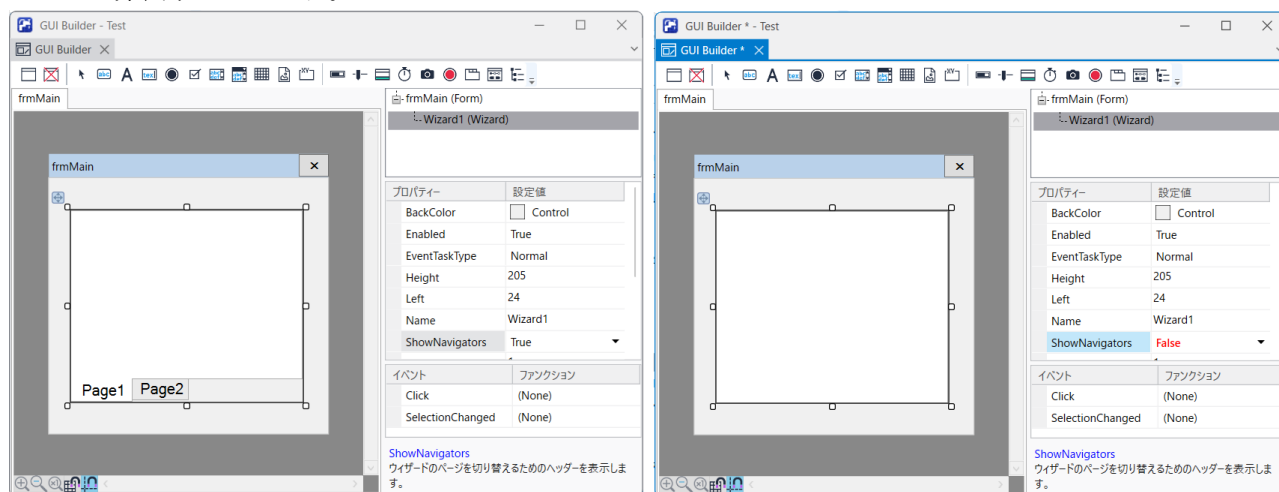
Wizard

説明

ウィザードのページを切り替えるためのヘッダーを表示するかを設定します。

使い方

ShowNavigatorsプロパティをTrueに設定した場合、ヘッダーが表示されます。Falseに設定した場合、ヘッダーが非表示になります。



注記 このプロパティは、デザインエリアでの表示のみに反映されます。フォーム実行時は、このプロパティの設定に関係なくヘッダーは表示されません。

値

False

True

デフォルト: True

参照

[ウィザードコントロール](#)

8.19.9 ShowPrint プロパティ

適用

TextBox

説明

Printステートメントの出力を表示するかを設定・取得します。

使い方

```
GGet Form.Control.ShowPrint, var  
GSet Form.Control.ShowPrint, value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

Control

コントロールの名前、またはコントロールの名前を含む文字列を指定します。そのコントロールは、指定されたフォーム上に存在しなくてはなりません。

var

プロパティの値を取得する論理変数を指定します。

value

プロパティに設定する論理式を指定します。

値

False

True

デフォルト: False

注記

ShowPrintがTrueの場合、Spelタスクで実行されたPrintステートメントのすべての出力がテキストボックスに表示されます。複数のテキストボックスでShowPrintがTrueに設定されている場合、すべてのテキストボックスにプリント出力が表示されます。

参照

[テキストボックスコントロール, ScrollBars プロパティ](#)

使用例

```
GSet frmMain.txtStatus.ShowPrint, True
```

8.19.10 ShowRobot プロパティ

適用

StatusBar

説明

RobotNumber プロパティで指定したロボット表示の有無を設定・取得します。

使い方

```
GGet Form.Control.ShowRobot, var  
GSet Form.Control.ShowRobot, value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

Control

コントロールの名前、またはコントロールの名前を含む文字列を指定します。そのコントロールは、指定されたフォーム上に存在しなくてはなりません。

var

プロパティの値を取得する論理変数を指定します。

value

プロパティに設定する論理式を指定します。

値

False

True

デフォルト: False

参照

[ステータスバーコントロール](#), [RobotNumber プロパティ](#), [ShowDateTime プロパティ](#),
[ShowEStop プロパティ](#), [ShowSafeguard プロパティ](#)

使用例

```
GSet frmMain.StatusBar1.ShowRobot, True
```

8.19.11 ShowSafeguard プロパティ

適用

StatusBar

説明

安全扉状態表示の有無を設定・取得します。

使い方

```
GGet Form.Control.ShowSafeguard, var  
GSet Form.Control.ShowSafeguard, value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

Control

コントロールの名前、またはコントロールの名前を含む文字列を指定します。そのコントロールは、指定されたフォーム上に存在しなくてはなりません。

var

プロパティの値を取得する論理変数を指定します。

value

プロパティに設定する論理式を指定します。

値

False

True

デフォルト: False

参照

[ステータスバーコントロール](#), [ShowDateTime プロパティ](#), [ShowEStop プロパティ](#), [ShowRobot プロパティ](#)

使用例

```
GSet frmMain.StatusBar1.ShowSafeguard, True
```

8.19.12 SizeMode プロパティ

適用

PictureBox

説明

イメージの表示方法を設定、取得します。

使い方

```
GGet Form.Control.SizeMode, var  
GSet Form.Control.SizeMode, value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

Control

コントロールの名前、またはコントロールの名前を含む文字列を指定します。そのコントロールは、指定されたフォーム上に存在しなくてはなりません。

var

プロパティの値を取得する整数変数を指定します。

value

プロパティに設定する整数式を指定します。

値

0 - Normal

1 - StretchImage

2 - AutoSize

3 - CenterImage

4 - Zoom

デフォルト:Normal

注記

Normal (デフォルト): イメージはピクチャーボックスの左上に表示されます。ピクチャーボックスよりイメージが大きい場合、その部分は表示されません。

StretchImage: イメージは、ピクチャーボックスに合わせて拡大縮小されます。

AutoSize: イメージに合わせて、ピクチャーボックスを拡大縮小します。

CenterImage: イメージは、ピクチャーボックスの中央に配置されます。

Zoom: イメージは、ピクチャーボックスに合わせて縦横比を固定して拡大縮小されます。

参照

[ピクチャーボックスコントロール](#), [Image プロパティ](#), [ImageAlign プロパティ](#)

