

# EPSON

## Epson RC+ 8.0选项 GUI Builder 8.0

翻译版

© Seiko Epson Corporation 2024

Rev. 2  
SCM24ZS6885F

# 目录

- 1. 前言 ..... 11**
  - 1.1 前言 ..... 12
  - 1.2 商标 ..... 12
  - 1.3 本手册中的商标符号 ..... 12
  - 1.4 注意 ..... 12
  - 1.5 制造商 ..... 12
  - 1.6 联系方式 ..... 12
  - 1.7 阅读本手册之前 ..... 13
    - 1.7.1 关于Epson RC+ 8.0的安装文件夹 ..... 13
- 2. 简介 ..... 14**
  - 2.1 关于GUI Builder 8.0 ..... 15
  - 2.2 功能 ..... 15
- 3. 安装 ..... 16**
  - 3.1 安装GUI Builder ..... 17
- 4. 操作入门 ..... 18**
  - 4.1 操作入门 ..... 19
  - 4.2 GUI Builder教程 ..... 19
- 5. GUI Builder环境 ..... 24**
  - 5.1 概述 ..... 25
  - 5.2 了解GUI Builder所需的基本概念 ..... 25
  - 5.3 打开GUI Builder窗口 ..... 25
  - 5.4 GUI Builder窗口的组成部分 ..... 26
    - 5.4.1 设计区域 ..... 26
    - 5.4.2 工具栏 ..... 26
    - 5.4.3 窗口资源管理器 ..... 27
    - 5.4.4 属性网格 ..... 28
    - 5.4.5 事件网格 ..... 28
    - 5.4.6 状态栏 ..... 28
  - 5.5 窗体和控件的使用 ..... 28
    - 5.5.1 创建窗体 ..... 28

- 5.5.2 删除窗体 ..... 28
- 5.5.3 打开和关闭窗口 ..... 28
- 5.5.4 放大缩小窗体显示 ..... 29
- 5.5.5 更改窗体尺寸 ..... 29
- 5.5.6 编辑多个窗体 ..... 29
- 5.5.7 创建控件 ..... 29
- 5.5.8 删除控件 ..... 29
- 5.5.9 缩放和移动控件 ..... 29
- 5.5.10 控件的复制、剪切和粘贴 ..... 30
- 5.5.11 编辑属性 ..... 30
- 5.5.12 事件处理程序的使用 ..... 30
- 5.5.13 保存工作 ..... 31
- 5.6 设置GUI Builder选项 ..... 31
- 5.7 导入其他项目的窗体 ..... 32
- 6. GUI Builder组件 ..... 33**
- 6.1 窗体 ..... 34
  - 6.1.1 描述 ..... 34
  - 6.1.2 使用 ..... 34
  - 6.1.3 窗体属性 ..... 35
  - 6.1.4 窗体事件 ..... 36
- 6.2 按钮控件 ..... 36
  - 6.2.1 描述 ..... 36
  - 6.2.2 使用 ..... 36
  - 6.2.3 按钮控件属性 ..... 36
  - 6.2.4 按钮控件事件 ..... 37
- 6.3 标签控件 ..... 38
  - 6.3.1 描述 ..... 38
  - 6.3.2 使用 ..... 38
  - 6.3.3 标签控件属性 ..... 38
  - 6.3.4 标签控件事件 ..... 39
- 6.4 文本框控件 ..... 39
  - 6.4.1 描述 ..... 40
  - 6.4.2 使用 ..... 40
  - 6.4.3 文本框控件属性 ..... 40
  - 6.4.4 文本框控件事件 ..... 42

---

6.5 单选按钮控件 .....	42
6.5.1 描述 .....	42
6.5.2 使用 .....	42
6.5.3 单选按钮控件属性 .....	43
6.5.4 单选按钮控件事件 .....	44
6.6 复选框控件 .....	44
6.6.1 描述 .....	44
6.6.2 使用 .....	44
6.6.3 复选框控件属性 .....	44
6.6.4 复选框控件事件 .....	45
6.7 列表框控件 .....	46
6.7.1 描述 .....	46
6.7.2 使用 .....	46
6.7.3 列表框控件属性 .....	46
6.7.4 列表框控件事件 .....	47
6.8 组合框控件 .....	48
6.8.1 描述 .....	48
6.8.2 使用 .....	48
6.8.3 组合框控制属性 .....	48
6.8.4 组合框控件事件 .....	49
6.9 图片框控件 .....	49
6.9.1 描述 .....	50
6.9.2 使用 .....	50
6.9.3 图片框控件属性 .....	50
6.9.4 图片框控件事件 .....	51
6.9.5 图像列表画面 .....	51
6.10 分组框控件 .....	51
6.10.1 描述 .....	51
6.10.2 使用 .....	51
6.10.3 分组框控件属性 .....	52
6.10.4 分组框控件事件 .....	53
6.11 定时器控件 .....	53
6.11.1 描述 .....	53
6.11.2 使用 .....	53
6.11.3 定时器控件属性 .....	53

---

---

6. 11. 4 定时器控件事件 .....	53
6. 12 视频框控件 .....	53
6. 12. 1 描述 .....	54
6. 12. 2 使用 .....	54
6. 12. 3 视频框控件属性 .....	54
6. 12. 4 视频框控件事件 .....	55
6. 13 LED控件 .....	55
6. 13. 1 描述 .....	55
6. 13. 2 使用 .....	55
6. 13. 3 LED控件属性 .....	55
6. 13. 4 LED控件事件 .....	57
6. 14 状态栏控件 .....	57
6. 14. 1 描述 .....	57
6. 14. 2 使用 .....	57
6. 14. 3 状态栏控件属性 .....	57
6. 14. 4 状态栏控件事件 .....	58
6. 15 进度条控件 .....	58
6. 15. 1 描述 .....	58
6. 15. 2 使用 .....	59
6. 15. 3 进度条控件属性 .....	59
6. 15. 4 进度条控件事件 .....	60
6. 16 跟踪栏控件 .....	60
6. 16. 1 描述 .....	60
6. 16. 2 使用 .....	60
6. 16. 3 跟踪栏控件属性 .....	60
6. 16. 4 跟踪栏控件事件 .....	61
6. 17 网格控件 .....	61
6. 17. 1 描述 .....	62
6. 17. 2 使用 .....	62
6. 17. 3 网格控件属性 .....	62
6. 17. 4 网格控件事件 .....	63
6. 17. 5 网格编辑器 .....	63
6. 18 选项卡控件 .....	63
6. 18. 1 描述 .....	63
6. 18. 2 使用 .....	64

---

6. 18.3 选项卡控件属性 ..... 64

**7. 操作 ..... 65**

7.1 概述 ..... 66

7.2 编程模式下的GUI开发 ..... 66

7.2.1 设计GUI ..... 66

7.2.2 调试 ..... 66

7.3 自动模式 ..... 67

7.4 处理暂停和继续 ..... 67

7.5 处理紧急停止 ..... 67

7.6 使用帮助文件 ..... 67

**8. GUI Builder参考 ..... 69**

8.1 概述 ..... 70

8.2 GUI Builder属性和事件格式描述 ..... 70

8.3 A ..... 71

8.3.1 AcceptButton属性 ..... 71

8.3.2 AddItem属性 ..... 72

8.3.3 AddRow 属性 ..... 73

8.3.4 AppendText属性 ..... 74

8.3.5 AllowStateChange属性 ..... 75

8.4 B ..... 76

8.4.1 BackColor属性 ..... 76

8.4.2 BackColorMode属性 ..... 77

8.4.3 BorderStyle属性 ..... 78

8.5 C ..... 79

8.5.1 Camera属性 ..... 79

8.5.2 CancelButton属性 ..... 80

8.5.3 CellBackColor属性 ..... 81

8.5.4 CellChanged事件 ..... 82

8.5.5 CellForeColor属性 ..... 83

8.5.6 CellText属性 ..... 84

8.5.7 Checked属性 ..... 85

8.5.8 Click事件 ..... 86

8.5.9 Closed事件 ..... 87

8.5.10 ControlBox属性 ..... 88

---

8.5.11 Controls属性 .....	89
8.5.12 Count属性 .....	90
8.6 D .....	91
8.6.1 DbIClick事件 .....	91
8.6.2 DialogResult属性 .....	92
8.6.3 DropDownStyle属性 .....	93
8.7 E .....	94
8.7.1 Enabled属性 .....	94
8.7.2 EventTaskType属性 .....	95
8.8 F .....	96
8.8.1 Font属性 .....	96
8.8.2 FontBold属性 .....	97
8.8.3 FontItalic属性 .....	98
8.8.4 FontName属性 .....	99
8.8.5 FontSize属性 .....	100
8.8.6 ForeColor属性 .....	101
8.8.7 FormBorderStyle属性 .....	102
8.8.8 FormTemplate 属性 .....	103
8.9 G .....	104
8.9.1 GClose语句 .....	104
8.9.2 GGet语句 .....	105
8.9.3 GraphicsEnabled属性 .....	106
8.9.4 GridEditor属性 .....	107
8.9.5 GSet语句 .....	109
8.9.6 GShow语句 .....	110
8.9.7 GShowDialog功能 .....	111
8.9.8 GShowDialog语句 .....	112
8.10 H .....	113
8.10.1 Height属性 .....	113
8.10.2 HelpButton属性 .....	114
8.10.3 HelpID属性 .....	115
8.11 I .....	116
8.11.1 Image属性 .....	116
8.11.2 ImageAlign属性 .....	117
8.11.3 ImageOff属性 .....	118

---

---

8. 11. 4 ImageOn属性 .....	119
8. 11. 5 ImageIndex属性 .....	120
8. 11. 6 Interval属性 .....	121
8. 11. 7 IOBit属性 .....	122
8. 11. 8 IOType属性 .....	123
8. 12 K .....	124
8. 12. 1 KeyPress事件 .....	124
8. 13 L .....	125
8. 13. 1 LargeChange 属性 .....	125
8. 13. 2 Left属性 .....	126
8. 13. 3 List属性 .....	127
8. 13. 4 ListCount属性 .....	128
8. 13. 5 Load事件 .....	129
8. 14 M .....	130
8. 14. 1 MaximizeBox属性 .....	130
8. 14. 2 Maximum属性 .....	131
8. 14. 3 MinimizeBox属性 .....	132
8. 14. 4 Minimum 属性 .....	133
8. 14. 5 MultiLine属性 .....	134
8. 15 N .....	135
8. 15. 1 Name属性 .....	135
8. 16 O .....	136
8. 16. 1 Orientation 属性 .....	136
8. 17 P .....	137
8. 17. 1 PasswordChar属性 .....	137
8. 17. 2 PressDelay 属性 .....	138
8. 17. 3 PressSound 属性 .....	139
8. 17. 4 ProgressBarStyle属性 .....	140
8. 18 R .....	141
8. 18. 1 ReadOnly属性 .....	141
8. 18. 2 RemoveRow 属性 .....	142
8. 18. 3 Resize事件 .....	143
8. 18. 4 RobotNumber属性 .....	144
8. 18. 5 RowCount 属性 .....	145

---

---

8. 19 S .....	146
8. 19. 1 Scroll事件 .....	146
8. 19. 2 ScrollBars属性 .....	147
8. 19. 3 SelectedIndex属性 .....	148
8. 19. 4 ShowDateTime 属性 .....	149
8. 19. 5 ShowEStop 属性 .....	150
8. 19. 6 ShowPrint 属性 .....	151
8. 19. 7 ShowRobot属性 .....	152
8. 19. 8 ShowSafeguard 属性 .....	153
8. 19. 9 SizeMode属性 .....	154
8. 19. 10 SmallChange属性 .....	155
8. 19. 11 Sorted属性 .....	156
8. 19. 12 StartPosition属性 .....	157
8. 20 T .....	158
8. 20. 1 TabAlignment属性 .....	158
8. 20. 2 TabEditor属性 .....	159
8. 20. 3 TabIndex属性 .....	160
8. 20. 4 TabPageBackColor属性 .....	161
8. 20. 5 TabPageFont 属性 .....	162
8. 20. 6 TabFontBold 属性 .....	163
8. 20. 7 TabPageFontItalic 属性 .....	164
8. 20. 8 TabPageFontName 属性 .....	165
8. 20. 9 TabFontSize 属性 .....	166
8. 20. 10 TabPageText 属性 .....	167
8. 20. 11 Text属性 .....	168
8. 20. 12 TextAlign属性 .....	169
8. 20. 13 Tick事件 .....	170
8. 20. 14 TickFrequency属性 .....	171
8. 20. 15 TickStyle属性 .....	172
8. 20. 16 ToolTipText属性 .....	173
8. 20. 17 Top属性 .....	174
8. 20. 18 Type属性 .....	175
8. 21 U .....	176
8. 21. 1 Update属性 .....	176

---

---

8.22 V .....	177
8.22.1 Value属性 .....	177
8.22.2 Variable属性 .....	178
8.22.3 VideoEnabled属性 .....	179
8.22.4 Visible属性 .....	180
8.23 W .....	181
8.23.1 Width属性 .....	181
8.23.2 WindowState属性 .....	182
8.23.3 WordWrap属性 .....	183

# 1. 前言

## 1.1 前言

感谢您购买本公司的机器人系统。本手册记载了正确使用机器人系统的所需事项。  
安装该机器人系统前，请仔细阅读本手册与其他相关手册。  
阅读之后请妥善保管，以便随时取阅，如有不明之处，请再次阅读。

本公司的产品均通过严格的测试和检查，以确保机器人系统的性能符合本公司的标准。但是在超出本手册所描述的环境中使用本产品，则可能会影响产品的基本性能。

本手册阐述了本公司可以预见的危险和问题。请务必遵守本手册中的安全注意事项，安全正确地使用机器人系统。

## 1.2 商标

Microsoft, Windows, Windows标识为美国Microsoft Corporation在美国及其它国家的注册商标或商标。其它品牌与产品名称均为各公司的注册商标或商标。

## 1.3 本手册中的商标符号

Microsoft® Windows® 10 operating system

Microsoft® Windows® 11 operating system

在本手册中，Windows 10、Windows 11指的是上述各操作系统。在某些情况下，Windows一般是指Windows 10和Windows 11。

## 1.4 注意

禁止擅自复印或转载本手册的部分或全部内容。  
本手册记载的内容将来可能会随时变更，恕不事先通告。  
如您发现本手册的内容有误或需要改进之处，请不吝斧正。

## 1.5 制造商

**SEIKO EPSON CORPORATION**

## 1.6 联系方式

联系方式的详细内容登载于以下手册中的“销售商”处。  
各地区的咨询处有所不同，敬请注意。  
“安全手册” - 联系方式”

从以下网站也可浏览安全手册。

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



## 1.7 阅读本手册之前

本节介绍了您在阅读本手册之前应了解的事项。

### 1.7.1 关于Epson RC+ 8.0的安装文件夹

Epson RC+ 8.0可安装在任意指定的路径。本手册中是默认Epson RC+ 8.0被安装在C:\EpsonRC80中进行说明。

## 2. 简介

## 2.1 关于GUI Builder 8.0

Epson RC+ GUI Builder 8.0选件使您可以在SPEL+ 中创建GUI (Graphical User Interface)。GUI Builder设计用于在Epson RC+ 开发环境下为创建SPEL+ 应用GUI创建简单易用的集成工具。其非常适用于需要简单的GUI且不愿使用Visual Studio等第三方产品的用户。即使用户之前从未创建过GUI，使用GUI Builder仍可轻松实现。

### 要点

对于高级的GUI应用，您可能会想要考虑同时使用Epson RC+ 8.0 RC+ API选件和Visual Studio或能够与VB Guide .NET库连接的其他工具。

## 2.2 功能

GUI Builder 8.0包支持以下功能：

- GUI完全集成于Epson RC+ 环境，以实现简单的设计、调试以及运行时的显示。无需任何第三方工具。
- 您可以在Epson RC+ 项目中创建和调试GUI窗体。
- 具有多种标准控件，包括按钮、标签、文本框等。此外，还提供有显示视频、变量状态和I/O状态的控件。
- 窗体和控件事件会作为SPEL+ 任务执行。您可以指定这些任务在Normal、NoPause或NoEmgAbort模式下运行。
- 在自动模式下，Epson RC+ 可在启动时自动显示主窗体，或者您可以通过SPEL+ 代码显示窗体。

## 3. 安装

## 3.1 安装GUI Builder

请遵守本章说明，以助于确保正确安装GUI Builder 8.0软件。

启动前，确保已关闭所有Windows应用。

1. 安装Epson RC+ 8.0。GUI Builder选件会自动安装。
2. 确保已在将使用的控制器中激活GUI Builder 8.0的软件密钥。有关激活控制器选件的方式信息，请参阅以下手册。  
“Epson RC+ 8.0用户指南”

这样便完成了GUI Builder 8.0的安装。

## 4. 操作入门

# 4.1 操作入门

本章介绍了有关GUI Builder 8.0操作入门的信息。继续前，确保已激活GUI Builder选项。有关详细信息，请参阅以下章节。

## 安装GUI Builder

如果您之前从未使用过Epson RC+，则应阅读以下手册。“Epson RC+ 8.0用户指南 - 项目和程序的创建”

下一节将展示一些简单概念的说明教程。

# 4.2 GUI Builder教程

在本节中，我们将创建运行机器人循环的简单GUI应用。我们将完成以下任务。有关GUI Builder的专用术语和说明，请参阅4.2章以后的内容。

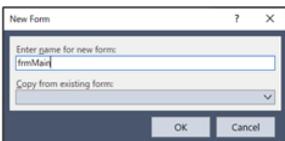
- 创建具有运行机器人循环功能的新Epson RC+ 项目。
- 创建具有[开始]和[停止]按钮的窗体，以运行机器人循环。这将展示如何创建窗体及添加按钮和事件来开始和停止 SPEL+ 任务。
- 在窗体中添加[暂停]和[继续]按钮。这将展示EventTaskType属性的使用。
- 添加设置窗体。此窗体将使用标签和文本框控件来允许用户更改机器人速度。
- 在主窗体上添加按钮，以显示设置窗体。这将展示GShowDialog语句和DialogResult属性的使用。

遵循以下步骤：

1. 创建新的Epson RC+ 8.0项目，命名为GUITest。
2. 使用机器人管理器，示教两个不同位置的两个机器人点P0和P1。
3. 在Main.prg中的函数main中添加代码，如下所示：

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

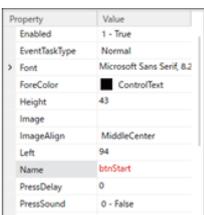
4. 从菜单栏选择[GUI Builder]，打开GUI Builder窗口。
5. 点击GUI Builder窗口工具栏上的“创建窗格”按钮，创建窗体并命名为 frmMain。然后点击[确定]按钮。



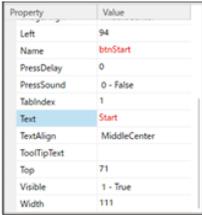
6. 点击GUI Builder窗口工具栏上的“新建按钮”按钮，然后点击窗体上的鼠标。将创建一个新按钮。

7. 设置新建按钮的属性。

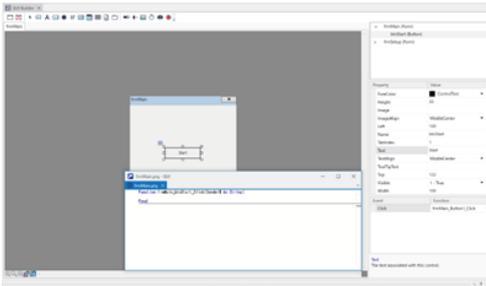
向下滚动属性网格，显示[Name]属性。将名称更改为“btnStart”。按下键盘上的[ENTER]键。



- 8. 向下滚动属性网格，显示[Text]属性。  
将“Button1”更改为“Start”。按下[ENTER]键。



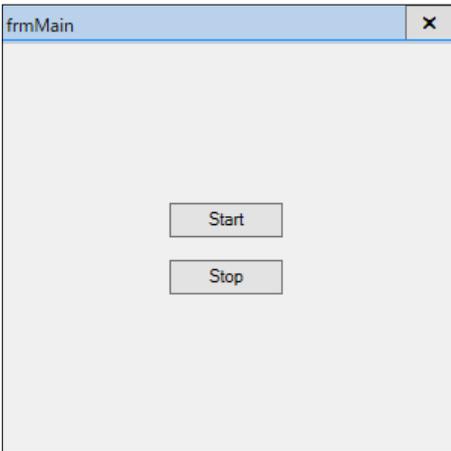
- 9. 双击窗体上的[Start]按钮。  
将打开一个名为frmMain.prg的新程序窗口，内有一个按钮点击事件处理程序的新函数。



- 10. 更改frmMain\_btnStart\_Click函数，以启动main任务，如下所示。  
这样将在用户点击[Start]按钮时启动main函数。

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

- 11. 点击GUI Builder窗口工具栏上的“新建按钮”按钮，然后点击[Start]按钮下方的窗体创建另一按钮。
- 12. 对于新按钮，将Name属性改为btnStop，Text属性改为Stop。  
此时，窗体应为如下所示：



- 13. 双击[Stop]按钮以创建按钮点击事件处理程序。  
然后，按如下所示更改代码：

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

- 14. 按F5构建项目并显示运行窗口。  
如果发生任何构建错误，则修改代码并再按F5。
- 15. 选择运行窗口上的[型号]单选按钮。

16. 点击运行窗口上的[开始]按钮。
17. 将显示frmMain。点击frmMain上的[Start ]按钮。  
此时，机器人应在P0和P1之间移动。
18. 点击窗体上的[Stop]按钮。  
机器人任务将停止。
19. 此时，点击窗体右上角的X按钮。  
窗体将关闭。  
现在我们将在GUI上添加[Pause ]和[Continue ]按钮。
20. 点击GUI Builder窗口工具栏上的[创建按钮]按钮。  
点击[Start]按钮右侧的窗体创建新按钮。
21. 对于新按钮，将Name属性改为btnPause，Text属性改为Pause。
22. 将[Pause]按钮的EventTaskType改为1 - NoPause。  
这样便允许按钮点击事件处理程序执行Pause语句，而无需暂停任务本身。
23. 双击[Pause ]按钮创建事件处理程序函数。  
然后添加Pause语句更改代码，如下所示：

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

24. 点击GUI Builder窗口工具栏上的[创建按钮]按钮。  
点击[Stop]按钮右侧的窗体创建新按钮。
25. 对于新按钮，将Name属性改为“btnCont”，Text 属性改为“Continue”。
26. 将[Continue ]按钮的EventTaskType改为1 - NoPause。  
这样便允许按钮点击事件处理程序在正常任务暂停时执行Cont语句。
27. 双击[Continue ]按钮创建事件处理程序函数。  
然后添加Cont语句更改代码，如下所示：

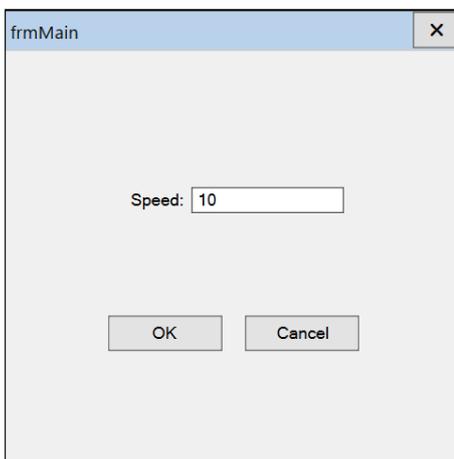
```
Function frmMain_btnCont_Click(Sender$ As String)
    Cont
End
```

28. 按F5构建项目并显示运行窗口。  
如果发生任何构建错误，则修改代码并再按F5。
29. 点击运行窗口上的[开始]按钮。将显示窗体。
30. 点击窗体上的[Star]按钮。  
机器人循环将执行。
31. 点击窗体上的[ Pause]按钮。  
机器人循环将暂停。
32. 点击窗体上的[Continue ]按钮。  
机器人循环将继续。

33. 点击[Stop]按钮。  
然后点击窗体右上角的[关闭]按钮关闭窗口。  
现在我们将在GUI上添加设置窗体。
34. 点击GUI Builder窗口工具栏上的[创建窗格]按钮，创建窗体并命名为 frmSetup。  
按下[确定]按钮，则在新的选项卡页面上显示新建窗格。
35. 点击[创建标签]按钮。  
点击[frmSetup]窗体，创建新标签。
36. 在属性网格中，将新标签名称改为“lblSpeed”。
37. 将Text属性改为“Speed:”。
38. 点击[创建文本框]按钮。  
点击[frmSetup]窗体的[Speed]标签右侧，创建新文本框。
39. 将Name属性改为“txtSpeed”，Text属性改为“10”。
40. 双击[frmSetup]窗体，为窗体创建Load事件处理程序函数。  
如下添加程序。

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

41. 点击GUI Builder窗口工具栏上的“新建按钮”按钮，然后点击窗体以添加按钮。  
将此按钮命名为“btnOK”，并将Text属性设为“OK”。
42. 点击GUI Builder窗口工具栏上的“新建按钮”按钮，然后点击窗体以添加按钮。  
将此按钮命名为“btnCancel”，并将Text属性设为“Cancel”。此时，设置窗体应类似于如下所示窗体：