使用例

```
GSet frmMain.picLogo.SizeMode, SIZEMODE_AUTOSIZE
```

8.19.13 SmallChange プロパティ

適用

TrackBar

説明

キーボードの入力 (方向キー) に対して、スライダーが移動するポジションの数を設定・取得します。

使い方

```
GGet Form.Control.SmallChange, var  
GSet Form.Control.SmallChange, value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

Control

コントロールの名前、またはコントロールの名前を含む文字列を指定します。

var

プロパティの値を取得する整数変数を指定します。

value

プロパティに設定する整数式を指定します。

値

スライダーが移動するポジションの数を表す整数値です。

参照

[トラックバーコントロール](#), [LargeChange プロパティ](#)

使用例

```
GSet frmMain.TrackBar1.SmallChange, 5
```

8.19.14 Sorted プロパティ

適用

ComboBox, ListBox

説明

リストボックスやコンボボックスのアイテムを、アルファベット順にソートするかを設定・取得します。数字(全角半角なし)、英字、カタカナ、ひらがな、漢字の順にソートされます。

使い方

```
GGet Form.Control.Sorted, var  
GSet Form.Control.Sorted, value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

Control

コントロールの名前、またはコントロールの名前を含む文字列を指定します。そのコントロールは、指定されたフォーム上に存在しなくてはなりません。

var

プロパティの値を取得する論理変数を指定します。

value

プロパティに設定する論理式を指定します。

値

False

True

デフォルト: False

参照

[コンボボックスコントロール](#), [リストボックスコントロール](#)

使用例

```
GSet frmMain.lstParts.Sorted, True
```

8.19.15 StartPosition プロパティ

適用

Form

説明

フォームを実行するときの開始位置を設定・取得します。

使い方

```
GGet Form.StartPosition, var  
GSet Form.StartPosition, value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

var

プロパティの値を取得する整数変数を指定します。

value

プロパティに設定する整数式を指定します。

値

0 - Manual

1 - CenterScreen

2 - CenterParent

デフォルト:CenterScreen

参照

フォーム, [WindowState プロパティ](#)

使用例

```
GSet frmMain.StartPosition, STARTPOSITION_CENTERSCREEN
```

8.20 T

8.20.1 TabAlignment プロパティ

適用

TabControl

説明

コントロールに表示するタブの配置を設定、取得します。

使い方

```
GGet Form.Control.TabAlignment, var  
GSet Form.Control.TabAlignment, value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

Control

コントロールの名前、またはコントロールの名前を含む文字列を指定します。

var

プロパティの値を取得する整数変数を指定します。

value

プロパティに設定する整数式を指定します。

本プロパティは、RC800コントローラー接続時にのみGGet, GSetにて使用可能です。

値

0 – Left

1 – Top

2 – Right

3 – Bottom

デフォルト:TOP

参照

[タブコントロール](#)

使用例

```
' 現在設定されているタブの配置を取得  
GGet frmMain.tabControl1.TabAlignment, alignment  
' タブの配置を0 (Left) に設定。  
GSet frmMain.tabControl1.TabAlignment, 0
```

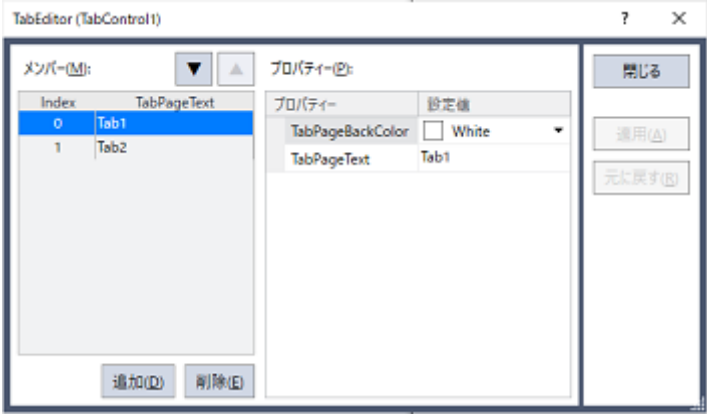
8.20.2 TabEditor プロパティ

適用

TabControl

説明

タブコントロールをセットアップするためにタブエディターを開きます。



使い方

TabEditorプロパティは、デザイン時のみ使用できます。タブページの追加や削除、タブページの背景色、タブ表示文字を設定できます。

設定	説明
追加ボタン	タブページを最後尾に追加します。
削除ボタン	選択中のタブページを削除します。
TabPageBackColor	選択中のタブページ背景色を設定します。
TabPageText	選択中のタブページのタブ表示文字を設定します。

値

- 0 – Left
- 1 – Top
- 2 – Right
- 3 – Bottom
- デフォルト:TOP

参照

[タブコントロール](#)

8.20.3 TabHeaderSize プロパティ

設計GUI上のみの機能です。

適用

TabControl

説明

タブヘッダーのサイズを設定します。(自動で調整するかどうか、幅、高さ)

使い方

GUI Builderのプロパティグリッド内で表示されるTabHeaderSizeプロパティの値を変更することで、タブヘッダーのサイズを変更できます。

プロパティ	設定値
TabEditor	Click to Edit
▼ TabHeaderSize	
TabHeaderAutoFit	True
TabHeaderWidth	100
TabHeaderHeight	20
TabIndex	4
TabTextAlign	Center
ToolTipText	

サイズの設定を変更しプロジェクトを保存した後、設定が保存されます。TabHeaderAutoFitプロパティをFalseに設定した場合、TabHeaderWidth、TabHeaderHeightプロパティを変更できます。

プロパティ	設定値
TabEditor	Click to Edit
▼ TabHeaderSize	
TabHeaderAutoFit	False
TabHeaderWidth	100
TabHeaderHeight	20
TabIndex	4
TabTextAlign	Center
ToolTipText	

参照

[タブコントロール](#)

8.20.4 TabHeaderTextAlign プロパティ

適用

TabControl

説明

タブヘッダーのテキストの配置を設定・取得します。

使い方

```
GGet Form.Control.TabControl.TabHeaderTextAlign, var  
GSet Form.Control.TabControl.TabHeaderTextAlign, value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

Control

コントロールの名前、またはコントロールの名前を含む文字列を指定します。

var

プロパティの値を取得する文字列変数を指定します。

value

プロパティに設定する文字列式を指定します。

値

1 – Left

2 – Center

3 – Right

デフォルト:Center

参照

[タブコントロール](#)

使用例

```
' 現在設定されているタブヘッダーのテキストの配置を取得  
GGet frmMain.tabControl1.TabAlignment, alignment  
' タブの配置を1 (Left) に設定。  
GSet frmMain.tabControl1.TabAlignment, 1
```

8.20.5 TabHeaderAutoFit プロパティ

適用

TabControl

説明

タブヘッダーのサイズをテキストのサイズに合わせて自動で調整するかどうかを設定・取得します。

使い方

```
GGet Form.Control.TabHeaderAutoFit, var  
GSet Form.Control.TabHeaderAutoFit, value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

Control

コントロールの名前、またはコントロールの名前を含む文字列を指定します。

var

プロパティの値を取得する文字列変数を指定します。

value

プロパティに設定する文字列式を指定します。

値

0 – False

1 – True

デフォルト:0 – False

参照

[タブコントロール](#)

使用例

```
・現在設定されているタブヘッダーの調整モードを取得  
GGet frmMain.tabControl1.TabHeaderAutoFit, autoFit  
・タブヘッダーの調整モードをFalse(手動調整)に設定。  
GSet frmMain.tabControl1.TabHeaderAutoFit, False
```

8.20.6 TabHeaderHeight プロパティ

適用

TabControl

説明

タブヘッダーの高さをピクセル値で設定・取得します。

使い方

```
GGet Form.Control.TabHeaderHeight, var  
GSet Form.Control.TabHeaderHeight, value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

Control

コントロールの名前、またはコントロールの名前を含む文字列を指定します。

var

プロパティの値を取得する文字列変数を指定します。

value

プロパティに設定する文字列式を指定します。

値

高さを表すピクセル整数値です。

参照

[タブコントロール](#)

使用例

```
' 現在設定されているタブヘッダーの高さを取得  
GGet frmMain.tabControl1.TabHeaderHeight, height  
' タブヘッダーの高さを設定。(設定した値を反映するためにはTabHeaderAutoFitをFalseに設定する  
必要があります。)  
GSet frmMain.tabControl1.TabHeaderAutoFit, False  
GSet frmMain.tabControl1.TabHeaderHeight, 48
```

8.20.7 TabHeaderWidth プロパティ

適用

TabControl

説明

タブヘッダーの幅をピクセル値で設定・取得します。

使い方

```
GGet Form.Control.TabHeaderWidth, var  
GSet Form.Control.TabHeaderWidth, value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

Control

コントロールの名前、またはコントロールの名前を含む文字列を指定します。

var

プロパティの値を取得する文字列変数を指定します。

value

プロパティに設定する文字列式を指定します。

値

幅を表すピクセル整数値です。

参照

[タブコントロール](#)

使用例

```
・現在設定されているタブヘッダーの幅を取得  
GGet frmMain.tabControl1.TabHeaderWidth, width  
・タブヘッダーの幅を設定。(設定した値を反映するためにはTabHeaderAutoFitをFalseに設定する必  
要があります。)  
GSet frmMain.tabControl1.TabHeaderAutoFit, False  
GSet frmMain.tabControl1.TabHeaderWidth, 120
```

8.20.8 TabIndex プロパティ

適用

Button, Label, TextBox, RadioButton, CheckBox, ListBox, ComboBox, PictureBox, GroupBox, VideoBox, LED, ProgressBar, TrackBar, StatusBar, Grid, TabControl, Wizard, TreeView

説明

コンテナ内のコントロールのタブオーダーを設定・取得します。

使い方

```
GGet Form.Control.TabPageBackColor(Index), var  
GSet Form.Control.TabPageBackColor(Index), value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

Control

コントロールの名前、またはコントロールの名前を含む文字列を指定します。

var

プロパティの値を取得する整数変数を指定します。

value

プロパティに設定する整数式を指定します。

値

コントロールのタブインデックス番号の整数値です。

参照

[ボタンコントロール](#), [ラベルコントロール](#), [テキストボックスコントロール](#), [ラジオボタンコントロール](#), [チェックボックスコントロール](#), [リストボックスコントロール](#), [コンボボックスコントロール](#), [ピクチャーボックスコントロール](#), [グループボックスコントロール](#), [ビデオボックスコントロール](#), [LEDコントロール](#), [プログレスバーコントロール](#), [トラックバーコントロール](#), [ステータスバーコントロール](#), [ウィザードコントロール](#), [ツリービューコントロール](#)

使用例

```
GSet frmMain.txttStatus.TabIndex, 3
```

8.20.9 tabPageBackColor プロパティ

適用

TabControl

説明

タブコントロールに表示するタブページの背景色を設定します。

使い方

```
GGet Form.Control.tabPageBackColor(index), var  
GSet Form.Control.tabPageBackColor(index), value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

Control

コントロールの名前、またはコントロールの名前を含む文字列を指定します。

Index

タブコントロールのインデックス番号を指定します。

var

プロパティの値を取得する文字列変数を指定します。

value

プロパティに設定する文字列式を指定します。

本プロパティは、RC800コントローラー接続時にのみGGet, GSetにて使用可能です。

値

タブページ背景色に指定する色の名前です。

デフォルト: Control

参照

[タブコントロール, TabEditor プロパティ](#)

使用例

```
' 現在設定されているインデックス番号2のタブページの背景色を取得  
GGet frmMain.tabControl1.tabPageBackColor(2), backColor  
' インデックス番号0のタブページの背景色を"Red"に設定。  
GSet frmMain.tabControl1.tabPageBackColor(0), "Red"
```