43. 双击[OK]按钮创建事件处理程序函数。然后更改代码以设置窗体的DialogResult属性并调用GClose，如下所示：

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

44. 双击[取消]按钮创建事件处理程序函数。然后更改代码以设置窗体的DialogResult属性并调用GClose，如下所示：

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

45. 点击GUI Builder窗口的frmMain选项卡，重新与frmMain一起工作。
46. 点击GUI Builder窗口工具栏上的“新建按钮”按钮，然后点击窗体以添加按钮。  
将此按钮命名为“btnSetup”，并将Text属性设为“Setup”。
47. 双击[设置]按钮创建事件处理程序函数。然后更改代码以显示设置对话框并设置新机器人速度，如下所示：

```
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. 按F5构建项目并打开运行窗口。
49. 点击运行窗口上的[开始]按钮。将显示主窗体。
50. 点击[frmMain]主窗体上的[Start]按钮。  
机器人开始执行循环动作。
51. 点击主窗体上的[设置]按钮。将显示设置对话框以及文本框内的当前机器人速度。
52. 输入新速度并点击[确定]。
53. 机器人的循环动作会被变更为设置好的速度。  
如果点击了[Stop]按钮且机器人停止，则动作速度将重置为默认速度。  
至此便完成了教程。

## 5. GUI Builder环境

## 5.1 概述

在本章中，我们将着重介绍一些概念和定义，以完全了解GUI Builder及其组件。我们将涉及以下主题：

- 使用GUI Builder应了解的基本概念
- 如何打开GUI Builder窗口
- GUI Builder窗口的组成部分
- 如何使用窗体和控件
- 设置GUI Builder选项

## 5.2 了解GUI Builder所需的基本概念

一些基本概念的简单说明将有助于您更好地理解本章内容。请在继续阅读本章其他内容之前回顾以下概念。

### ■ 什么是GUI？

GUI即图形用户界面（Graphical User Interface）。GUI可使您的操作员轻松与SPEL+ 应用互动，以运行循环或执行设置功能。GUI的基本构件即窗体。

### ■ 什么是窗体？

窗体即含有控件的窗口或对话框。它是GUI 应用的基本组成单元。当运行期间显示窗体时，窗体上的控件为有效状态并准备接收用户的键盘和鼠标事件。GUI项目可具有一个窗体或多个窗体。

### ■ 什么是控件？

控件即窗体中包含的对象，如按钮、复选框、文本框等。每种控件均具有一套各自的属性和事件。

### ■ 什么是事件？

事件即创建的SPEL+ 功能，当窗体或控件事件发生时，GUI会调用此功能。例如，当用户点击按钮控件时，按钮点击事件可以在发生点击时调用指定运行的SPEL+ 功能。

## 5.3 打开GUI Builder窗口

GUI Builder窗口可在Epson RC+ 开发环境中打开。

Epson RC+ 启动后，可通过两种方式打开GUI Builder窗口：

### ■ 通过主工具栏：

在主工具栏中有GUI Builder图标 。点击GUI Builder图标将打开GUI Builder窗口。

### ■ 通过工具菜单：

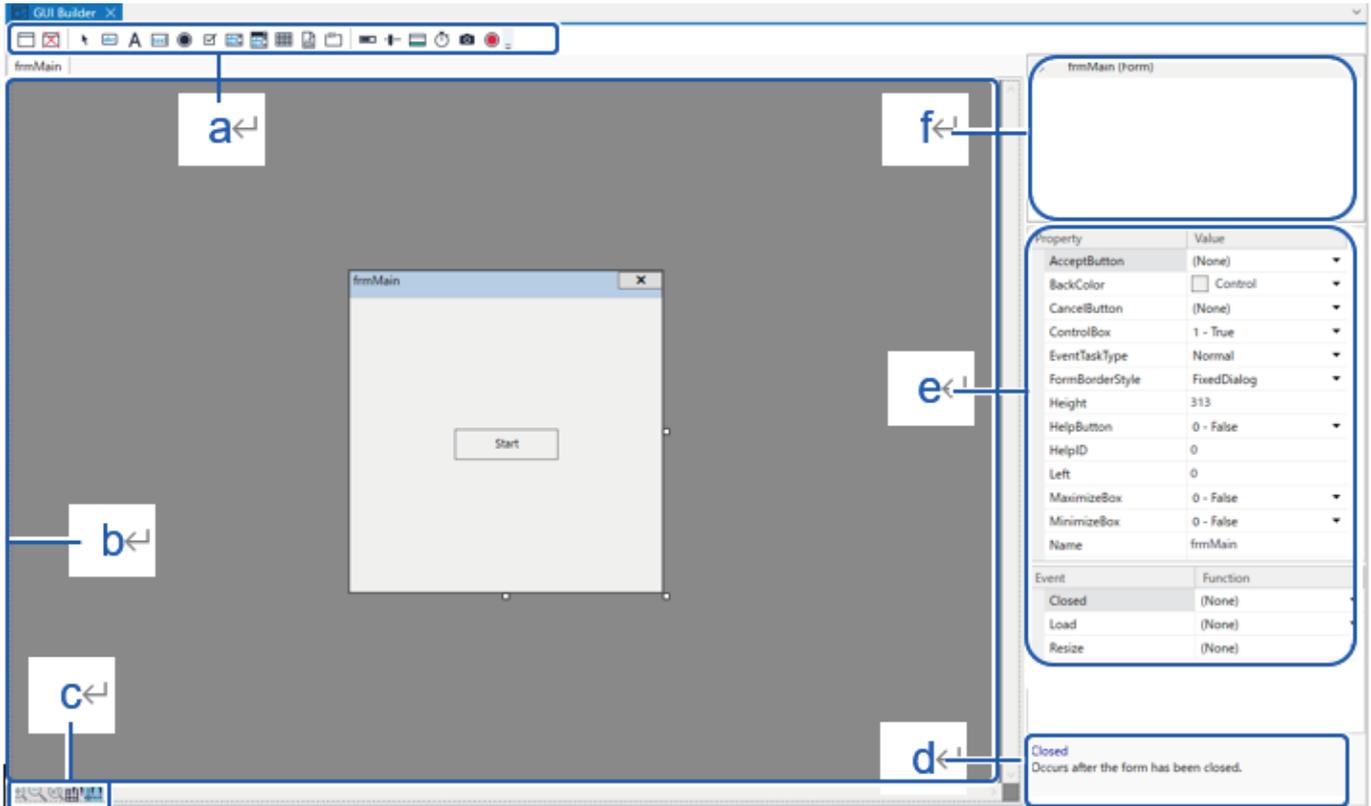
通过在工具菜单中选择GUI Builder将打开GUI Builder窗口。

一旦GUI Builder窗口打开，您便可以开始使用GUI Builder设计您的GUI应用。

接下来几页将介绍GUI Builder窗口的组成部分。

# 5.4 GUI Builder窗口的组成部分

GUI Builder窗口如下所示。后述章节将分别介绍各所示部分。



符号	项目
a	工具栏
b	设计区域
c	状态栏
d	事件网格
e	属性网格
f	窗口资源管理器

## 5.4.1 设计区域

这里是设计时显示窗体的位置。

每个打开的窗体会显示在各自的选项卡上。您可以通过点击相关选项卡或双击窗体资源管理器上的窗体在窗体之间轻松切换。

如果窗体大于设计区域，则会显示滚动条，您可以使用滚动条滚动访问窗体的所有区域。

## 5.4.2 工具栏

工具栏包括创建窗体和控件的按钮。

GUI Builder工具栏位于GUI Builder窗口顶部的标题栏正下方，如下所示：



下文所示为每个GUI Builder工具栏按钮的一般描述。

按钮	名称	描述
	新建窗体	创建新窗体。将弹出一个对话框并要求用户输入新窗体的名称。
	删除窗体	删除当前项目中的窗体。如果当前项目中没有窗体，此按钮将会变暗。
	指针	点击此按钮可中止新控件的添加。
	新建按钮	创建新的按钮控件。
	新建标签	创建用于显示静态文本的新标签控件。
	新建文本框	创建允许操作员输入文本的新文本框控件。
	新建单选按钮	创建新的单选按钮控件。
	新建复选框	创建新的复选框控件。
	新建列表框	创建新的列表框控件。
	新建组合框	创建新的组合框控件。
	新建网格框	创建新的网格控件。
	新建图片框	创建新图片框。
	新建分组框	创建新的分组框控件。
	新建状态栏	新建一个新的状态栏控件。
	新建进度条	新建一个新的进度栏控件。
	新建跟踪栏	新建一个新的跟踪栏控件。
	新建定时器	创建新的定时器控件。
	新建视频框	创建新的视频框控件。此控件可使您使用Vision Guide选项显示视频。
	新建LED	创建新LED控件。此控件可使您显示I/O状态。您可以视情况允许操作员双击控件更改输出状态。此控件可采用内置图片和各种颜色，或者您可以使用个人图像显示关闭和打开状态。

### 5.4.3 窗口资源管理器

窗体资源管理器是包括当前项目各窗体及其相关控件的树。每当创建新窗体或控件时，便会添加到此树。

点击窗体便会在设计区域打开位于各选项卡的窗体。同时显示此窗体的属性和事件。右击窗体打开菜单，设置启动窗体、删除窗体或将其关闭。

点击树上的控件会激活其相关的窗体并将当前设计集中到此控件。之后会显示此控件的属性和事件。

## 5.4.4 属性网格

属性网格用于显示和编辑窗体和控件属性。

当您选择一个窗体或控件时，网格中会显示相关属性。共有两栏：Property和Value。

Property: 属性名称。

Value: 当前值（可编辑）。

## 5.4.5 事件网格

事件网格用于显示和更改相关窗体或控件的事件。每个事件均具有一个在发生事件时会调用的用户函数。

## 5.4.6 状态栏

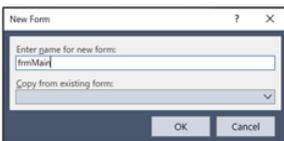
状态栏用于设置窗体显示的放大缩小、布局模式。

# 5.5 窗体和控件的使用

## 5.5.1 创建窗体

要创建窗体：

1. 打开GUI Builder窗口。
2. 点击工具栏上的“创建窗格”按钮。



3. 输入窗体名称。可以使用以下名称。

- 以数字或下划线开头的窗体名称
- 与当前正在编辑项目相同的名称

出现“创建窗格”对话框，您可以从“从现有窗格中复制”选择其他窗体随意复制。

## 5.5.2 删除窗体

若要删除窗体，右击窗体资源管理器中的窗体，然后选择[删除]。此时将显示确认消息。点击[是]删除窗体。此外，还可使用以下方法。

- 点击工具栏上的[删除窗体]按钮。
- 右击窗体选项卡选择-[删除]菜单
- 右击项目资源管理器上的对象窗体，然后选择[删除]菜单

## 5.5.3 打开和关闭窗口

可通过以下两种方式打开窗体。

- 从项目资源管理器中双击窗体。
- 从窗体资源管理器中单击窗体。

若要关闭窗口，右击窗体资源管理器，然后选择[关闭]菜单。或者，右击窗体选项卡，选择[关闭]菜单。

## 5.5.4 放大缩小窗体显示

放大缩小窗体的显示大小，有以下两种方法。

- 在GUI Builder窗口的设计区域单击任意位置后，按住[Ctrl]键操作鼠标滚轮。
- 单击GUI Builder窗口状态栏上的以下按钮进行操作。

按钮显示	说明
+	放大窗体显示。
-	缩小窗体显示。
×1	将窗体显示设为标准倍率。

## 5.5.5 更改窗体尺寸

可采用两种方式更改窗体尺寸：

- 将光标移到GUI Builder窗口设计区域窗体处显示窗体的边缘，使光标变为双箭头，然后拖拽调整大小。
- 更改属性网格的Height(高度)、Width(宽度)属性的值，或从SPEL+程序使用GSet语句改写数值。

## 5.5.6 编辑多个窗体

GUI Builder窗口允许通过选项卡界面使用多个窗体。

您可以通过点击相应的选项卡在当前打开的窗体之间切换。



您可以在一个窗体中复制控件并将其粘贴到另一窗体。

## 5.5.7 创建控件

要创建控件

1. 打开想要添加控件的窗体。
2. 点击GUI Builder窗口工具栏上的控件按钮。  
将显示十字光标。
3. 您可以在想要的控件位置旁单击窗体，将会以默认尺寸创建此控件。  
或者您可以用鼠标在窗体上拖拽控件尺寸的轮廓，然后释放鼠标以轮廓尺寸创建控件。

## 5.5.8 删除控件

要删除控件

1. 打开想要删除一个或多个控件的窗体。
2. 点击您想删除的控件。  
若要选择更多控件，按住Ctrl或Shift键并点击各附加控件。
3. 按下键盘上的Del键。将删除所选控件。

## 5.5.9 缩放和移动控件

要更改控件尺寸

- 使用鼠标拖拽GUI Builder窗口设计区域内一个控件的尺寸句柄。  
注意  
在选择多个控件的状态下无法操作。
- 更改属性网格的Height(高度)、Width(宽度)属性的值, 或从SPEL+程序使用GSet语句改写数值。
- 按住[Ctrl]键, 同时点击要更改尺寸方向的箭头[↑/↓/←/→]。要更改较多尺寸时, 则按住[Ctrl]键及[Shift]键, 同时点击箭头[↑/↓/←/→]。

要移动控件

- 点击控件将其激活, 然后释放鼠标。鼠标位于控件上方时, 将显示移动光标 。点击并拖拽控件至新位置。
- 设计时更改属性网格中的Left和Top属性, 或者在运行时使用GSet更改Left和Top属性。

## 5.5.10 控件的复制、剪切和粘贴

选择复制或剪切的控件

首先, 点击一个控件。若要选择更多控件, 按住Ctrl或Shift键并点击各附加控件。

复制选定控件共有三种创建事件处理程序的方式:

- 按下Ctrl+C
- 点击主工具栏上的复制按钮
- 从编辑菜单中选择复制

剪切选定控件共有三种创建事件处理程序的方式:

- 按下Ctrl+X
- 点击主工具栏上的剪切按钮
- 从编辑菜单中选择剪切

粘贴控件共有三种创建事件处理程序的方式:

- 按下Ctrl+V
- 点击主工具栏上的粘贴按钮
- 从编辑菜单中选择粘贴

## 5.5.11 编辑属性

若要编辑属性, 首先点击窗体或控件, 以在属性网格中显示相关属性。

- 对于需要文本输入的属性  
点击属性在属性网格中更改。然后输入新值并按下Enter, 或选择其他属性行应用更改。
- 对于具有下拉列表值的属性  
点击属性在属性网格中更改。点击值右侧的下箭头按钮。从下拉列表中选择新值。选择之后便会应用更改。
- 对于具有数值选择按钮的属性

点击属性在属性网格中更改。点击值右侧的浏览  按钮。从对话框中选择新值。

## 5.5.12 事件处理程序的使用

事件处理程序为具有指定事件的正确参数的SPEL+ 函数。

事件处理程序函数可位于当前项目中的任一SPEL+ 程序文件内。

默认情况下, 将在为窗体创建的程序文件内创建事件处理程序。例如, 在为按钮控件点击事件创建事件处理程序时, 如果没有窗体程序文件, 则会首先创建文件, 然后再将函数添加至此文件。

要创建事件处理程序共有三种创建事件处理程序的方式：

- 双击窗体或控件创建默认事件处理程序。例如，如果双击按钮控件，则会创建点击事件处理程序函数。
- 在事件网格中双击事件名称。
- 在事件网格中选择事件值下拉列表的项目中已存在的函数。列表中仅显示具有正确参数的函数。

要更改事件处理程序

您可以通过在事件值下拉列表中选择另一函数来更改事件处理程序函数。



要禁用事件处理程序

有时在创建事件处理程序之后，您可能会决定不再需要此程序。

若要禁用事件处理程序，在函数值下拉列表中选择 (None)。这并不会删除相关函数。但在事件发生时不会再调用相关函数。



使用EventTaskType

事件发生时，事件处理程序函数会作为SPEL+ 任务启动。

EventTaskType属性可使您指定将执行的任务类型。这对于需要在暂停情况或紧急停止情况下执行的事件非常重要。

例如，如果按钮点击事件处理程序将执行Pause或Cont，则EventTaskType必须设为1 - NoPause。或者如果事件处理程序在紧急停止情况下必须执行Reset，则 EventTaskType必须设为2 - NoEmgAbort。

### 5.5.13 保存工作

在GUI Builder窗口中进行更改之后，您可以通过三种方式保存工作：

- 按下Ctrl+S键。
- 从文件菜单中选择保存。
- 点击Epson RC+ 主工具栏上的项目保存按钮。

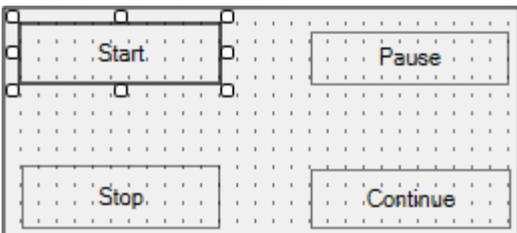
## 5.6 设置GUI Builder选项

单击GUI Builder窗口状态栏上的[与网格连在一起]、[与行连在一起]进行操作。可以通过GUI Builder的设置更改GUI Builder窗口中窗体上的控件的布局方法。

单击状态栏中的[与网格连在一起]。

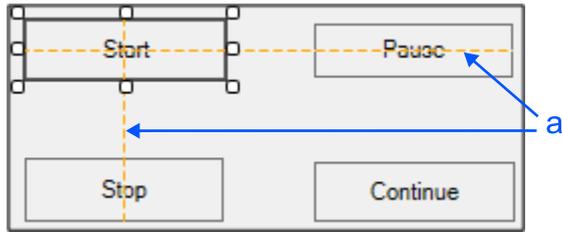


单击[与网格连在一起]时，GUI Builder显示网格，将控件布局到网格的位置。



更改网格的大小(间隔)时，应右击[与网格连在一起]，更改[网格宽度]的值。

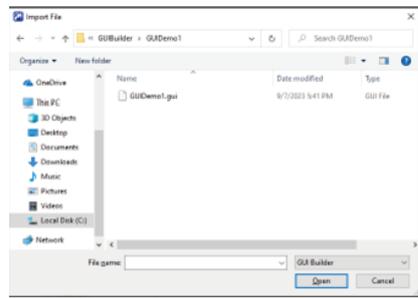
单击[与行连在一起]时，GUI Builder显示行(a)，将控件添加布局到行。



## 5.7 导入其他项目的窗体

您可以使用文件|导入菜单导入其他项目的GUI Builder窗体。

1. 选择[文件]-[导入]，或者从项目浏览器的GUI创建器中，在[窗口]上单击鼠标右键然后选择[添加窗体]-[当前存在的窗体]。



2. 导航到所需项目文件夹导入并选择GUI Builder文件类型。
3. 选择GUI文件名称并点击[打开]。
4. 将出现一个具有导入项目中窗体列表的对话框。



5. 选择您想要导入的一个或多个窗体，然后点击[确定]。
6. 选定窗体将添加至当前项目。如果窗体已存在，将提示是否覆盖现有窗体。

### 要点

使用按其它项目定义的事件处理程序时，需要导入窗体程序[form name .prg]。

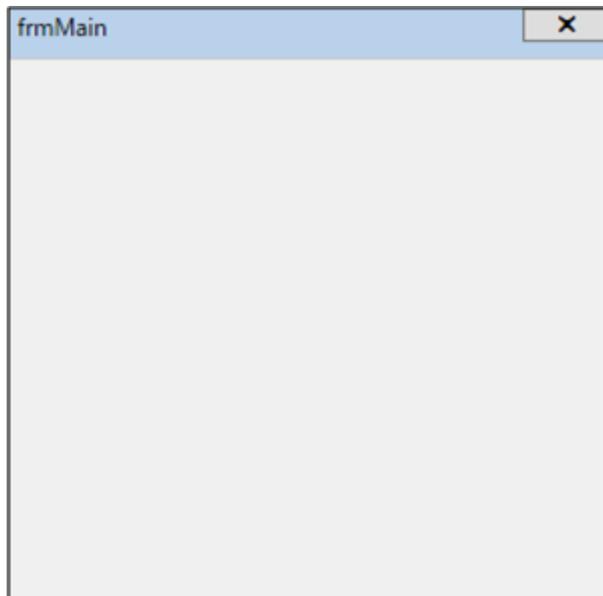
## 6. GUI Builder组件

本章将介绍窗体以及窗体上可用的所有控件。每个组件章节将提供与组件相关的所有属性和事件的使用信息。有关属性、事件和语句的更多详细信息，请参阅GUI Builder参考章节。

## 6.1 窗体

### 6.1.1 描述

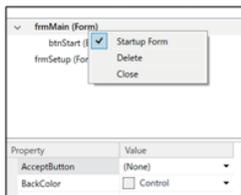
窗体是GUI应用的基本构件。它可使您显示具有控件的窗口或对话框，以实现用户交互。GUI应用可具有一个窗体或多个窗体。



### 6.1.2 使用

显示窗体窗体可通过四种方式显示：

1. 如果窗体设为启动窗体，则在自动模式下，如果系统启动，将自动显示窗体。若要设置启动窗体，在窗体资源管理器中右击窗体，然后选择“启动窗格”，如下所示：



窗体将以黑体显示，表示其为启动窗体。您还可以在项目 | 属性 | GUI Builder 中为项目设置启动窗体。

2. 使用GShowDialog函数。这样会将窗体显示为对话框并返回DialogResult值。

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

通常情况下，DialogResult值采用窗体上的按钮设置，如[确定]按钮和[取消]按钮。

3. 使用GShow语句。这样会将窗体显示为窗口。

```
GShow frmIODiags
```

4. 在运行窗口中从下拉列表中选择希望执行的窗体，单击[开始]按钮，显示选择的窗体。

设置窗体外观

设置FormBorderStyle属性。对于用作对话框的窗体，应使用3 - FixedDialog。

设置ControlBox、MaximizeBox、MinimizeBox，以配置窗体的标题栏。

设置WindowState，以便在显示窗体时以Normal size、Maximized或Minimized形式显示。

## 使用帮助

您可以在帮助文件中将HelpButton属性设为True并将HelpID属性设为主题ID，进而显示帮助文件中的帮助主题。有关详细信息，请参阅以下章节。

[使用帮助文件](#)

### 6.1.3 窗体属性

属性	描述
AcceptButton	设置用户键入Enter时执行点击事件的按钮控件。 默认: None
BackColor	设置窗体的背景颜色。 默认: Control
CancelButton	设置用户键入Esc时执行点击事件的按钮控件。执行后，窗体关闭。 默认: None
ControlBox	设置是否显示标题栏控件框项目。 默认: True
Controls	窗体上的控件数组。
Count	获取控件数组中的控件数量。
Dialog Result	设置窗体关闭时的返回值（仅运行时）。
EventTaskType	设置用于事件的任务类型。 默认: Normal
FormBorderStyle	设置用于窗体的边框类型。 默认: FixedDialog
FormTemplate	从模板中选择并设置窗体的尺寸。
Height	设置控件高度，单位为像素。
HelpButton	设置是否显示标题栏帮助按钮。 默认: False
HelpID	设置用户点击标题栏帮助按钮时显示的帮助主题ID。 默认: 0
Left	设置窗体的左坐标，单位为像素。
MaximizeBox	设置是否显示标题栏最大化按钮。 默认: False
MinimizeBox	设置是否显示标题栏最小化按钮。 默认: False
Name	设置控件名称。
StartPosition	设置窗体的启动位置。 默认: CenterScreen
Text	用于设置控件文本。 默认: 窗体名称
Top	获取或设置窗体的顶坐标，单位为像素。

属性	描述
Type	获取控件的类型名称。
Width	设置窗体宽度，单位为像素。
WindowState	设置窗体的默认窗口状态。 默认：Normal

### 6.1.4 窗体事件

事件	描述
Closed	窗体关闭后执行。
Load	窗体加载时执行。
Resize	重新调整窗体尺寸时执行。

## 6.2 按钮控件

### 6.2.1 描述

按钮控件可通过用户点击按钮来执行动作。按钮控件可显示文本和图像。点击按钮时，看上去就像按下按钮，然后释放。



### 6.2.2 使用

按钮控件可使操作员通过点击鼠标执行某些动作。

您可以使用多个属性改变按钮外观，如ForeColor（用于按钮文本）、BackColor、Font、TextAlign、Image和ImageAlign。您必须提供点击事件函数来确定是否在操作员点击按钮后会在事件处理程序中执行某些动作。

### 6.2.3 按钮控件属性

属性	描述
BackColor	控件的背景颜色。 默认：Control
BackColorMode	控件的背景颜色模式。 默认：Visual Style
Enabled	设置是否在运行时启用控件。 默认：True
EventTaskType	用于事件的任务类型。 默认：0 - Normal
Font	设置控件文本的字体。（字体名称、样式、大小） 默认：Microsoft Sans Serif 8.25 pt
FontBold	将当前字体设为粗体。 默认：False

属性	描述
FontItalic	将当前字体设为斜体。 默认: False
FontName	设置当前字体的名称。 默认: Microsoft Sans Serif
FontSize	以点设置当前字体的大小。 默认: 8.25
ForeColor	控件文本的前景颜色。 默认: ControlText
Height	控件高度, 单位为像素。
Image	控件图片。若要删除, 可按下[删除]键。 默认: 空
ImageAlign	控件的图片对齐。 默认: MiddleCenter
Left	控件的左坐标, 单位为像素。
Name	控件名称。 默认: ButtonXX
PressDelay	设置单击控件时, 执行时间之前的时间(以毫秒msec为单位)。如果在设定时间过去之前释放鼠标单击, 则该事件将不会执行。 默认: 立即执行事件
PressSound	设置点击控件时是否发出蜂鸣声。 默认: False - 不发出蜂鸣声
TabIndex	控件的选项卡索引。
Text	控件文本。 默认: 控件名称
TextAlign	选择文本对齐。 默认: MiddleCenter
ToolTipText	控件提示信息中使用的文本。 默认: 空
Top	控件的顶坐标, 单位为像素。
Visible	控件在运行时是否可见。 默认: True
Width	控件宽度, 单位为像素。

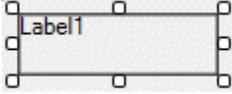
## 6.2.4 按钮控件事件

事件	描述
Click	用户使用鼠标单击控件时执行。

## 6.3 标签控件

### 6.3.1 描述

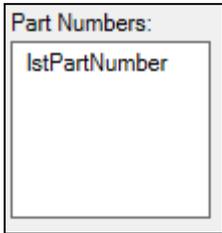
标签控件用于显示用户无法编辑的文本或图像。还用于识别窗体上的对象，以描述点击某个控件会执行的动作，例如，根据运行时间事件或应用中的过程显示信息。



### 6.3.2 使用

使用标签控件可在窗体上显示静态文本。

通常情况下，标签控件用于为另一控件提供文本标签。例如，如果窗体上有一个ListBox控件，您可以将标签控件放在具有列表名称的列表框上方。在下例中，具有Text属性的标签设为“Part Numbers:”，用以标识列表框。



#### 设置标签外观

要更改标签的外观，使用以下属性。

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

#### 使用热键（助记符）

您可以将标签Text属性中的一个字符指定为热键。

在此标签的Text属性中，在热键前使用字符（&）。

例如，在上例中标签的Text属性为“Part &Numbers:”。

运行时，如果用户键入Alt+N，则焦点将进入列表框。

使用热键时，务必设置选项卡顺序，使标签的选项卡索引位于标识的控件选项卡索引之前。

### 6.3.3 标签控件属性

属性	描述
BackColor	控件的背景颜色。 默认: Control
BorderStyle	用于控件的边框。 默认: None
Enabled	设置是否在运行时启用控件。 默认: True
EventTaskType	用于事件的任务类型。 默认: Normal
Font	设置控件文本的字体（字体名称、样式、大小）。 默认: Microsoft Sans Serif 8.25 pt
FontBold	将当前字体显示为粗体。 默认: False

属性	描述
FontItalic	将当前字体显示为斜体。 默认: False
FontName	设置当前字体的名称。 默认: Microsoft Sans Serif
FontSize	以点设置当前字体的大小。 默认: 8.25
ForeColor	控件文本的前景颜色。 默认: ControlText
Height	控件高度, 单位为像素。
Image	控件图片。若要删除, 可按下[删除]键。 默认: 空
ImageAlign	控件的图片对齐。 默认: MiddleCenter
Left	控件的左坐标, 单位为像素。
Name	控件名称。 默认: LabelXX
TabIndex	控件的选项卡索引。
Text	控件文本。 默认: 控件名称
TextAlign	选择文本对齐。 默认: TopLeft
ToolTipText	控件提示信息中使用的文本。 默认: 空
Top	控件的顶坐标, 单位为像素。
Visible	控件在运行时是否可见。 默认: True
Width	控件宽度, 单位为像素。

### 6.3.4 标签控件事件

事件	描述
Click	用户使用鼠标单击控件时执行。
Db1Click	用户使用鼠标双击控件时执行。

## 6.4 文本框控件

## 6.4.1 描述

文本框用于获取操作员的输入或显示文本。文本框控件通常用于可编辑文本，尽管也可设为只读。文本框可显示多行，并根据控件尺寸进行文本换行。文本框控件允许在控件中显示或输入单一格式的文本。



## 6.4.2 使用

文本框可用于单行模式或多行模式。

### 单行文本框

默认情况下，创建文本框时，文本框为单行模式。您仅可更改控件宽度。用户仅可键入一行字符。

### 多行文本框

若要使用多行模式，将Multiline属性设为True。在此模式下，您可以更改控件的高度和宽度。通过设置Scrollbars属性还可以显示滚动条。若要显示水平滚动条，需将WordWrap属性设为False。

### 设置文本框外观

要更改文本框的外观，使用以下属性。

BorderStyle、ForeColor、BackColor、Font、TextAlign

### 显示SPEL+ 全局变量(数组除外) 状态

您可以通过设置Variable属性自动显示SPEL+ 全局变量的值。

### 注意

如果SPEL+的全局变量未先显示在Variable属性中时，请重新创建项目。

## 6.4.3 文本框控件属性

属性	描述
AppendText	运行时附加文本。
BackColor	控件的背景颜色。 默认: Window
BorderStyle	用于控件的边框。 默认: Fixed3D
Enabled	设置是否在运行时启用控件。 默认: True
EventTaskType	用于事件的任务类型。 默认: Normal
Font	设置控件文本的字体。 默认: Microsoft Sans Serif 8.25 pt
FontBold	将当前字体设为粗体。 默认: False
FontItalic	将当前字体显示为斜体。 默认: False

属性	描述
FontName	设置当前字体的名称。 默认: Microsoft Sans Serif
FontSize	以点设置当前字体的大小。 默认: 8.25
ForeColor	控件文本的前景颜色。 默认: WindowText
Height	控件高度, 单位为像素。
Left	控件的左坐标, 单位为像素。
Multiline	设置是否显示一行或多行。 默认: False
Name	控件名称。 默认: TextBoxXX
PasswordChar	设置用于隐藏输入的各字符的字符。 默认: 空
ReadOnly	设置用户是否可以编辑文本。 默认: False
ScrollBars	设置如何显示控件的滚动条。 默认: None
ShowPrint	设置是否显示Print语句的输出。 默认: False
TabIndex	控件的选项卡索引。
Text	控件文本。 默认: 空
TextAlign	选择文本对齐。 默认: Left
ToolTipText	控件提示信息中使用的文本。 默认: 空
Top	控件的顶坐标, 单位为像素。
Update	更新文本框控件的显示。
Variable	设置要显示的SPEL+的全局变量(数组除外)。 默认: None
Visible	控件在运行时是否可见。 默认: True
Width	控件宽度, 单位为像素。
WordWrap	设置文字是否换行。 默认: True

## 6.4.4 文本框控件事件

事件	描述
KeyPress	当控件存在焦点并按下按键时，会发生此事件。

## 6.5 单选按钮控件

### 6.5.1 描述

单选按钮控件会为操作员提供一系列两个或多个互斥的选项。

尽管单选按钮和复选框的功能类似，但仍存在重要差异：当用户选择一个单选按钮时，便无法同时选择同一组的其他单选按钮。



### 6.5.2 使用

单选按钮控件应在分组框控件内分组。

每次操作员点击一组中的一个单选按钮时，便取消选定其他按钮。



使用Click事件确定操作员是否点击了单选按钮。使用Checked属性确定用户是否选中了复选框。

在某些情况下，对一组内的所有单选按钮使用一个点击事件处理程序可能会很方便。

您可以使用Sender\$参数确定是否点击了单选按钮。Sender\$是发送事件的控件名称。