8.20.10 TabPageFont プロパティ

設計GUI上のみの機能です。

適用

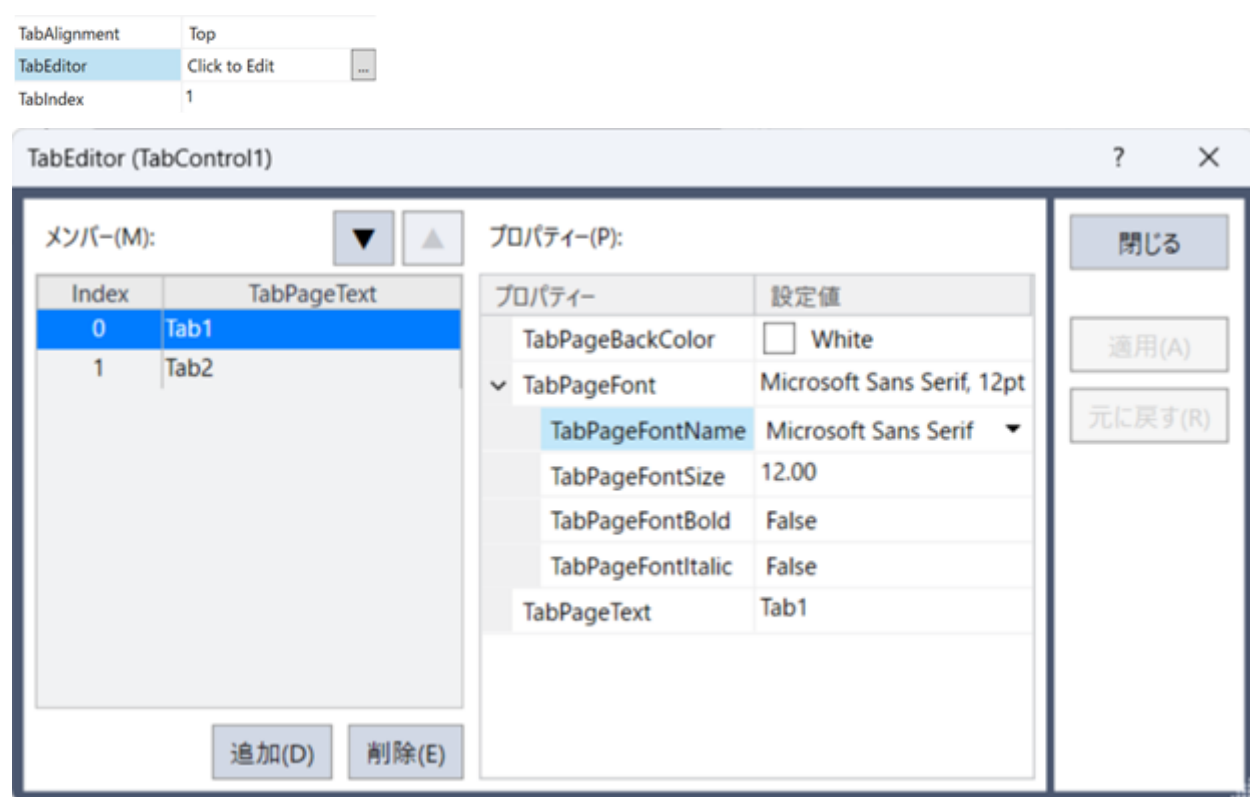
TabControl

説明

フォーム設計時に、TabControlのタブページのテキストのフォントを変更することができます。(フォント名、スタイル、サイズ)

使い方

GUI Builderのプロパティグリッド内で表示されるTabEditorのボタンを押下してタブエディターを表示します。タブエディターにてTabPageFontプロパティの値を変更することで表示文字のフォントを変更することができます。



フォントの設定を変更しプロジェクトを保存した後、設定が保存されます。

参照

[タブコントロール](#)

8.20.11 tabPageFontBold プロパティ

適用

TabControl

説明

フォーム設計時に、TabControlのタブページのテキストのフォントが“標準”または“太字”かを設定・取得します。

使い方

```
GGet Form.Control.tabPageFontBold(index), var  
GSet Form.Control.tabPageFontBold(index), value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

Control

コントロールの名前、またはコントロールの名前を含む文字列を指定します。

Index

タブコントロールのインデックス番号を指定します。

var

プロパティの値を取得する論理変数を指定します。

value

プロパティに設定する論理式を指定します。

本プロパティは、RC800コントローラー接続時にのみGGet, GSetにて使用可能です。

値

0 - False

1 - True

デフォルト: 0 - False

参照

[タブコントロール](#), [TabPageFontName プロパティ](#), [TabPageFontItalic プロパティ](#), [TabPageFontSize プロパティ](#)

使用例

・現在設定されているインデックス番号2のタブページのテキストのフォントスタイル（標準・太字）を取得

```
GGet frmMain.tabControl1.tabPageFontBold(2), fontBold
```

・インデックス番号0のタブページのテキストのフォントスタイルを太字に設定。

```
GSet frmMain.tabControl1.tabPageFontBold(0), True
```

8.20.12 tabPageFontItalic プロパティ

適用

TabControl

説明

フォーム設計時に、TabControlのタブページのテキストのフォントが“標準”または“イタリック(斜体)”かを設定・取得します。

使い方

```
GGet Form.Control.TabPageFontItalic(index), var  
GSet Form.Control.TabPageFontItalic(index), value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

Control

コントロールの名前、またはコントロールの名前を含む文字列を指定します。

Index

タブコントロールのインデックス番号を指定します。

var

プロパティの値を取得する論理変数を指定します。

value

プロパティに設定する論理式を指定します。

値

0 - False 1 - True

デフォルト: 0 - False

参照

[タブコントロール](#), [TabPageFontName プロパティ](#), [TabPageFontBold プロパティ](#),
[TabPageFontSize プロパティ](#)

使用例

・現在設定されているインデックス番号2のタブページのテキストのフォントスタイル（標準・斜体）を取得

```
GGet frmMain.tabControl1.TabPageFontItalic(2), fontItalic
```

・インデックス番号0のタブページのテキストのフォントスタイルを斜体に設定。

```
GSet frmMain.tabControl1.TabPageFontItalic(0), True
```

8.20.13 TabPageFontName プロパティ

適用

TabControl

説明

TabControlのタブページのテキストのフォントの名称を設定・取得します。

使い方

```
GGet Form.Control.TabPageFontName(index), var  
GSet Form.Control.TabPageFontName(index), value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

Control

コントロールの名前、またはコントロールの名前を含む文字列を指定します。

Index

タブコントロールのインデックス番号を指定します。

var

プロパティの値を取得する文字列変数を指定します。

value

プロパティに設定する文字列式を指定します。

値

フォントの名称の文字列です。デフォルト: Microsoft Sans Serif

参照

[タブコントロール](#), [TabPageFontSize プロパティ](#), [TabPageFontItalic プロパティ](#),
[TabPageFontBold プロパティ](#)

使用例

```
' 現在設定されているインデックス番号2のタブページのテキストのフォントの名称を取得  
GGet frmMain.tabControl1.TabPageFontName(2), fontName  
' インデックス番号0のタブページのテキストのフォントの名称を "Courier New" に設定。  
GSet frmMain.tabControl1.TabPageFontName(0), "Courier New"
```

8.20.14 TabPageFontSize プロパティ

適用

TabControl

説明

TabControlのタブページのテキストのフォントのサイズを、設定・取得します。

使い方