```
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
```

设置单选按钮外观

要更改单选按钮的外观，使用以下属性。

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

### 6.5.3 单选按钮控件属性

属性	描述
BackColor	控件的背景颜色。 默认: Control
Checked	设置是否在运行时选择控件。 默认: False
Enabled	设置是否在运行时启用控件。 默认: True
EventTaskType	用于事件的任务类型。 默认: Normal
Font	设置控件文本的字体。 默认: Microsoft Sans Serif 8.25 pt
FontBold	将当前字体显示为粗体。 默认: False
FontItalic	将当前字体显示为斜体。 默认: False
FontName	设置当前字体的名称。 默认: Microsoft Sans Serif
FontSize	以点设置当前字体的大小。 默认: 8.25
ForeColor	控件文本的前景颜色。 默认: ControlText
Height	控件高度, 单位为像素。
Image	控件图片。若要删除, 可按下[删除]键。 默认: 空
ImageAlign	控件的图片对齐。 默认: MiddleCenter
Left	控件的左坐标, 单位为像素。
Name	控件名称。 默认: RadioButtonXX
TabIndex	控件的选项卡索引。
Text	控件文本。 默认: 控件名称
TextAlign	选择文本对齐。 默认: Middle Left
ToolTipText	控件提示信息中使用的文本。 默认: 空
Top	控件的顶坐标, 单位为像素。

属性	描述
Visible	控件在运行时是否可见。 默认: True
Width	控件宽度, 单位为像素。

### 6.5.4 单选按钮控件事件

事件	描述
Click	当控件被点击时执行。

## 6.6 复选框控件

### 6.6.1 描述

复选框控件表示是否打开或关闭某一特定条件。通常用于为用户提供是/否或True/False的选择。

您可以成组使用复选框显示用户可从中选择一个或多个选项的诸多选项。

复选框类似于单选按钮, 但可选择任意数量的分组复选框控件。

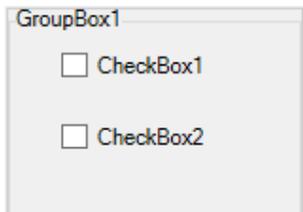


### 6.6.2 使用

使用复选框可使操作员打开或关闭应用选项。

通常情况下, 您将在分组框控件中添加复选框。

使用Checked属性确定用户是否选中了复选框。



#### 设置复选框外观

要更改复选框的外观, 使用以下属性。

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

### 6.6.3 复选框控件属性

属性	描述
BackColor	控件的背景颜色。 默认: Control
Checked	设置是否在运行时选中控件。 默认: False
Enabled	设置是否在运行时启用控件。 默认: True

EventTaskType	用于事件的任务类型。 默认: Normal
Font	设置控件文本的字体。 默认: Microsoft Sans Serif 8.25 pt
FontBold	将当前字体设为粗体。 默认: False
FontItalic	将当前字体设为斜体。 默认: False
FontName	设置当前字体的名称。 默认: Microsoft Sans Serif
FontSize	以点设置当前字体的大小。 默认: 8.25
ForeColor	控件文本的前景颜色。 默认: ControlText
Height	控件高度, 单位为像素。
Image	控件图片。若要删除, 可按下[删除]键。 默认: 空
ImageAlign	控件的图片对齐。 默认: 5 - MiddleCenter
Left	控件的左坐标, 单位为像素。
Name	控件名称。 默认: CheckBoxXX
TabIndex	控件的选项卡索引。
Text	控件文本。 默认: 控件名称
TextAlign	选择文本对齐。 默认: MiddleLeft
ToolTipText	控件提示信息中使用的文本。 默认: 空
Top	控件的顶坐标, 单位为像素。
Visible	控件在运行时是否可见。 默认: True
Width	控件宽度, 单位为像素。

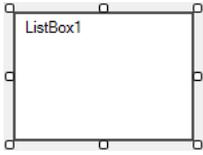
#### 6.6.4 复选框控件事件

事件	描述
Click	用户使用鼠标单击控件时执行。

## 6.7 列表框控件

### 6.7.1 描述

列表框控件会显示操作员可从中选择一个或多个项目的项目列表。



### 6.7.2 使用

使用列表框控件显示操作员可从中选择的项目静态列表。

#### 要在列表框中添加项目

运行时使用AddItem属性在列表框中添加项目。通常会在窗体加载事件中完成。

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

此外，您可以通过将Sorted属性设为True对列表进行分类。

#### 要确定选定的项目

使用SelectedIndex属性确定用户选择。如果未选定任何项目，则SelectedIndex为-1。

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

#### 列表数组

您可以使用List array属性访问列表中的所有项目。

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

您可以通过将List属性设为空字符串来删除项目。

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

#### 设置列表框外观

使用BorderStyle、ForeColor、BackColor、Font属性可更改列表框的外观。

### 6.7.3 列表框控件属性

属性	描述
AddItem	运行时在列表中添加项目。
BackColor	控件的背景颜色。 默认: Window
BorderStyle	用于控件的边框。 默认: Fixed3D

Enabled	设置是否在运行时启用控件。 默认: True
EventTaskType	用于事件的任务类型。 默认: Normal
Font	设置控件文本的字体。 默认: Microsoft Sans Serif 8.25 pt
FontBold	将当前字体设为粗体。 默认: False
FontItalic	将当前字体设为斜体。 默认: False
FontName	设置当前字体的名称。 默认: Microsoft Sans Serif
FontSize	以点设置当前字体的大小。 默认: 8.25
ForeColor	控件文本的前景颜色。 默认: WindowText
Height	控件高度, 单位为像素。
Left	控件的左坐标, 单位为像素。
List	用于在运行时访问列表中的项目。
ListCount	用于在运行时获取列表中的项目数量。
Name	控件名称。 默认: ListBoxXX
SelectedIndex	用于在运行时确定操作员选择的项目。
Sorted	设置是否对列表中的项目分类。 默认: False
TabIndex	控件的选项卡索引。
ToolTipText	控件提示信息中使用的文本。 默认: 空
Top	控件的顶坐标, 单位为像素。
Visible	控件在运行时是否可见。 默认: True
Width	控件宽度, 单位为像素。

#### 6.7.4 列表框控件事件

事件	描述
Click	用户使用鼠标单击控件时执行。
Db1Click	用户使用鼠标双击控件时执行。

## 6.8 组合框控件

### 6.8.1 描述

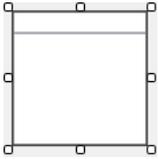
组合框控件用于显示下拉组合框内的数据。

默认情况下，组合框控件包括两部分：上面部分是允许用户键入列表项目的文本框。第二部分是显示用户可从中选择一个项目的列表框。

### 6.8.2 使用

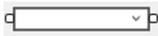
根据DropDownStyle属性的值，组合框控件具有三种不同的使用方法。

DropDownStyle = Simple



在Simple模式下，列表框上方会有一个文本框。操作员可通过在文本框区域键入信息来编辑或添加选择。

DropDownStyle = DropDown



在此模式下，操作员点击文本区域右侧的下箭头按钮之前，不会显示列表框部分。操作员可在文本框区域编辑选定的文本。

DropDownStyle = DropDownList



指定DropDownList时，操作员不可在文本框区域键入文本。仅可选择列表中的项目。

有关如何在组合框控件的列表框部分添加项目的信息，请参阅列表框控件的使用信息，了解AddItem、List、ListCount 和 SelectedIndex的相关信息。

设置组合框外观

使用ForeColor、BackColor和Font属性可更改组合框的外观。

### 6.8.3 组合框控制属性

属性	描述
AddItem	运行时在列表中添加项目。
BackColor	控件的背景颜色。 默认: Window
DropDownStyle	指定组合框的类型。 默认: DropDown
Enabled	设置是否在运行时启用控件。 默认: True
EventTaskType	用于事件的任务类型。 默认: Normal

属性	描述
Font	设置控件文本的字体。 默认: Microsoft Sans Serif 8.25 pt
FontBold	将当前字体设为粗体。 默认: False
FontItalic	将当前字体设为斜体。 默认: False
FontName	设置当前字体的名称。 默认: Microsoft Sans Serif
FontSize	以点设置当前字体的大小。 默认: 8.25
ForeColor	控件文本的前景颜色。 默认: WindowText
Height	控件高度, 单位为像素。
Left	控件的左坐标, 单位为像素。
List	用于在运行时访问列表中的项目。
ListCount	用于在运行时获取列表中的项目数量。
Name	控件名称。 默认: ComboBoxXX
SelectedIndex	用于在运行时确定操作员选择的项目。
Sorted	设置是否对列表中的项目分类。 默认: False
TabIndex	控件的选项卡索引。
Text	控件文本。 默认: 空
ToolTipText	控件提示信息中使用的文本。 默认: 空
Top	控件的顶坐标, 单位为像素。
Visible	控件在运行时是否可见。 默认: True
Width	控件宽度, 单位为像素。

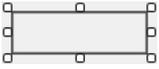
#### 6.8.4 组合框控件事件

事件	描述
Click	单击控件时发生。

#### 6.9 图片框控件

## 6.9.1 描述

图片框控件用于显示位图、GIF、JPEG、PNG或图标格式的图形。



## 6.9.2 使用

使用图片框控件可显示文件中的图像。

您可以在设计或运行时设置图片框控件的图像。

### 设置图像

设计时，使用Image属性在图像列表画面中添加图像文件。可使用ImageIndex属性选择要从已添加图像文件显示的图像文件编号。

要在运行时设置图像文件，需要在Image属性中指定图像文件的完整路径。此外，连接至RC800控制器运行时，通过更改ImageIndex属性的值，可以切换要显示的图像。

### 更改图像尺寸

您可以使用SizeMode属性指定图像尺寸。

### 设置图片框外观

使用BackColor、BorderStyle和SizeMode属性可更改图片框的外观。

## 6.9.3 图片框控件属性

属性	描述
BackColor	控件的背景颜色。 默认: Control
BorderStyle	用于控件的边框。 默认: None
Enabled	设置是否在运行时启用控件。 默认: True
EventTaskType	用于事件的任务类型。 默认: Normal
Height	控件高度，单位为像素。
Image	控件图片。 设计时打开图像列表画面，可以对图像文件进行添加、删除、排序。
ImageIndex	从在图像列表画面中已添加的图像文件中选择要在控件上显示的图像文件。
Left	控件的左坐标，单位为像素。
Name	控件名称。 默认: PictureBoxXX
SizeMode	指定图像尺寸。 默认: Normal
TabIndex	控件的选项卡索引。
ToolTipText	控件提示信息中使用的文本。 默认: 空

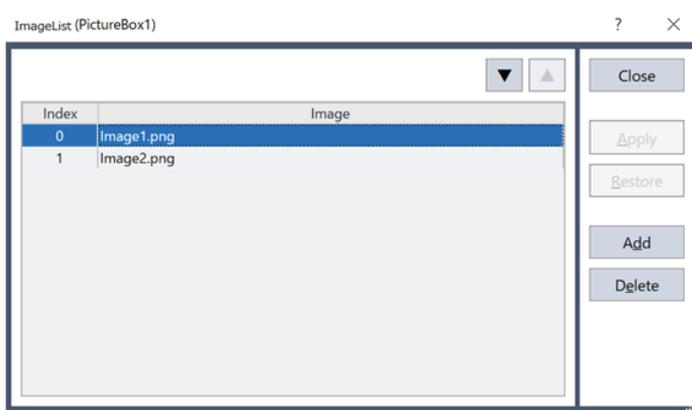
属性	描述
Top	控件的顶坐标，单位为像素。
Visible	控件在运行时是否可见。 默认: True
Width	控件宽度，单位为像素。

### 6.9.4 图片框控件事件

事件	描述
Click	用户使用鼠标单击控件时执行。
DbClick	用户使用鼠标双击控件时执行。

### 6.9.5 图像列表画面

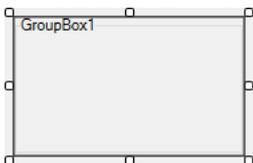
在图像列表画面中，用户可以对要显示到图片框的图像文件进行添加、删除、排序。可以从Image属性打开图像列表画面。



## 6.10 分组框控件

### 6.10.1 描述

分组框控件用于为其他控件提供识别分组。通常情况下，您可以使用分组框按功能划分窗体。在分组框中对所有选项分组可为用户提供合乎逻辑的视觉提示。



### 6.10.2 使用

使用分组框控件可为选择选项（使用RadioButtons）或设置选项（使用CheckBoxes）的控件进行分组。您也可以对其他控件分组。

首先，将分组框控件添加到窗体中。

然后，在分组框上创建新控件，或将现有控件拖至分组框。

将分组框的Text属性设为组名。

## 设置分组框外观

使用BackColor、ForeColor和Font属性可更改分组框的外观。

### 6.10.3 分组框控件属性

属性	描述
BackColor	控件的背景颜色。 默认: Control
Enabled	设置是否在运行时启用控件。 默认: True
EventTaskType	用于事件的任务类型。 默认: 0 - Normal
Font	设置控件文本的字体。 默认: Microsoft Sans Serif 8.25 pt
FontBold	将当前字体设为粗体。 默认: False
FontItalic	将当前字体设为斜体。 默认: False
FontName	设置当前字体的名称。
FontSize	以点设置当前字体的大小。 默认: 8.25
ForeColor	控件文本的前景颜色。 默认: ControlText
Height	控件高度, 单位为像素。
Left	控件的左坐标, 单位为像素。
Name	控件名称。 默认: GroupBoxXX
TabIndex	控件的选项卡索引。
Text	控件文本。 默认: 控件名称
ToolTipText	控件提示信息中使用的文本。 默认: 空
Top	控件的顶坐标, 单位为像素。
Visible	控件在运行时是否可见。 默认: True
Width	控件宽度, 单位为像素。

## 6.10.4 分组框控件事件

事件	描述
Click	用户使用鼠标单击控件时执行。

## 6.11 定时器控件

### 6.11.1 描述

定时器是定期发出事件的控件。即使发生紧急停止或其他正常任务暂停时，定时器控件仍可执行。



### 6.11.2 使用

使用定时器控件可定时执行代码。

例如，定时器控件可用于每两秒更新一次状态标签。

#### 要使用定时器

1. 在窗体中添加定时器控件。由于此控件在运行时不可见，因此会显示在设计区域内的窗体下方。
2. 将Interval属性设为所需时间，单位为毫秒。
3. 如果定时器应始终运行，则在设计时将Enabled属性设为True。否则，可在运行时在代码中将Enabled属性设为True。
4. 添加Tick事件的事件处理程序，以执行代码。

#### 要点

当定时器正在执行Tick事件处理程序时，将忽略同一定时器的其他Tick事件。

### 6.11.3 定时器控件属性

属性	描述
Enabled	设置是否启用控件。 默认：False
Interval	设置时间间隔，单位为毫秒。 默认：100
Name	控件名称。 默认：TimerXX

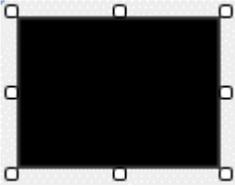
### 6.11.4 定时器控件事件

事件	描述
Tick	定时器启用且达到间隔时发生。

## 6.12 视频框控件

## 6.12.1 描述

视频框控件可实现在窗体内显示Vision Guide选项的视频。



## 6.12.2 使用

通过使用视频框控件可轻松在应用中的窗体上显示视觉窗口。运行视觉顺序时，还可在窗口上显示图形。

执行以下步骤创建视觉窗口：

1. 将视频框控件放在想要显示视频的窗体上。  
控件尺寸可更改为全尺寸。
2. 将VideoEnabled属性设为True。
3. 如果您想要显示视觉图形，则将GraphicsEnabled属性设为True。
4. 默认情况下，Camera属性值为0。  
即允许视频框在序列运行时显示任一相机的视频。将Camera属性设为项目中的任一相机编号，以显示此相机的视频和序列图形。

视频会自动缩放至适合视频框的尺寸。当您更改视频框的高度或宽度时，屏幕比例将保持不变。

### 设置视频框外观

使用BorderStyle属性可更改视频框的外观。

## 6.12.3 视频框控件属性

属性	描述
BorderStyle	用于控件的边框。 默认：None
Camera	选择将显示视频的相机。 默认：0
Enabled	设置是否在运行时启用控件。 默认：True
EventTaskType	用于事件的任务类型。 默认：Normal
GraphicsEnabled	设置是否显示视觉图形。 默认：False
Height	控件高度，单位为像素。
Left	控件的左坐标，单位为像素。
Name	控件名称。 默认：VideoBoxXX
TabIndex	控件的选项卡索引。
ToolTipText	控件提示信息中使用的文本。

属性	描述
Top	控件的顶坐标，单位为像素。
VideoEnabled	设置是否显示视频。 默认：False
Visible	控件在运行时是否可见。 默认：True
Width	控件宽度，单位为像素。

## 6.12.4 视频框控件事件

事件	描述
Click	用户使用鼠标单击控件时执行。
Db1Click	用户使用鼠标双击控件时执行。

## 6.13 LED控件

### 6.13.1 描述

LED 控件用于显示I/O状态。



### 6.13.2 使用

要使用LED控件

1. 将Text属性设为状态名称。
2. 设置IOType属性。  
可以选择input, output, 或memory I/O。
3. 设置IOBit属性。即想要显示状态的位。

对于输出，您可以视情况允许操作员双击LED控件切换输出状态。为此，需将AllowStateChange属性设为True。

#### 设置LED外观

要更改LED控件的外观，使用以下属性。

BackColor、BorderStyle、ForeColor、Font、ImageAlign、TextAlign

您还可以更改ImageOn和ImageOff属性，以使用内置彩色图像，或者可以提供自己的图像。

### 6.13.3 LED控件属性

属性	描述
AllowStateChange	允许操作员切换输出状态。 默认：False
BackColor	控件的背景颜色。 默认：Control

属性	描述
BorderStyle	用于控件的边框。 默认: None
Enabled	设置是否在运行时启用控件。 默认: True
EventTaskType	用于事件的任务类型。 默认: Normal
Font	设置控件文本的字体。 默认: Microsoft Sans Serif 8.25 pt
FontBold	将当前字体设为粗体。 默认: False
FontItalic	将当前字体显示为斜体。 默认: False
FontName	设置当前字体的名称。 默认: Microsoft Sans Serif
FontSize	以点设置当前字体的大小。 默认: 8.25
ForeColor	控件文本的前景颜色。 默认: ControlText
Height	控件高度, 单位为像素。
ImageAlign	控件的图片对齐。 默认: MiddleLeft
ImageOff	设置I/O状态关闭时显示的图像。 默认: LedOff.ico
ImageOn	设置I/O状态打开时显示的图像。 默认: LedRed.ico
IOBit	范围: 0 至 9999 默认: 0
IOType	设置待监测I/O的类型。 默认: Input
Left	控件的左坐标, 单位为像素。
Name	控件名称。 默认: LedXX
TabIndex	控件的选项卡索引。
Text	控件文本。 默认: 控件名称
TextAlign	选择文本对齐。 默认: MiddleRight

属性	描述
ToolTipText	控件提示信息中使用的文本。 默认：空
Top	控件的顶坐标，单位为像素。
Visible	控件在运行时是否可见。 默认：True
Width	控件宽度，单位为像素。

### 6.13.4 LED控件事件

事件	描述
Db1Click	用户使用鼠标双击控件时执行。

## 6.14 状态栏控件

### 6.14.1 描述

状态栏控件用于显示文本面板和选项面板(日期、紧急停止状态、安全门状态和机器人信息)。



### 6.14.2 使用

使用状态栏控件，显示任意文本和其他内置状态。文本面板始终显示。

使用Text属性，设置要在文本面板中显示的字符串。默认情况下，其他文本面板（日期、紧急停止状态、安全门状态和机器人信息）不会显示。

使用ShowDateTime, ShowEStop, ShowRobot和ShowSafeguard属性，可以显示或隐藏这些面板。系统会自动更新这些面板。每个表单只能使用一个状态栏。如果表单中已经使用了一个状态栏，则工具栏上的“新建状态栏”按钮将呈灰色且无法选择。

### 6.14.3 状态栏控件属性

属性	描述
BackColor	控件的背景颜色。 默认：Control
BorderStyle	设置控件的边框样式。 默认：None
Enabled	设置是否启用控件。 默认：True
EventTaskType	设置用于事件的任务类型。 默认：Normal
Font	设置控件文本的字体。 默认：Microsoft Sans Serif 8.25 pt

属性	描述
ForeColor	设置控件的文本颜色。 默认: ControlText
Height	设置控件的高度, 以像素为单位。
Left	设置控件显示位置的左坐标, 以像素为单位。
Name	设置控件的名称。 默认: StatusBarXX
RobotNumber	设置用于ShowRobot的机器人编号。
ShowDateTime	设置是否显示当前日期和时间。 默认: False
ShowEStop	设置是否显示Estop状态。 默认: False
ShowRobot	设置是否显示由RobotNumber指定的机器人。 默认: False
ShowSafeguard	设置是否显示安全门状态。 默认: False
TabIndex	设置控件的选项卡索引。
Text	设置控件的显示文本。 默认: 控件名称
ToolTipText	设置控件的工具提示中显示的文本。 默认: 空
Top	设置控件显示位置的上坐标, 以像素为单位。
Visible	控件在运行时是否可见。 默认: True
Width	设置控件的宽度, 以像素为单位。

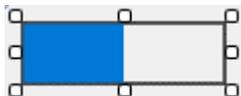
#### 6.14.4 状态栏控件事件

事件	描述
Click	单击控件时执行。

### 6.15 进度条控件

#### 6.15.1 描述

使用进度条控件显示任务的处理进度。



## 6.15.2 使用

使用进度条控件显示长时间运行的操作的状态和变量的值。

### 如何显示SPEL+ 的全局变量(数组除外)

通过设置Variable属性自动显示SPEL+全局变量的值。

注意

如果SPEL+的全局变量未先显示在Variable属性中时，请重新创建项目。

## 6.15.3 进度条控件属性

属性	描述
BackColor	控件的背景颜色。 默认: Control
BorderStyle	设置控件的边框样式。 默认: None
Enabled	设置是否启用控件。 默认: True
EventTaskType	设置用于事件的任务类型。 默认: Normal
ForeColor	设置控件的文本颜色。 默认: ControlText
Height	设置控件的高度，以像素为单位。
Left	设置控件显示位置的左坐标，以像素为单位。
Maximum	设置控件正在处理的范围的上限。 默认: 100
Minimum	设置控件正在处理的范围的下限。 默认: 0
Name	设置控件的名称。 默认: ProgressBarXX
Orientation	设置控件的方向。 默认: Horizontal
ProgressBarStyle	设置控件的样式。 默认: Continuous
TabIndex	设置控件的选项卡索引。
ToolTipText	设置控件的工具提示中显示的文本。 默认: 空
Top	设置控件显示位置的上坐标，以像素为单位。
Value	设置显示控件当前位置的值。 默认: 50

属性	描述
Variable	设置要显示的SPEL+的全局变量(数组除外)。 默认: None
Visible	设置显示或隐藏控件。 默认: True
Width	设置控件的宽度, 以像素为单位。

## 6.15.4 进度条控件事件

事件	描述
Click	单击控件时执行。
DoubleClick	双击控件时执行。

## 6.16 跟踪栏控件

### 6.16.1 描述

跟踪栏控件用于拖动滑块以设置相应的值, 并显示该设置。



### 6.16.2 使用

如需以图形方式变更值, 请使用跟踪栏控件。

#### 如何显示SPEL+ 的全局变量(数组除外)

通过设置Variable属性, 可以根据跟踪栏中的值自动设置SPEL+全局变量的值。

注意

如果SPEL+的全局变量未在Variable属性中更新, 请重新创建项目。

### 6.16.3 跟踪栏控件属性

属性	描述
BackColor	控件的背景颜色。 默认: Control
Enabled	设置控件的边框样式。 默认: True
EventTaskType	设置用于事件的任务类型。 默认: Normal
Height	设置控件的高度, 以像素为单位。
LargeChange	设置单击鼠标, 或按Page Up和Page Down键时候, 滑块移动的位置数。 默认: 5
Left	设置控件显示位置的左坐标, 以像素为单位。

属性	描述
Maximum	设置控件正在处理的范围的上限。 默认: 10
Minimum	设置控件正在处理的范围的下限。 默认: 0
Name	设置控件的名称。 默认: TrackBarXX
Orientation	设置控件的方向。 默认: Horizontal
SmallChange	设置使用键盘输入(方向键)时, 滑块移动的位置数。 默认: 1
TabIndex	设置控件的选项卡索引。
TickFrequency	设置刻度线之间的位置数。 默认: 1
TickStyle	设置在跟踪栏上显示刻度线的位置。 默认: BottomRight
ToolTipText	设置控件的工具提示中的显示文本。 默认: 空
Top	设置控件显示位置的上坐标, 以像素为单位。
Value	设置显示控件当前位置的值。 默认: 0
Variable	设置要显示的SPEL+的全局变量(数组除外)。 默认: None
Visible	设置显示或隐藏控件。 默认: True
Width	设置控件的宽度, 以像素为单位。

#### 6.16.4 跟踪栏控件事件

事件	描述
Scroll	跟踪栏的滑块移动时, 执行滚动。

#### 6.17 网格控件

## 6.17.1 描述

网格控件用于以Excel格式，显示和编辑数据。

Column 0	Column 1

## 6.17.2 使用

网格控件具有包含行和列数据的单元格。操作人员可选择行和列，自由编辑指定范围内的单元格。

设置网格控件：

1. 点击GridEditor属性，打开[GridEditor]对话框。
2. 将属性设置为您需要的值，设计网格。
3. 关闭对话框。

执行：

- 使用CellTex属性，读取和写入单元格文本。
- 使用AddRow属性和RemoveRow属性，可添加或删除行。
- 使用CellForeColor属性和CellBackColor属性，更改单元格的ForeColor和BackColor。

## 6.17.3 网格控件属性

属性	描述
AddRow	在执行时添加行。
BorderStyle	设置控件的背景颜色。 默认：Control
CellBackColor	设置或获取单元格的背景颜色。
CellForeColor	设置或获取单元格的ForeColor。
CellText	设置或获取单元格中的文本。
Enabled	设置是否启用控件。 默认：True
EventTaskType	设置是否在事件中激活任务类型。 默认：Normal
GridEditor	打开[GridEditor]对话框，设置网格控件。
Height	以像素为单位，设置控件的高度。
Left	以像素为单位，设置控件显示位置的左侧的坐标。
Name	设置控件名称。 默认：GridXX
RemoveRow	删除行。
RowCount	设置或获取行数。

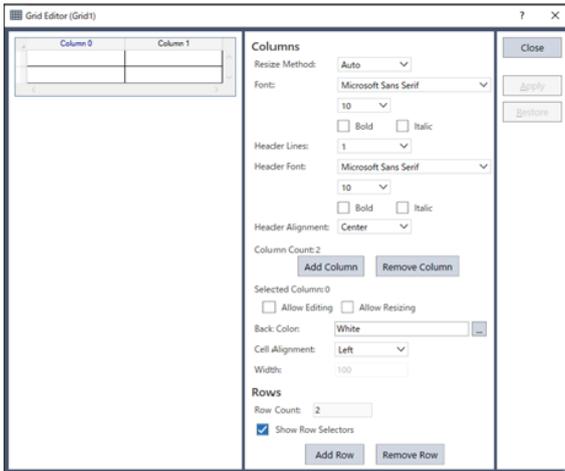
属性	描述
ScrollBars	显示滚动条时设置。 默认: None
SelectedIndex	设置或获取选定的行。
Top	以像素为单位设置控件显示位置的上方坐标。
Visible	设置是否显示控件。 默认: True
Width	以像素为单位, 设置控件的宽度。

### 6.17.4 网格控件事件

事件	描述
CellChanged	当离开修改后的单元格时执行。
Click	当控件被点击时执行。
Db1Click	当控件被双击时执行。

### 6.17.5 网格编辑器

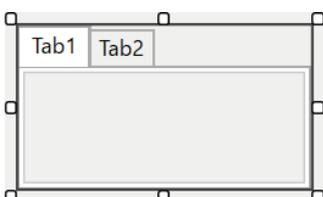
网格编辑器是一个对话框, 允许用户修改和设置网格控件。该功能仅能在设计时使用。  
为达到最佳设定状态, 有很多属性可供选择。有关网格编辑器的详细信息, 请参阅GridEditor属性。



## 6.18 选项卡控件

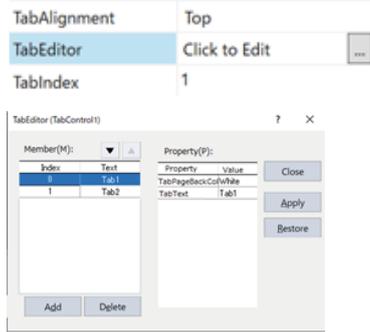
### 6.18.1 描述

选项卡控件可以创建可切换选项卡的页面。可针对设置及确认状态等目的, 创建可切换显示页面的画面。



## 6.18.2 使用

使用选项卡控件，创建各种目的的页面。进行选项卡控件的布局后，按下TabEditor属性的按钮打开选项卡编辑器。



使用选项卡编辑器进行选项卡页面的添加和删除。此外，设置各选项卡的文本或字体及选项卡页面背景颜色后，关闭选项卡编辑器。更改选项卡控件的大小及位置后，选择要添加页面对应的选项卡。显示选项卡页面后，在选项卡页面上布局要显示的控件。

## 6.18.3 选项卡控件属性

属性	描述
Enabled	设置是否启用控件。 默认: True
EventTaskType	设置用于事件的任务类型。 默认: Normal
Height	设置控件的高度，以像素为单位。
Left	设置控件显示位置的左坐标，以像素为单位。
Name	设置控件的名称。 默认: PictureBoxXX
SelectedIndex	用于在运行时确定操作员选择的项目。
TabAlignment	对选项卡控件设置选项卡的布局位置。 默认: Top
TabEditor	打开进行选项卡的添加、删除及排序所需的选项卡编辑器画面。
TabIndex	设置控件的选项卡索引。
ToolTipText	设置控件的工具提示中显示的文本。 默认: 空
Top	设置控件显示位置的上坐标，以像素为单位。
Visible	设置显示或隐藏控件。 默认: True
Width	设置控件的宽度，以像素为单位。

## 7. 操作

## 7.1 概述

本章包括以下主题：

- 编程模式下的GUI开发
- 自动模式下的GUI启动配置
- 处理暂停和继续
- 处理紧急停止
- 使用帮助文件

## 7.2 编程模式下的GUI开发

您可以在编程模式下设计和调试GUI应用。  
开始前，需在编程模式下启动Epson RC+ 8.0。

### 7.2.1 设计GUI

若要为应用设计GUI，需遵循以下步骤作为简单指南。

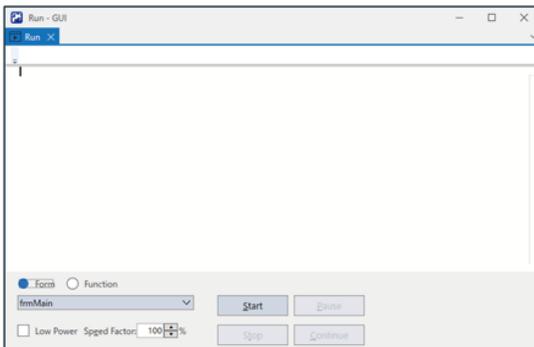
1. 为应用创建一个或多个窗体。通常需指定其中一个窗体为主窗体。
2. 确定想要启动GUI的方式。您可以将其中一个窗体设为启动窗体。  
当RC+ 在自动模式下启动时，将自动显示此窗体。  
如果您不想在启动时显示窗体，则会显示操作员窗口，并且您可以使用GShow或GShowDialog显示SPEL+ 代码中的任一窗体。
3. 为窗体添加控件。
4. 为控件添加所需事件。
5. 有关使用窗体和控件的详细信息，请参阅以下章节。

[窗体和控件的使用](#)

### 7.2.2 调试

您可以通过运行窗口运行项目中的任一窗体。

1. 打开运行窗口。



2. 从下拉列表中选择想要运行的窗体。
3. 点击[开始]。将显示选定窗体，并且您可以使用窗体上的控件。
4. 若要停止，可关闭所有窗体，或点击运行窗口上的[停止]按钮。

GUI 事件处理程序会作为 SPEL+ 任务运行，因此您可以在任一事件处理程序中设置断点，单步调试代码并查看变量值。

## 7.3 自动模式

在开发应用之后，您将需要设置Epson RC+ 在自动模式下运行。启动时，您可以选择显示其中一个GUI窗体，或者您可以显示操作员窗口，然后显示程序中的其中一个窗体。

**\*\* 要设置Epson RC+ 在自动模式下启动\*\***

1. 打开设置的[系统配置]。
2. 在启动模式页面，选中[自动]并点击[应用]，然后点击[关闭]。

### 要配置启动时显示的窗体

1. 打开GUI Builder窗口。
2. 在窗体资源管理器中，右击想要在启动时显示的窗体，并选择“启动窗体”。
3. 点击主工具栏上的[项目保存]按钮。

## 7.4 处理暂停和继续

您可能会想要在GUI中使用控制器的暂停和继续功能。

通常，会有一个具有暂停按钮和继续按钮的对话框。若要使这些按钮正常工作，需要将各按钮的EventTaskType属性设为1 - NoPause。

这是因为按钮点击事件执行时，事件任务必须忽略控制器的暂停状态。

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

## 7.5 处理紧急停止

紧急停止后，您必须在可以重新运行机器人之前重置紧急停止条件。

为此，需提供重置按钮，并将其EventTaskType设为2 - NoEmgAbort。

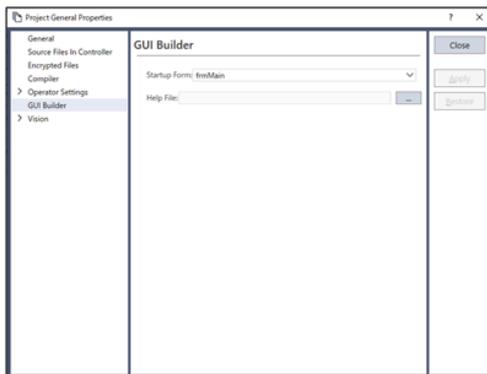
```
Function frmMain_btnReset_Click(Sender$ As String)
    Reset
    If EstopOn Then
        MsgBox "EStop could not be reset"
    EndIf
Fend
```

## 7.6 使用帮助文件

GUI Builder支持简单的帮助系统，即允许您为项目指定帮助文件，以在操作员点击窗体帮助按钮时显示主题。帮助文件必须采用Microsoft HTML帮助格式。

要使用帮助文件：

1. 打开项目-[属性]-[GUI创建器]。
2. 浏览到帮助文件并点击打开以设置帮助文件。



3. 对于将显示帮助的主题，将HelpButton属性设为True并为想要显示的主题设置HelpID属性。

## 8. GUI Builder参考

## 8.1 概述

本章将介绍所有GUI Builder窗体和控件属性和事件以及所有GUI Builder SPEL+ 命令。有关如何使用GUI Builder的更多信息，请参阅GUI Builder手册。

## 8.2 GUI Builder属性和事件格式描述

所有GUI Builder属性和事件将在后文列出。属性标题以及结果参考页码的说明如下所示：

项目	描述
使用范围	如果属性或事件用于GUI Builder对象，则本节将简单列出此属性适用的对象。 (例. Button、Label、CheckBox...) 属性或事件被用于窗体时，显示Form。
描述	将为每个属性或事件提供简单描述。为简单起见，本节内容通常很短。
使用	使用章节将介绍如果通过SPEL+ 语言访问属性或事件。
数值	描述属性可设至或结果将返回的可接受数值范围。对于具有默认值的属性，还会显示默认值。
备注	介绍描述章节未涉及的更多详细信息。本节通常用于描述可能适用于特定属性或事件的所有警告或特殊信息。(强烈建议在使用各属性之前阅读相关的备注章节。)
另见	提供一系列可能有助于回顾的相关属性、结果、对象和其他主题。
仅运行时	在适用时显示在属性或结果名称下方。通过GUI Builder无法访问仅运行时的属性和结果。只能通过SPEL+ 语言访问。

## 8.3 A

### 8.3.1 AcceptButton属性

#### 适用范围

Form

#### 描述

获取或设置用户按下ENTER键时点击的窗体按钮。

#### 使用

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

Form

指定窗体名称或含有窗体名称的字符串变量。

var

指定要取得属性值的字符串变量。

value

指定要在属性中设置的字符串表达式。

#### 数值

窗体上任一按钮的名称。

默认: None

#### 备注

该属性允许您指定用户在应用中按下ENTER键时发生的默认动作。分配至该属性的按钮必须是位于当前窗体或位于当前窗体上的容器内的按钮。

您可以使用该属性允许用户在完成时仅按下ENTER键而无需用鼠标手动点击接受按钮，从而允许用户快速导航简单窗体。如果窗体上当前选定的控件拦截 ENTER 键并对其进行处理，则接受按钮可能未激活。例如，多行文本框控件允许在控件中插入新一行字符时按下ENTER键。

#### 另见

Form、Button、CancelButton

#### 示例

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

## 8.3.2 AddItem属性

### 适用范围

ListBox、ComboBox

### 描述

用于在列表框或组合框控件中添加项目。

### 使用

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

Form

指定窗体名称或含有窗体名称的字符串变量。

Control

指定控件名称或含有控件名称的字符串。控件必须位于指定窗体内。

value

指定要在属性中设置的字符串表达式。

### 数值

待添加至列表的字符串值。

### 另见

ComboBox、ListBox、List、ListCount、Sorted

### 示例

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

### 8.3.3 AddRow 属性

#### 适用范围

Grid

#### 描述

在网格控件中添加行。

#### 使用

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

Form

指定窗体名称或含有窗体名称的字符串变量。

Control

指定控件名称或含有控件名称的字符串。控件必须位于指定窗体内。

value

表示新添加行的索引的整数值。如果指定为“-1”，则在最后一行之后添加新的行。否则将在指定索引中添加新的行。

#### 另见

Grid, RemoveRow, RowCount

#### 示例

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

### 8.3.4 AppendText属性

#### 适用范围

TextBox

#### 描述

在文本框控件上附加文本。

#### 使用

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

Form

指定窗体名称或含有窗体名称的字符串变量。

Control

指定控件名称或含有控件名称的字符串。控件必须位于指定窗体内。

value

指定要在属性中设置的字符串表达式。

#### 数值

将添加至控件中文本结尾处的字符串。

#### 另见

TextBox

#### 示例

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

### 8.3.5 AllowStateChange属性

#### 适用范围

LED

#### 描述

获取或设置双击时LED控件的反转驱动。

#### 使用

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

Form

指定窗体名称或含有窗体名称的字符串变量。

Control

指定控件名称或含有控件名称的字符串。控件必须位于指定窗体内。

var

指定要取得属性值的逻辑变量。

value

指定要在属性中设置的逻辑表达式。

#### 数值

False

True

默认: False

#### 备注

如果IOType属性被设置为0 - Input, 则即使将AllowStateChange属性设为True, 也无法转换输出。若要转换输出, 需将IOType属性设为1 - Output或2 - Memory。

#### 另见

LED、IOType

#### 示例

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

## 8.4 B

### 8.4.1 BackColor属性

#### 适用范围

Form、Button、Label、TextBox、RadioButton、CheckBox、ListBox、ComboBox、PictureBox、GroupBox、LED、ProgressBar、TrackBar、StatusBar

#### 描述

获取或设置窗体或控件的背景颜色。

#### 使用

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

Form

指定窗体名称或含有窗体名称的字符串变量。

Control

指定控件名称或含有控件名称的字符串。

var

指定要取得属性值的字符串变量。

value

指定要在属性中设置的字符串表达式。

#### 数值

用于窗体或控件背景颜色的颜色名称。

默认:

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

#### 另见

BackColorMode、Form、Button、Label、TextBox、RadioButton、Checkbox、ListBox、ComboBox、PictureBox、GroupBox、LED、ProgressBar、TrackBar、StatusBar

#### 示例

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

## 8.4.2 BackColorMode属性

### 适用范围

Button

### 描述

获取或设置用于按钮控件的BackColorMode。

### 使用

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

Form

指定窗体名称或含有窗体名称的字符串变量。

Control

指定控件名称或含有控件名称的字符串。

var

取得属性值的整数变量。

value

在属性中设定的整数表达式。

### 数值

0 -VisualStyle. 使用系统指定的BackColor。

1 - User. 使用指定的BackColor。

默认: Visual Style

### 备注

当BackColorMode为0时, 会根据当前方案由Windows设置控件的背景颜色。

当BackColorMode为1时, 会由BackColor属性值设置背景颜色。

### 另见

BackColor、Button

### 示例

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

### 8.4.3 BorderStyle属性

#### 适用范围

Label、TextBox、ListBox、PictureBox、VideoBox、LED、ProgressBar、StatusBar、Grid

#### 描述

获取或设置控件的边框样式。

#### 使用

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

Form

指定窗体名称或含有窗体名称的字符串变量。

Control

指定控件名称或含有控件名称的字符串。

var

取得属性值的整数变量。

value

在属性中设定的整数表达式。

#### 数值

0 - None

1 - FixedSingle

2 - Fixed3D

默认:

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

#### 备注

您可以使用该属性为控件添加边框。例如，该属性可用于区分标识其他控件的标签和显示应用中过程状态的标签。

#### 另见

Label、TextBox、ListBox、PictureBox、VideoBox、LED、ProgressBar、StatusBar

#### 示例

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

## 8.5 C

### 8.5.1 Camera属性

#### 适用范围

VideoBox

#### 描述

获取或设置视频框控件的相机编号。

#### 使用

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

Form

指定窗体名称或含有窗体名称的字符串变量。

Control

指定控件名称或含有控件名称的字符串。

var

取得属性值的整数变量。

value

在属性中设定的整数表达式。

#### 数值

用于视频显示的相机编号。如果数值为0，则会在VRun执行时显示任一相机的视频。若要显示图像，需将VideoEnabled属性设为“True”。

#### 另见

VideoBox、VideoEnabled

#### 示例

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

## 8.5.2 CancelButton属性

### 适用范围

Form

### 描述

获取或设置用户按下ESC键时点击的窗体按钮。

### 使用

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

Form

指定窗体名称或含有窗体名称的字符串变量。

var

指定要取得属性值的字符串变量。

value

指定要在属性中设置的字符串表达式。

### 数值

窗体上任一按钮的名称。

默认: None

### 备注

窗体的取消按钮是无论何时用户按下ESC键均会点击的按钮控件。分配至该属性的按钮必须是位于当前窗体或位于当前窗体上的容器内的按钮。

该属性允许您指定用户在应用中按下ESC键时发生的默认动作。您可以使用该属性允许用户仅通过按下ESC键来关闭窗口，而无需用鼠标手动点击取消按钮来提交更改，从而允许用户快速导航简单窗体。

CancelButton在窗体上的其他控件拦截ESC键时可能会无效。例如，如果您的窗体上打开了一个组合框，则ESC将关闭组合框，而不会关闭窗口。

分配至CancelButton的按钮控件必须在窗体上可见，否则按下ESC键将无效。

### 另见

Form、Button、AcceptButton

### 示例

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

### 8.5.3 CellBackColor属性

#### 适用范围

Grid

#### 描述

设置或获取单元格的背景颜色。

#### 使用

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

Form

指定窗体名称或含有窗体名称的字符串变量。

Control

指定控件名称或含有控件名称的字符串。

row

指定单元格行的整数表达式。第一行的索引是“0”。

column

指定单元格列的整数表达式。第一列的索引是“0”。

var

指定要取得属性值的字符串变量。

value

指定要在属性中设置的字符串表达式。

#### 数值

包含指定网格单元的背景颜色名称的字符串。

#### 另见

Grid, CellForeColor, CellText

#### 示例

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

## 8.5.4 CellChanged事件

### 适用范围

Grid

### 描述

当离开修改后的单元格时发生的事件。

### 使用

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

Sender\$

被设置为引发事件的控件的名称。

CellText\$

被设置为已更改单元格的文本。

RowNumber

被设置为已更改单元格的行号。

ColumnNumber

被设置为已更改单元格的列号。

### 备注

CellChanged事件用于，当用户离开了更改的单元格时做出相应。使用Sender\$参数，可确定时哪个控件发送了事件。使用CellText\$参数，可以查看单元格的当前值。使用RowNumber参数和ColumnNumber参数，可以确定时哪个单元格引发了事件。当您向查看特定单元格或单元格何时发生更改时，可使用此功能。

### 另见

Grid

### 示例

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

## 8.5.5 CellForeColor属性

### 适用范围

Grid

### 描述

设置或获取网格单元格的文本颜色。

### 使用

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

Form

指定窗体名称或含有窗体名称的字符串变量。

Control

指定控件名称或含有控件名称的字符串。

row

指定单元格行的整数表达式。第一行的索引是“0”。

column

指定单元格列的整数表达式。第一列的索引是“0”。

var

指定要取得属性值的字符串变量。

value

指定要在属性中设置的字符串表达式。

### 数值

包含指定网格单元格的文本颜色名称的字符串。

### 另见

Grid, CellBackColor, CellText

### 示例

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

## 8.5.6 CellText属性

### 适用范围

Grid

### 描述

设置或获取网格单元格中的文本。

### 使用

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

Form

指定窗体名称或含有窗体名称的字符串变量。

Control

指定控件名称或含有控件名称的字符串。

row

指定单元格行的整数表达式。第一行的索引是“0”。

column

指定单元格列的整数表达式。第一列的索引是“0”。

var

指定要取得属性值的字符串变量。

value

指定要在属性中设置的字符串表达式。

### 数值

包含特定网格单元的文本的字符串。

### 另见

Grid, CellForeColor, CellBackColor

### 示例

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

## 8.5.7 Checked属性

### 适用范围

RadioButton、Checkbox

### 描述

获取或设置表示是否选中复选框或单选按钮控件的值。

### 使用

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

Form

指定窗体名称或含有窗体名称的字符串变量。

Control

指定控件名称或含有控件名称的字符串。

var

指定要取得属性值的逻辑变量。

value

指定要在属性中设置的逻辑表达式。

### 数值

False

True

默认: False

### 备注

使用该属性可确定是否选中了复选框或单选按钮控件。该属性还可用于以编程方式设置复选框或单选按钮控件的状态。

### 另见

RadioButton、CheckBox

### 示例

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

## 8.5.8 Click事件

### 适用范围

Button、Label、RadioButton、Checkbox、ListBox、ComboBox、PictureBox、GroupBox、VideoBox、ProgressBar、StatusBar、Grid

### 描述

单击控件时发生。

### 使用

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

Sender

被设置为引发事件的控件的名称。

### 备注

Click事件用于响应用户对控件的单击。您可以使用Sender\$参数确定发送事件的控件。这对于想要使用一个函数处理多个控件的情况非常有用，如，对于单选按钮和复选框。

使用一个函数创建处理更加容易。

### 另见

Button、Label、RadioButton、Checkbox、ListBox、ComboBox、PictureBox、GroupBox、VideoBox、ProgressBar、StatusBar

### 示例

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

## 8.5.9 Closed事件

### 适用范围

Form

### 描述

窗体关闭时发生。

### 使用

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

Sender

被设置为引发事件的窗体的名称。

### 备注

您可以使用该事件在窗体关闭之后执行任务。

### 另见

Form、Load、Resize

### 示例

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

## 8.5.10 ControlBox属性

### 适用范围

Form

### 描述

获取或设置表示是否在窗体标题栏显示控件框的值。

### 使用

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

Form

指定窗体名称或含有窗体名称的字符串变量。

var

指定要取得属性值的逻辑变量。

value

指定要在属性中设置的逻辑表达式。

### 数值

False

True

默认: True

### 备注

如果ControlBox属性被设为True, 则会在窗体的标题栏左上角显示控件框。控件框可用于关闭窗口体。

### 另见

Form、MaximizeBox、MinimizeBox

### 示例

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

## 8.5.11 Controls属性

### 适用范围

Form

### 描述

窗体上的控件数组。用于通过索引访问控件属性。

### 使用

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

Form

指定窗体名称。

Index

指定含有窗体上控件索引的整数表达式。

Property

指定要访问的控件属性名称。

var

指定要取得属性值的变量。数据类型取决于指定的属性。

value

指定要在属性中设置的表达式。数据类型取决于指定的属性。

### 备注

使用Controls属性可通过索引访问窗体上的控件。这样便允许您循环访问控件集合并获取或设置一般属性。

### 另见

Count、Type

### 示例