```
GGet Form.Control.TabPageFontSize(index), var  
GSet Form.Control.TabPageFontSize(index), value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

Control

コントロールの名前、またはコントロールの名前を含む文字列を指定します。

Index

タブコントロールのインデックス番号を指定します。

var

プロパティの値を取得する実数変数を指定します。

value

プロパティに設定する実数式を指定します。

本プロパティは、RC800コントローラー接続時にのみGGet, GSetにて使用可能です。

値

フォントのサイズのポイント値です。

デフォルト: 12

参照

[タブコントロール](#), [TabPageFontName プロパティ](#), [TabPageFontItalic プロパティ](#),
[TabPageFontBold プロパティ](#)

使用例

```
' 現在設定されているインデックス番号2のタブページのテキストのフォントのサイズを取得  
GGet frmMain.tabControl1.TabPageFontSize(2), fontSize  
' インデックス番号0のタブページのテキストのフォントのサイズを20に設定。  
GSet frmMain.tabControl1.TabPageFontSize(0), 20
```

8.20.15 TabPageIcon プロパティ

適用

TabControl

説明

タブコントロールに表示するタブページのヘッダーのアイコンのイメージを設定・取得します。

使い方

```
GGet Form.Control.TabPageIcon(index), var  
GSet Form.Control.TabPageIcon(index), value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

Control

コントロールの名前、またはコントロールの名前を含む文字列を指定します。

Index

タブコントロールのインデックス番号を指定します。

var

プロパティの値を取得する文字列変数を指定します。

value

プロパティに設定する文字列式を指定します。

値

イメージファイルのフルパスの文字列です。

デフォルト: 空白

参照

[タブコントロール](#), [TabEditor プロパティ](#)

使用例

```
' 現在設定されているインデックス番号2のタブページのヘッダーのアイコンを取得  
GGet frmMain.tabControl1.TabPageIcon(2), icon$  
' インデックス番号0のタブページの背景色を"LedGreen.ico"に設定。  
GSet frmMain.tabControl1.TabPageIcon(0), "C:\EpsonRC80\GUI\Icons\LedGreen.ico"
```

8.20.16 TabPageText プロパティ

適用

TabControl

説明

タブコントロールに表示するタブのテキストを設定します。

使い方

```
GGet Form.Control.TabPageText(Index), var  
GSet Form.Control.TabPageText(Index), value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

Control

コントロールの名前、またはコントロールの名前を含む文字列を指定します。

Index

タブコントロールのインデックス番号を指定します。

var

プロパティの値を取得する文字列変数を指定します。

value

プロパティに設定する文字列式を指定します。

本プロパティは、RC800コントローラー接続時にのみGGet, GSetにて使用可能です。

値

タブに指定するテキストです。

デフォルト: TabXX

参照

[タブコントロール, TabEditor プロパティ](#)

使用例

```
' 現在設定されているインデックス番号2のタブのテキストを取得  
GGet frmMain.tabControl1.TabPageText(2), backColor  
' インデックス番号0のタブのテキストを"モニター"に設定。  
GSet frmMain.tabControl1.TabPageText(0), "モニター"
```

8.20.17 Text プロパティ

適用

Form, Button, Label, TextBox, RadioButton, CheckBox, ComboBox, GroupBox, LED, StatusBar,TreeView

説明

コントロールに関連づけられたテキストを設定・取得します。

使い方

```
GGet Form.[Control].Text, var  
GGet Form.[Control].Text(name), var  
GSet Form.[Control].Text, value  
GSet Form.[Control].Text(name), var
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

Control

コントロールの名前、またはコントロールの名前を含む文字列を指定します。

name

テキストを設定するツリービューの項目(ノード)を示す文字列式を指定します。

var

プロパティの値を取得する文字列変数を指定します

value

プロパティに設定する文字列式を指定します。.

値

フォームやコントロールのテキストを含む文字列です。

参照

フォーム, ボタンコントロール, ラベルコントロール, テキストボックスコントロール,
ラジオボタンコントロール, チェックボックスコントロール, コンボボックスコントロール,
グループボックスコントロール, LEDコントロール, ステータスバーコントロール, TextAlign プロパティ,
ツリービューコントロール

使用例

```
GSet frmMain.lblName.Text, "Name: "
```

8.20.18 TextAlign プロパティ

適用

Button, Label, TextBox, RadioButton, CheckBox, LED

説明

コントロールのテキストの位置情報を設定・取得します。

使い方

```
GGet Form.Control.TextAlign, var  
GSet Form.Control.TextAlign, value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

Control

コントロールの名前、またはコントロールの名前を含む文字列を指定します。

var

プロパティの値を取得する整数変数を指定します。

value

プロパティに設定する整数式を指定します。

値

- ボタン、ラベル、ラジオボタン、チェックボックスコントロール
 - 1 - TopLeft
 - 2 - TopCenter
 - 3 - TopRight
 - 4 - MiddleLeft
 - 5 - MiddleCenter
 - 6 - MiddleRight
 - 7 - BottomLeft
 - 8 - BottomCenter
 - 9 - BottomRight
- デフォルト: 1 - TopLeft (Label 1)
 - 4 - MiddleLeft (RadioButton, CheckBox)
 - 5 - MiddleCenter (Button)
 - 6 - MiddleRight (LED)
- テキストボックスコントロール
 - 1 - Left
 - 2 - Center
 - 3 - Right
- デフォルト: Left

参照

[ボタンコントロール](#), [ラベルコントロール](#), [テキストボックスコントロール](#), [ラジオボタンコントロール](#), [チェックボックスコントロール](#), [LEDコントロール](#), [Text プロパティ](#)

使用例

```
GSet frmMain.lblName.TextAlign, TEXTALIGN_LEFT
```

8.20.19 Tickイベント

適用

Timer

説明

設定されたタイマー間隔に達したときに発生するイベントです。

使い方

```
Form_Control_Tick (Sender$ As String)
```

Sender\$

イベントが発生したコントロールの名前です。

参照

[タイマーコントロール](#), [Interval プロパティ](#), [Enabled プロパティ](#)

使用例

```
Function frmMain_Timer1_Tick(Sender$ As String)
    GSet frmMain.lblDateTime.Text, Date$ + " " + Time$
End
```