```
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
```

## 8.5.12 Count属性

### 适用范围

Form.Controls

### 描述

获取窗体上的控件数量。

### 使用

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

Form

指定窗体名称。

var

指定要取得属性值的变量。

### 数值

返回窗体上的控件数量。

### 备注

使用Count属性可确定窗体上的控件数量。然后可以使用Controls属性循环访问所有控件。

### 另见

Controls、Type

### 示例

```
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 DbIClick事件

#### 适用范围

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

#### 描述

双击控件时发生。

#### 使用

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

#### Sender

被设置为引发事件的控件的名称。

#### 备注

DblClick 事件用于响应用户对控件的双击。您可以使用Sender\$参数确定发送事件的控件。

#### 另见

Label、ListBox、PictureBox、VideoBox、LED、ProgressBar

#### 示例

```
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属性

### 适用范围

Form

### 描述

获取或设置窗体的对话框结果。

### 使用

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

Form

指定窗体名称或含有窗体名称的字符串变量。

var

取得属性值的整数变量。

value

在属性中设定的整数表达式。

### 数值

0 - None

1 - OK

2 - Cancel

默认: Cancel

### 备注

DialogResult用于表示用户是否接受窗体或取消窗体。

通常，一个对话框具有一个[确定]按钮和一个[取消]按钮。当用户单击[确定]按钮时，DialogResult 属性会在按钮单击事件中设为1—OK。如果单击[取消]按钮，则DialogResult属性会设为2 - Cancel。

### 另见

Form

### 示例

```
Function frmSetSpeed_btnOK_Click(Sender$ As String)  
    GSet frmSetSpeed.DialogResult, DIALOGRESULT_OK  
    GClose frmSetSpeed  
End  
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  
End
```

### 8.6.3 DropDownStyle属性

#### 适用范围

ComboBox

#### 描述

获取或设置指定组合框类型的值。

#### 使用

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

Form

指定窗体名称或含有窗体名称的字符串变量。

Control

控件名称或含有控件名称的字符串变量。

var

取得属性值的整数变量。

value

在属性中设定的整数表达式。

#### 数值

0 - Simple

1 - DropDown

2 - DropDownList

默认: DropDown

#### 备注

DropDownStyle属性可控制为用户展示的界面。

从以下选择。

- 输入数值提供始终显示列表的简单下拉列表框
- 文本部分不可编辑且必须选择箭头查看下拉列表的下拉列表框
- 文本部分可编辑且用户必须按下箭头键查看列表的默认下拉列表框。

若要始终显示用户不可编辑的列表，使用ListBox控件。

#### 另见

ComboBox

#### 示例

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

## 8.7 E

### 8.7.1 Enabled属性

#### 适用范围

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

#### 描述

获取或设置是否在运行时启用控件。

#### 使用

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

#### Form

指定窗体名称或含有窗体名称的字符串变量。

#### Control

指定控件名称或含有控件名称的字符串。

#### var

指定要取得属性值的逻辑变量。

#### value

指定要在属性中设置的逻辑表达式。

#### 数值

False

True

默认: True (Timer以外) False (Timer)

#### 备注

在某些情况下, 需要防止操作员单击控件, 如按钮。例如, SPEL+任务未运行时, 请将Enabled属性设为False, 使[Pause]按钮为非按下状态。

#### 另见

Button, Label, TextBox, RadioButton, Checkbox, ListBox, ComboBox, PictureBox, GroupBox, Timer, VideoBox, LED, ProgressBar, TrackBar, StatusBar, TabControl

#### 示例

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

## 8.7.2 EventTaskType属性

### 适用范围

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

### 描述

获取以事件开始的函数的任务类型。

### 使用

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

Form

指定窗体名称或含有窗体名称的字符串变量。

Control

指定控件名称或含有控件名称的字符串。

var

取得属性值的整数变量。

### 数值

0 - Normal

1 - NoPause

2 - NoEmgAbort

默认: Normal

### 备注

当该属性设为1 - NoPause时, 事件处理程序函数在其他任务暂停时仍可运行。当该属性设为2 - NoEmgAbort时, 事件处理程序函数在紧急停止状态下仍可运行。

### 另见

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

### 示例

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

## 8.8 F

### 8.8.1 Font属性

此属性仅在设计GUI上可用。

#### 适用范围

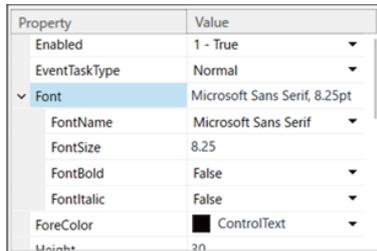
Button、Label、TextBox、RadioButton、CheckBox、ListBox、ComboBox、GroupBox、LED、StatusBar

#### 描述

Font属性允许在设计时更改控件的字体参数。(字体名称、类型、大小)

#### 使用

可以通过更改GUI Builder窗口属性网格中显示的Font属性的值来更改显示字符的字体。



Property	Value
Enabled	1 - True
EventTaskType	Normal
Font	Microsoft Sans Serif, 8.25pt
FontName	Microsoft Sans Serif
FontSize	8.25
FontBold	False
FontItalic	False
ForeColor	ControlText
MinSize	20

更改字体设置并保存项目后，设置将被保存。

#### 另见

Button、Label、TextBox、RadioButton、Checkbox、ListBox、ComboBox、GroupBox、LED、StatusBar

## 8.8.2 FontBold属性

### 适用范围

Button、Label、TextBox、RadioButton、CheckBox、ListBox、ComboBox、GroupBox、LED、StatusBar

### 描述

获取或设置控件上使用的文本字体在运行时是否采用粗体。

### 使用

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

Form

指定窗体名称或含有窗体名称的字符串变量。

Control

指定控件名称或含有控件名称的字符串。

var

指定要取得属性值的逻辑变量。

value

指定要在属性中设置的逻辑表达式。

### 数值

False

True

默认: False

### 另见

Button、Label、TextBox、RadioButton、Checkbox、ListBox、ComboBox、GroupBox、LED、StatusBar、FontName、FontItalic、FontSize

### 示例

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

### 8.8.3 FontItalic属性

#### 适用范围

Button、Label、TextBox、RadioButton、CheckBox、ListBox、ComboBox、GroupBox、LED、StatusBar

#### 描述

获取或设置控件字体在运行时是否采用斜体。

#### 使用

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

Form

指定窗体名称或含有窗体名称的字符串变量。

Control

指定控件名称或含有控件名称的字符串。

var

指定要取得属性值的逻辑变量。

value

指定要在属性中设置的逻辑表达式。

#### 数值

False

True

默认: False

#### 另见

Button、Label、TextBox、RadioButton、Checkbox、ListBox、ComboBox、GroupBox、LED、StatusBar、FontName、FontBold、FontSize

#### 示例

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

## 8.8.4 FontName属性

### 适用范围

Button、Label、TextBox、RadioButton、CheckBox、ListBox、ComboBox、GroupBox、LED、StatusBar

### 描述

获取或设置控件所使用文本的字体名称。

### 使用

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

Form

指定窗体名称或含有窗体名称的字符串变量。

Control

指定控件名称或含有控件名称的字符串。

var

指定要取得属性值的字符串变量。

value

指定要在属性中设置的字符串表达式。

### 数值

含有字体名称的字符串。

默认: Microsoft Sans Serif

### 另见

Button、Label、TextBox、RadioButton、Checkbox、ListBox、ComboBox、GroupBox、LED、StatusBar、FontSize、FontItalic、FontBold

### 示例

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

## 8.8.5 FontSize属性

### 适用范围

Button、Label、TextBox、RadioButton、CheckBox、ListBox、ComboBox、GroupBox、LED、StatusBar

### 描述

获取或设置控件所使用文本的字体大小。

### 使用

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

Form

指定窗体名称或含有窗体名称的字符串变量。

Control

指定控件名称或含有控件名称的字符串。

var

指定要取得属性值的实数变量。

value

指定要在属性中设置的实数表达式。

### 数值

字体大小值，单位为点。

默认：8.25

### 另见

Button、Label、TextBox、RadioButton、Checkbox、ListBox、ComboBox、GroupBox、LED、StatusBar、FontName、FontItalic、FontBold

### 示例

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

## 8.8.6 ForeColor属性

### 适用范围

Button、Label、TextBox、RadioButton、CheckBox、ListBox、ComboBox、GroupBox、LED、ProgressBar、StatusBar

### 描述

获取或设置控件的前景颜色。

### 使用

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

Form

指定窗体名称或含有窗体名称的字符串变量。

Control

指定控件名称或含有控件名称的字符串。

var

指定要取得属性值的字符串变量。

value

指定要在属性中设置的字符串表达式。

### 数值

表示控件前景颜色的字符串。

默认: Control Text

### 另见

Button、Label、TextBox、RadioButton、Checkbox、ListBox、ComboBox、GroupBox、LED、ProgressBar、StatusBar

### 示例

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

## 8.8.7 FormBorderStyle属性

### 适用范围

Form

### 描述

获取或设置窗体的边框样式。

### 使用

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

Form

指定窗体名称或含有窗体名称的字符串变量。

var

取得属性值的整数变量。

value

在属性中设定的整数表达式。

### 数值

0 - None

1 - Fixed Single

2 - Fixed3D

3 - FixedDialog

4 - Sizable

默认: FixedDialog

### 另见

Form

### 示例

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

## 8.8.8 FormTemplate 属性

### 适用范围

Form

### 描述

从模板选择、设置窗体的大小。  
希望将窗体的大小适配示教器TP4的屏幕大小时使用。

### 使用

本属性不支持从SPEL+程序进行设置。  
请在对象窗体的属性网格中指定数值进行设置。

### 数值

0 - Standard  
1 - TP4Vertical  
2 - TP4Horizontal  
默认: Standard

### 备注

将值设置为1或2，则以下属性将被固定。

- FormBorderStyle
- Height
- Left
- Width
- Top
- StartPosition 希望任意设置时，请将值设为0。

### 另见

Form、FormBorderStyle、Height、Left、Width

## 8.9 G

### 8.9.1 GClose语句

#### 适用范围

Form

#### 描述

关闭窗体。

#### 使用

```
GClose Form
```

Form

指定窗体名称或含有窗体名称的字符串变量。还可以提供系统窗体ID关闭任一以下系统窗口：

WIN\_IOMON

关闭I/O监视器窗口。

WIN\_TASKMGR

关闭任务管理器窗口。

WIN\_FORCEMON

关闭力监视器窗口。

WIN\_SIMULATOR

关闭仿真器窗口。

#### 备注

请不要对不显示的窗体使用GClose语句。

#### 另见

GShow、GShowDialog

#### 示例

```
GClose frmSetup
```

```
GClose WIN_TASKMGR
```

## 8.9.2 GGet语句

### 描述

GGet 用于获取当前项目中窗体和控件的属性值。

### 使用

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

Form

指定窗体名称或含有窗体名称的字符串变量。

Control

指定控件名称或含有控件名称的字符串。控件必须位于指定窗体内。

Property

指定要取得属性的名称。

var

指定要取得属性值的变量。

### 备注

GGet 用于在运行时检索属性值。

### 另见

GSet

### 8.9.3 GraphicsEnabled属性

#### 适用范围

VideoBox

#### 描述

获取或设置VideoBox 是否显示视觉图形。

#### 使用

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

Form

指定窗体名称或含有窗体名称的字符串变量。

Control

指定控件名称或含有控件名称的字符串。

var

指定要取得属性值的逻辑变量。

value

指定要在属性中设置的逻辑表达式。

#### 数值

False

True

默认: False

#### 另见

VideoBox、VideoEnabled

#### 示例

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

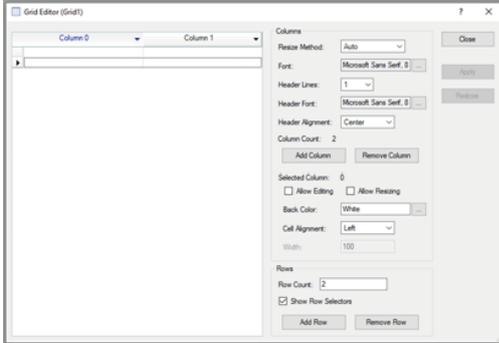
## 8.9.4 GridEditor属性

### 适用范围

Grid

### 描述

打开[GridEditor]对话框，以设置网格控件。



### 使用

GridEditor属性仅能在设计时使用。用户可根据需要设置网格。

设置	描述
Add Column	在网格末尾添加一列。 最小值：1 最大值：25
Add Row	在网格下方添加一行。 最小值：0 最大值：10000
Allow Editing	设置用户是否可以编辑所选列中的单元格。
Allow Resizing	设置用户是否可以修改所选列的大小。
Back Color	设置所选列的背景颜色。
Cell Alignment	指定单元格文本的对齐方式。 靠左：单元格文本与列的左侧对齐。 中心：单元格文本在列内居中对齐。 靠右：单元格文本与列的右侧对齐。
Font	设置列标题的字体。
Header Alignment	设置列标题的对齐方式。 靠左：列标题与列的左侧对齐。 中心：列标题在列内居中对齐。 靠右：列标题与列的右侧对齐。
Header Lines	在列标题中，设置可用的行数。 最小值：1 最大值：3
Remove Column	删除选中的列。
Remove Row	删除选中的行。

设置	描述
Resize Method	添加、删除或调整列的大小时，决定如何调整网格中列的大小。 自动：自动调整列的大小以适合网格控件。 手动：将每列更改为各自的列宽。 同步：调整列的大小时，将所有列也调整为相同大小。
Row Count	设定或显示网格的行数。如需手动修改该值，请在相应的行中输入，然后点击[Apply]按钮。
Show Row Selectors	设置是否在网格左侧显示行选择。
Width	设定或显示所选列的宽度。如需手动修改改值，请勿将Resize Method设置为Auto。

### 备注

使用[GridEditor]对话框，设置网格和初次显示时的方式。用户可以在设计时在单元格中添加文本和数据。可直接拖动列，将其移动到所需的位置。还可以使用列标题中的下拉箭头，将其替换为另一行。

### 另见

Grid

## 8.9.5 GSet语句

### 描述

GSet用于设置GUI Builder中的属性值。

### 使用

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

#### Form

指定窗体名称或含有窗体名称的字符串变量。

#### Control

指定控件名称或含有控件名称的字符串。控件必须位于指定窗体内。

#### Property

指定要设置属性的名称。

#### value

指定要设置属性值的表达式。

### 备注

GSet用于在运行时设置窗体或控件属性的值。

### 另见

GGet

## 8.9.6 GShow语句

### 适用范围

Form

### 描述

GShow 以非模态窗口的形式显示窗体。

### 使用