8.20.20 TickFrequency プロパティ

適用

TrackBar

説明

目盛りマーク間の位置の数を設定・取得します。

使い方

```
GGet Form.Control.TickFrequency, var  
GSet Form.Control.TickFrequency, value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

Control

コントロールの名前、またはコントロールの名前を含む文字列を指定します。

var

プロパティの値を取得する整数変数を指定します。

value

プロパティに設定する整数式を指定します。

値

目盛りマーク間の位置の数を表す整数値です。

参照

[トラックバーコントロール](#)

使用例

```
GSet frmMain.TrackBar1.TickFrequency, 5
```

8.20.21 TickStyle プロパティ

適用

TrackBar

説明

目盛りをトラックバーのどこに表示するかを設定・取得します。

使い方

```
GGet Form.Control.TickStyle, var  
GSet Form.Control.TickStyle, value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

Control

コントロールの名前、またはコントロールの名前を含む文字列を指定します。

var

プロパティの値を取得する整数変数を指定します。

value

プロパティに設定する整数式を指定します。

値

0 - None

1 - TopLeft

2 - BottomRight

3 - Both

デフォルト:BottomRight

参照

[トラックバーコントロール](#)

使用例

```
GSet frmMain.TrackBar1.TickStyle, TICKSTYLE_BOTH
```

8.20.22 ToolTipText プロパティ

適用

Button, Label, TextBox, RadioButton, CheckBox, ListBox, ComboBox, PictureBox, GroupBox, VideoBox, LED, ProgressBar, TrackBar, StatusBar, Grid, TabControl, Wizard, TreeView

説明

コントロールの上でマウスポインタを止めたときのポップアップに表示するテキストを設定・取得します。

使い方

```
GGet Form.Control.ToolTipText, var  
GSet Form.Control.ToolTipText, value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

Control

コントロールの名前、またはコントロールの名前を含む文字列を指定します。

var

プロパティの値を取得する文字列変数を指定します。

value

プロパティに設定する文字列式を指定します。

値

コントロールのツールチップのテキストを含む文字列です。

デフォルト: 空白

参照

[ボタンコントロール](#), [ラベルコントロール](#), [テキストボックスコントロール](#), [ラジオボタンコントロール](#), [チェックボックスコントロール](#), [リストボックスコントロール](#), [コンボボックスコントロール](#), [ピクチャーボックスコントロール](#), [グループボックスコントロール](#), [ビデオボックスコントロール](#), [LEDコントロール](#), [プログレスバーコントロール](#), [トラックバーコントロール](#), [ステータスバーコントロール](#), [タブコントロール](#), [ウィザードコントロール](#), [ツリービューコントロール](#)

使用例

```
GSet frmMain.btnStart.ToolTipText, "Click Here to Start"
```

8.20.23 Top プロパティ

適用

Form, Button, Label, TextBox, RadioButton, CheckBox, ListBox, ComboBox, PictureBox, GroupBox, VideoBox, LED, ProgressBar, TrackBar, StatusBar, Grid, TabControl, Wizard, TreeView

説明

フォームやコントロールの上側のピクセル座標を設定・取得します。

使い方

```
GGet Form.[Control].Top, var  
GSet Form.[Control].Top, value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

Control

コントロールの名前、またはコントロールの名前を含む文字列を指定します。

var

プロパティの値を取得する整数変数を指定します。

value

プロパティに設定する整数式を指定します。

値

上側の座標を表すピクセル整数値です。

参照

フォーム, ボタンコントロール, ラベルコントロール, テキストボックスコントロール, ラジオボタンコントロール, チェックボックスコントロール, リストボックスコントロール, コンボボックスコントロール, ピクチャーボックスコントロール, グループボックスコントロール, ビデオボックスコントロール, LEDコントロール, プログレスバーコントロール, トラックバーコントロール, ステータスバーコントロール, タブコントロール, ウィザードコントロール, ツリービューコントロール, Left プロパティ, Height プロパティ, Width プロパティ

使用例

```
GSet frmMain.txtStatus.Top, 200
```

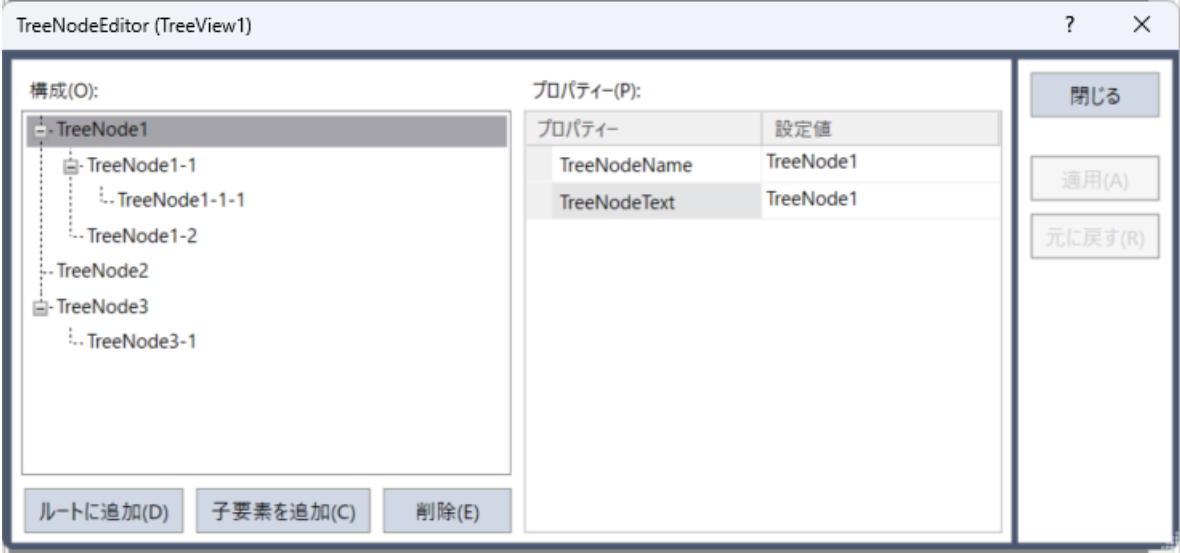
8.20.24 TreeNodeEditor プロパティ

適用

TreeView

説明

ツリービューコントロールをセットアップするためにツリーノードエディターを開きます。



使い方

TreeNodeEditorプロパティは、デザイン時のみ使用できます。ノードの追加や削除、ノードの名前、表示文字を設定できます。

設定	説明
ルートに追加ボタン	最上位の階層に新しいノードを追加します。
子要素を追加ボタン	現在選択されているノードの下に新しい子ノードを追加します。
削除ボタン	現在選択されているノードと、その子ノードをすべて削除します。
TreeNodeName	選択中のノードの名前を設定します。
TreeNodeText	選択中のノードの表示文字を設定します。

ノードを選択してドラッグ&ドロップすると、ノードの並べ替えができます。

値

上側の座標を表すピクセル整数値です。

参照

[ツリービューコントロール](#)

8.20.25 Typeプロパティ

適用

All Controls

説明

コントロールのタイプ名を取得します。

使い方

```
GGet Form.Control.Type, var
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

Control

コントロールの名前、またはコントロールの名前を含む文字列を指定します。

var

プロパティの値を取得する文字列変数を指定します。

値

コントロールのタイプ名を含む文字列。

参照

[Controls プロパティ](#)

使用例

```
GGet frmMain.Controls(index).Type, typeName$
```

8.21 U

8.21.1 Updateプロパティ

適用

TextBox

説明

テキストボックスにグローバル変数を表示するように設定されているとき、変数を書き換えられたときに自動的に表示を更新するかどうかを設定・取得します。

使い方

```
GGet Form.Control.Update, var  
GSet Form.Control.Update, value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

Control

コントロールの名前、またはコントロールの名前を含む文字列を指定します。

var

プロパティの値を取得する論理変数を指定します。

value

プロパティに設定する論理を指定します。

値

False

True

デフォルト: True

注記

SPEL+のグローバル変数(配列を除く)の値をテキストボックスに表示するには、テキストボックスのVariableプロパティに表示したいグローバル変数名を設定します。UpdateプロパティをFalseに設定した場合、設定したグローバル変数の値が更新されません。

TextBox Variableプロパティをグローバル変数に設定すると、デフォルトで、RC+の実行時にTextBox Text値を変数値に自動で更新します。Updateプロパティのデフォルト値はTrueです。ユーザーが同じ変数値を変更できるようにしたいときは、UpdateをFalseに設定する必要があります。これにより、自動更新は行われなくなり、ユーザーは新しい値を入力できるようになります。

UpdateがFalseからTrueに設定されているとき、RC+は変数値を、ユーザーがテキストボックスに入力した新しい値に設定します。メソッドの1つは、「値の変更」、または同様の名前のテキストボックスの上にチェックボックスを追加することです。チェックボックスのクリックイベントで、GGetを使用してチェックボックスのチェック状態を取得してから、GSetを使用して、TextBox Updateプロパティを設定してください。チェックボックスにチェックが入っている場合は、UpdateをFalseに設定してください。

参照

[テキストボックスコントロール](#), [Variable プロパティ](#)

使用例

```
GSet frmMain.txtStatus.Update, False
```

8.22 V

8.22.1 Value プロパティ

適用

ProgressBar, TrackBar

説明

コントロールの現在位置を表す値を設定・取得します。

使い方

```
GGet Form.Control.Value, var  
GSet Form.Control.Value, value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

Control

コントロールの名前、またはコントロールの名前を含む文字列を指定します。そのコントロールは、指定されたフォーム上に存在しなくてはなりません。

var

プロパティの値を取得する 整数変数 を指定します。

value

プロパティに設定する 整数式 を指定します。

値

現在値を表す整数値です。

参照

[プログレスバーコントロール](#), [トラックバーコントロール](#), [Maximum プロパティ](#), [Minimum プロパティ](#), [Variable プロパティ](#)

使用例

```
GSet form1.ProgressBar1.Value, 75
```

8.22.2 Variable プロパティ

適用

TextBox, ProgressBar, TrackBar

説明

テキストボックスに表示するSPEL+のグローバル変数(配列を除く)の名前を設定・取得します。

使い方

```
GGet Form.Control.Variable, var  
GSet Form.Control.Variable, value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

Control

コントロールの名前、またはコントロールの名前を含む文字列を指定します。そのコントロールは、指定されたフォーム上に存在しなくてはなりません。

var

プロパティの値を取得する文字列変数を指定します。

value

プロパティに設定する文字列式を指定します。

値

SPEL+プログラムで使うグローバル変数(配列を除く)の名前です。

デフォルト: None

注記

VariableプロパティにSPEL+のグローバル変数(配列を除く)を設定すると、自動的に変数の値をTextプロパティに設定、表示します。

TextBox Variableプロパティをグローバル変数に設定すると、デフォルトでは、RC+は実行時にTextBox Text値を変数値に自動で更新します。Updateプロパティのデフォルト値はTrueです。ユーザーが同じ変数値を変更できるようにするには、UpdateをFalseに設定する必要があります。

UpdateがFalseからTrueに設定されているとき、RC+は変数値を、ユーザーがテキストボックスに入力した新しい値に設定します。

方法の1つは、「値の変更」、または同様の名前のテキストボックスの上にチェックボックスを追加することです。チェックボックスのクリックイベントで、GGetを使用してチェックボックスのチェック状態を取得してから、GSetを使用して、TextBox Updateプロパティを設定してください。チェックボックスにチェックが入っている場合は、UpdateをFalseに設定してください。

ProgressBar Variableプロパティをグローバル変数に設定すると、プログレスバーに変数値が表示されます。TrackBar Variableプロパティをグローバル変数に設定すると、変数値はトラックバーによって設定されます。

参照

[テキストボックスコントロール](#), [プログレスバーコントロール](#), [トラックバーコントロール](#), [Value プロパティ](#)

使用例

```
GSet frmMain.txtStatus.Variable, "g_Status$"
```

8.22.3 VideoEnabled プロパティ

適用

VideoBox

説明

ビデオボックスに画像を表示するかどうかを設定、取得します。

使い方

```
GGet Form.Control.VideoEnabled, var  
GSet Form.Control.VideoEnabled, value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

Control

コントロールの名前、またはコントロールの名前を含む文字列を指定します。

var

プロパティの値を取得する論理変数を指定します。

value

プロパティに設定する論理式を指定します。

値

False

True

デフォルト: False

参照

[ビデオボックスコントロール, GraphicsEnabled プロパティ](#)

使用例

```
GSet frmMain.VideoBox1.VideoEnabled, True
```

8.22.4 Visible プロパティ

適用

Button, Label, TextBox, RadioButton, CheckBox, ListBox, ComboBox, PictureBox, GroupBox, VideoBox, LED, ProgressBar, StatusBar, TrackBar, Grid, TabControl, Wizard, TreeView

説明

コントロールを実行時に表示するかどうかを設定・取得します。

使い方

```
GGet Form.Control.Visible, var  
GSet Form.Control.Visible, value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

Control

コントロールの名前、またはコントロールの名前を含む文字列を指定します。

var

プロパティの値を取得する論理変数を指定します。

value

プロパティに設定する論理式を指定します。

値

True コントロールを表示します。

False コントロールを表示しません。

デフォルト: True

注記

GShow、あるいはGShowDialogでフォームをロードしていない状態でGGetを実行した場合、値はFalseが返ります。

参照

[ボタンコントロール](#), [ラベルコントロール](#), [テキストボックスコントロール](#), [ラジオボタンコントロール](#), [チェックボックスコントロール](#), [リストボックスコントロール](#), [コンボボックスコントロール](#), [ピクチャーボックスコントロール](#), [グループボックスコントロール](#), [ビデオボックスコントロール](#), [LEDコントロール](#), [プログレスバーコントロール](#), [ステータスバーコントロール](#), [トラックバーコントロール](#), [タブコントロール](#), [ウィザードコントロール](#), [ツリービューコントロール](#), [Enabled プロパティ](#)

使用例

```
GSet frmMain.txtStatus.Visible, True
```

8.23 W

8.23.1 Width プロパティ

適用

Form, Button, Label, TextBox, RadioButton, CheckBox, ListBox, ComboBox, PictureBox, GroupBox, VideoBox, LED, ProgressBar, TrackBar, StatusBar, Grid, TabControl, Wizard, TreeView

説明

フォームやコントロールの幅をピクセルで設定、参照します。

使い方

```
GGet Form.[Control].Width, var  
GSet Form.[Control].Width, value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

Control

コントロールの名前、またはコントロールの名前を含む文字列を指定します。

var

プロパティの値を取得する整数変数を指定します。

value

プロパティに設定する整数式を指定します。

値

フォームやコントロールの幅のピクセル整数値です。

参照

フォーム, ボタンコントロール, ラベルコントロール, テキストボックスコントロール, ラジオボタンコントロール, チェックボックスコントロール, リストボックスコントロール, コンボボックスコントロール, ピクチャーボックスコントロール, グループボックスコントロール, ビデオボックスコントロール, LEDコントロール, プログレスバーコントロール, トラックバーコントロール, ステータスバーコントロール, タブコントロール, ウィザードコントロール, ツリービューコントロール, Left プロパティ, Top プロパティ, Height プロパティ

使用例

```
GSet frmMain.txtStatus.Width, 300
```

8.23.2 WindowState プロパティ

適用

Form

説明

フォームのウィンドウの状態を設定・取得します。

使い方

```
GGet Form.WindowState, var  
GSet Form.WindowState, value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

var

プロパティの値を取得する整数変数を指定します。

value

プロパティに設定する整数式を指定します。

値

0 - Normal

1 - Minimized

2 - Maximized

デフォルト:Normal

参照

[フォーム](#)

使用例

```
GSet frmMain.WindowState, WINDOWSTATE_MAXIMIZED
```

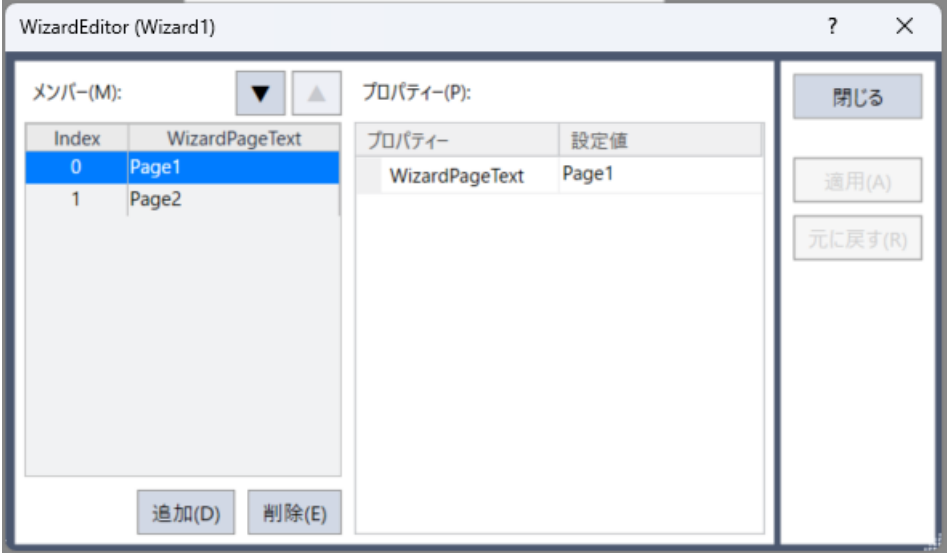
8.23.3 WizardEditor プロパティ

適用

Wizard

説明

ウィザードコントロールをセットアップするためにウィザードエディターを開きます。



使い方

WizardEditorプロパティは、デザイン時のみ使用できます。ページの追加や削除、ページの表示文字を設定できます。

設定	説明
追加ボタン	ページを最後尾に追加します。
削除ボタン	選択中のページを削除します。
WizardPageText	選択中のページのタブ表示文字を設定します。

値

True : 複数行テキストボックスのときに、その行に収まらない単語を次の行に送ります。
False : ユーザーの入力がテキストボックスの右端まできたときに、自動で水平方向へスクロールします。
デフォルト: True

参照

[ウィザードコントロール](#)

8.23.4 WordWrap プロパティ

適用

TextBox

説明

複数行のテキストボックスで、文字単位で改行するかどうかを設定、参照します。

使い方

```
GGet Form.Control.WordWrap, var  
GSet Form.Control.WordWrap, value
```

Form

フォームの名前、またはフォームの名前を含む文字列変数を指定します。

Control

コントロールの名前、またはコントロールの名前を含む文字列を指定します。そのコントロールは、指定されたフォーム上に存在しなくてはなりません。.

var

プロパティの値を取得する論理変数を指定します。

value

プロパティに設定する論理式を指定します。

値

True: 複数行テキストボックスのときに、その行に収まらない単語を次の行に送ります。

False: ユーザーの入力がテキストボックスの右端まできたときに、自動で水平方向へスクロールします。

デフォルト: True

参照

[テキストボックスコントロール](#), [MultiLine プロパティ](#), [ScrollBars プロパティ](#)

使用例

```
GSet frmMain.txtStatus.WordWrap, True
```