```
GShow Form
```

Form

指定窗体名称或含有窗体名称的字符串变量。还可以提供系统窗体ID显示任一以下系统窗口：

WIN\_IOMON

打开I/O监视器窗口。

WIN\_TASKMGR

打开任务管理器窗口。

WIN\_FORCEMON

打开力监视器窗口。

WIN\_SIMULATOR

打开仿真器窗口。

### 另见

GClose、GShowDialog

### 示例

```
GShow frmIODiags
```

```
GShow WIN_TASKMGR
```

## 8.9.7 GShowDialog功能

### 适用范围

Form

### 描述

以模态对话框的形式显示窗体，返回DialogResult的值。

### 使用

```
GShowDialog(Form)
```

Form

指定窗体名称或含有窗体名称的字符串变量。

### 返回值

返回窗体的DialogResult属性值。

### 另见

GShow, DialogResult语句

### 示例

```
result = GShowDialog(frmSetup)
If result = DialogResult.OK Then
    Call SaveSettings
EndIf
```

## 8.9.8 GShowDialog语句

### 适用范围

Form

### 描述

以模态对话框的形式显示窗体，不返回DialogResult的值。

### 使用

```
GShowDialog Form
```

Form

指定窗体名称或含有窗体名称的字符串变量。

### 另见

GShow, GShowDialog 功能

### 示例

```
GShowDialog frmInfoDisplay
```

## 8.10 H

### 8.10.1 Height属性

#### 适用范围

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

#### 描述

获取或设置窗体或控件的高度，单位为像素。

#### 使用

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

Form

指定窗体名称或含有窗体名称的字符串变量。

Control

指定控件名称或含有控件名称的字符串。

var

取得属性值的整数变量。

value

在属性中设定的整数表达式。

#### 数值

表示高度的整数，单位为像素。

#### 另见

Form, Button, Label, TextBox, RadioButton, Checkbox, ListBox, ComboBox, PictureBox, GroupBox, VideoBox, LED, ProgressBar, TrackBar, StatusBar, Left, Top, Width, TabControl

#### 示例

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

## 8.10.2 HelpButton属性

### 适用范围

Form

### 描述

获取或设置表示是否应在窗体标题框显示Help按钮的值。

### 使用

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

Form

指定窗体名称或含有窗体名称的字符串变量。

var

指定要取得属性值的逻辑变量。

value

指定要在属性中设置的逻辑表达式。

### 数值

False

True

默认: False

### 备注

该属性设为True时，标题栏[关闭]按钮的左侧将出现一个带问号的小按钮（MaximizeBox、MinimizeBox属性设为“False”时）。可使用此按钮显示应用的帮助信息。

### 另见

Form、HelpID

### 示例

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

## 8.10.3 HelpID属性

### 适用范围

Form

### 描述

返回或设置包含帮助文件内主题HelpID的整数。

### 使用

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

Form

指定窗体名称或含有窗体名称的字符串变量。

var

取得属性值的整数变量。

value

在属性中设定的整数表达式。

### 数值

表示帮助文件中主题ID的整数值。(0~999999)

默认: 0

### 备注

HelpID属性用于显示应用上下文相关的帮助。帮助文件在Epson RC+ [项目]-[属性]-[GUI Builder]页面指定。

当HelpID非0且用户按下F1或单击窗体帮助按钮时(如果HelpButton属性为True), 将显示帮助主题。

### 另见

Form、HelpButton

### 示例

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

## 8.11 |

### 8.11.1 Image属性

#### 适用范围

Button、Label、RadioButton、CheckBox、PictureBox

#### 描述

获取或设置控件上显示的图像。

#### 使用

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

Form

指定窗体名称或含有窗体名称的字符串变量。

Control

指定控件名称或含有控件名称的字符串。

var

指定要取得属性值的字符串变量。

value

指定要在属性中设置的字符串表达式。

#### 数值

含有图像文件完整路径的字符串。

默认：空

#### 另见

Button, Label, RadioButton, Checkbox, PictureBox, ImageAlign , ImageIndex

#### 示例

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

## 8.11.2 ImageAlign属性

### 适用范围

Button、Label、RadioButton、CheckBox、LED

### 描述

获取或设置控件上显示的图像的图像对齐。

### 使用

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

Form

指定窗体名称或含有窗体名称的字符串变量。

Control

指定控件名称或含有控件名称的字符串。

var

取得属性值的整数变量。

value

在属性中设定的整数表达式。

### 数值

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

默认:

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

### 另见

Button、Label、RadioButton、Checkbox、LED、Image、TextAlign

### 示例

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

### 8.11.3 ImageOff属性

#### 适用范围

LED

#### 描述

获取或设置LED控件关闭时显示的图像。

#### 使用

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

Form

指定窗体名称或含有窗体名称的字符串变量。

Control

指定控件名称或含有控件名称的字符串。

var

指定要取得属性值的字符串变量。

value

指定要在属性中设置的字符串表达式。

#### 数值

含有图像文件完整路径的字符串。文件扩展名应为bmp、ico、jpeg、gif或png。

默认: LedOff.ico

#### 另见

LED、ImageOn

#### 示例

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

## 8.11.4 ImageOn属性

### 适用范围

LED

### 描述

获取或设置 LED 控件打开时显示的图像。

### 使用

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

Form

指定窗体名称或含有窗体名称的字符串变量。

Control

指定控件名称或含有控件名称的字符串。

var

指定要取得属性值的字符串变量。

value

指定要在属性中设置的字符串表达式。

### 数值

含有图像文件完整路径的字符串。文件扩展名应为bmp、ico、jpeg、gif或png。

默认: LedRed.ico

### 另见

LED、ImageOff

### 示例

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

## 8.11.5 ImageIndex属性

### 适用范围

PictureBox

### 说明

获取或设置控件上显示的图像的索引编号。

### 使用

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

Form

指定窗体名称或含有窗体名称的字符串变量。

Control

指定控件名称或含有控件名称的字符串。

var

取得属性值的整数变量。

value

在属性中设定的整数表达式。

本属性仅在连接RC800控制器时在GGet、GSet中可以使用。

### 数值

添加到控件的图像文件的索引编号。默认：-1

### 另见

PictureBox, Image

### 示例

```
' 获取当前设置的图像文件的索引编号  
GGet frmMain.pictureBox1.ImageIndex, index  
' 将图像文件的索引编号设置为2。  
GSet frmMain.pictureBox1.ImageIndex, 2
```

## 8.11.6 Interval属性

### 适用范围

Timer

### 描述

获取或设置定时器滴答之间的时间，单位为毫秒。

### 使用

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

Form

指定窗体名称或含有窗体名称的字符串变量。

Control

指定控件名称或含有控件名称的字符串。

var

取得属性值的整数变量。

value

在属性中设定的整数表达式。

### 数值

每次定时器滴答之间的毫秒数。整数值范围为1 - 9999。

默认：100

### 备注

若要获取间隔的秒数，需将此数值除以1,000。

### 另见

Timer、Enabled

### 示例

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

## 8.11.7 IOBit属性

### 适用范围

LED

### 描述

获取或设置在 LED 控件上显示的I/O位数。

### 使用

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

Form

指定窗体名称或含有窗体名称的字符串变量。

Control

指定控件名称或含有控件名称的字符串。

var

取得属性值的整数变量。

value

在属性中设定的整数表达式。

### 数值

整数值范围为0 - 9999。

默认: 0

### 另见

LED、IOType

### 示例

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

## 8.11.8 IOType属性

### 适用范围

LED

### 描述

获取或设置在LED控件上显示的I/O类型。

### 使用

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

Form

指定窗体名称或含有窗体名称的字符串变量。

Control

指定控件名称或含有控件名称的字符串。

var

取得属性值的整数变量。

value

在属性中设定的整数表达式。

### 数值

0 - Input

1 - Output

2 - Memory

默认: Input

### 另见

LED、IOBit

### 示例

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

## 8.12 K

### 8.12.1 KeyPress事件

#### 适用范围

TextBox

#### 描述

按下按键时发生。

#### 使用

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

Sender\$

被设置为引发事件的控件的名称。

Key\$

按下按键的字符串。

#### 备注

KeyPress事件可用于检测操作员输入的按键。

此事件可用作过滤器。通过将Key\$设置为其他字符可更改按下的按键。通过将Key\$设为空字符串可取消keypress响应。

#### 另见

TextBox

#### 示例

```
Function frmMain_txtSpeed_KeyPress(Sender$ As String, ByRef Key$ As String)
    ' 仅允许输入数字
    If Instr("0123456789" Key$) < 0 Then
        Key$ = ""
    EndIf
End
Fend
```

## 8.13 L

### 8.13.1 LargeChange 属性

#### 适用范围

TrackBar

#### 描述

获取或设置单击鼠标或按Page Up和Page Down键时，滑块移动的位置数。

#### 使用

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

Form

指定窗体名称或含有窗体名称的字符串变量。

Control

指定控件名称或含有控件名称的字符串。

var

取得属性值的整数变量。

value

在属性中设定的整数表达式。

#### 数值

一个整数值，显示滑块移动的位置数。

#### 另见

TrackBar, SmallChange

#### 示例

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

## 8.13.2 Left属性

### 适用范围

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

### 描述

获取或设置窗体或控件的左坐标，单位为像素。

### 使用

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

Form

指定窗体名称或含有窗体名称的字符串变量。

Control

指定控件名称或含有控件名称的字符串。

var

取得属性值的整数变量。

value

在属性中设定的整数表达式。

### 数值

表示左坐标的整数，单位为像素。

### 另见

Form, Button, Label, TextBox, RadioButton, Checkbox, ListBox, ComboBox, PictureBox, GroupBox, VideoBox, LED, ProgressBar, TrackBar, StatusBar, Top, Height, Width, TabControl

### 示例

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

### 8.13.3 List属性

#### 适用范围

ListBox、ComboBox

#### 描述

获取或设置控件项目列表中指定元素的字符串值。

#### 使用

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

Form

指定窗体名称或含有窗体名称的字符串变量。

Control

指定控件名称或含有控件名称的字符串。控件必须位于指定窗体内。

index

以整数表达式指定从0开始的列表索引编号。

var

指定要取得属性值的字符串变量。

value

指定要在属性中设置的字符串表达式。

#### 数值

含有指定列表项目的字符串。

#### 备注

使用AddItem为列表框或组合框列表添加项目之后，可以使用List属性获取或编辑任一项目的值。  
设置空字符串可删除项目。

#### 另见

ComboBox、ListBox、ListCount、AddItem

#### 示例

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

## 8.13.4 ListCount属性

### 适用范围

ListBox、ComboBox

### 描述

设置并获取列表框或组合框中的项目计数

### 使用

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

Form

指定窗体名称或含有窗体名称的字符串变量。

Control

指定控件名称或含有控件名称的字符串。控件必须位于指定窗体内。

var

取得属性值的整数变量。

value

指定要在属性中设置的整数表达式。

### 数值

返回列表中的项目数量。

### 备注

将ListCount设置为“0”，可以删除ListBox和ComboBox中的项目。如果将ListCount设置为超过列表中项目数量的值，则会将多出的项目添加到空白文本中。

### 另见

ComboBox、ListBox、AddItem、List

### 示例

```
Integer count  
' 获取列表中的项目数量  
GGet frmMain.lstPartNames.ListCount, count  
  
' 删除列表中的项目  
GSet frmMain.lstPartName.ListCount, 0
```

## 8.13.5 Load事件

### 适用范围

Form

### 描述

首次显示窗体之前发生。

### 使用

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

Sender\$

发生事件的窗体的名称。

### 备注

您可以使用此事件在为用户显示窗体之前执行任务。

### 另见

Form、Resize、Closed

### 示例

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

## 8.14 M

### 8.14.1 MaximizeBox属性

#### 适用范围

Form

#### 描述

获取或设置表示是否在窗体标题栏显示最大化按钮的值。

#### 使用

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

Form

指定窗体名称或含有窗体名称的字符串变量。

var

指定要取得属性值的逻辑变量。

value

指定要在属性中设置的逻辑表达式。

#### 数值

False

True

默认: False

#### 备注

最大化按钮可使用户将窗体最大化为全屏。

#### 另见

Form、MinimizeBox

#### 示例

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

## 8.14.2 Maximum属性

### 适用范围

ProgressBar, TrackBar

### 描述

获取或设置控件处理范围的最大值。

### 使用

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

Form

指定窗体名称或含有窗体名称的字符串变量。

Control

指定控件名称或含有控件名称的字符串。控件必须位于指定窗体内。

var

取得属性值的整数变量。

value

指定要在属性中设置的整数表达式。

### 数值

一个整数值，表示处理范围的最大值。

### 另见

ProgressBar, TrackBar, Minimum, Value

### 示例

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

### 8.14.3 MinimizeBox属性

#### 适用范围

Form

#### 描述

获取或设置表示是否在窗体标题栏显示最小化按钮的值。

#### 使用

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

Form

指定窗体名称或含有窗体名称的字符串变量。

var

指定要取得属性值的逻辑变量。

value

指定要在属性中设置的逻辑表达式。

#### 数值

False

True

默认: False

#### 备注

最小化按钮可使用户将窗体最小化为图标。

#### 另见

Form、MaximizeBox

#### 示例

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

## 8.14.4 Minimum 属性

### 适用范围

ProgressBar, TrackBar

### 描述

获取或设置控件处理范围的最小值。

### 使用

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

Form

指定窗体名称或含有窗体名称的字符串变量。

Control

指定控件名称或含有控件名称的字符串。控件必须位于指定窗体内。

var

取得属性值的整数变量。

value

指定要在属性中设置的整数表达式。

### 数值

一个整数值，表示处理范围的最小值。

### 另见

ProgressBar, TrackBar, Maximum, Value

### 示例

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

## 8.14.5 MultiLine属性

### 适用范围

TextBox

### 描述

获取或设置表示是否为多行文本框控件的值。

### 使用

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

Form

指定窗体名称或含有窗体名称的字符串变量。

Control

指定控件名称或含有控件名称的字符串。控件必须位于指定窗体内。

var

指定要取得属性值的逻辑变量。

value

指定要在属性中设置的逻辑表达式。

### 数值

False

True

默认: False

### 备注

如果想要在文本框控件中显示多行，则必须将MultiLine设为True。

### 另见

TextBox、ScrollBars

### 示例

```
GSet frmMain.txtStatus.MultiLine, True
```

## 8.15 N

### 8.15.1 Name属性

#### 适用范围

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

#### 描述-

在运行时获取窗体或控件的名称。用于在设计时设置窗体或控件的名称。

#### 使用

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

Form

指定窗体名称或含有窗体名称的字符串变量。

Control

指定控件名称或含有控件名称的字符串。

var

指定要取得属性值的字符串变量。

#### 数值

含有窗体或控件名称的字符串。

#### 另见

Form, Button, Label, TextBox, RadioButton, Checkbox, ListBox, ComboBox, PictureBox, GroupBox, Timer, VideoBox, LED, ProgressBar, TrackBar, StatusBar , TabControl

#### 示例

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

## 8.16 0

### 8.16.1 Orientation 属性

#### 适用范围

ProgressBar, TrackBar

#### 描述

获取或设置控件的方向。

#### 使用

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

Form

指定窗体名称或含有窗体名称的字符串变量。

Control

指定控件名称或含有控件名称的字符串。

var

指定要取得属性值的整数变量。

value

指定要在属性中设置的整数表达式。

#### 数值

0 - Horizontal

1 - Vertical

默认: Horizontal

#### 另见

ProgressBar, TrackBar

#### 示例

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

## 8.17 P

### 8.17.1 PasswordChar属性

#### 适用范围

TextBox

#### 描述

获取或设置用于单行文本框控件中密码的掩码字符的字符。

#### 使用

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

Form

指定窗体名称或含有窗体名称的字符串变量。

Control

指定控件名称或含有控件名称的字符串。控件必须位于指定窗体内。

var

指定要取得属性值的字符串变量。

value

指定要在属性中设置的字符串表达式。

#### 数值

含有用于掩码字符的字符串。

#### 备注

使用PasswordChar可隐藏密码的字符。用户输入字符时,将显示PasswordChar。MultiLine属性必须设为False。

#### 另见

TextBox

#### 示例

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

## 8.17.2 PressDelay 属性

### 适用范围

Button

### 描述

单击按钮控件，设置到事件被执行为止的时间。

### 使用

本属性不支持从SPEL+程序进行设置。

请在对象按钮控件的属性网格中指定数值进行设置。

### 数值

以msec(毫秒)指定从单击开始到执行事件为止的时间。设置0~9999范围内的整数。

默认：立即执行事件

### 备注

在未达到设置的时间之前松开单击，事件不会被执行。

### 另见

Button

### 8.17.3 PressSound 属性

#### 适用范围

Button

#### 描述

设置单击按钮控件时是否发出嘟嘟音。

#### 使用

本属性不支持从SPEL+程序进行设置。  
请从对象按钮控件的属性网格进行设置。

#### 数值

False

True

默认: False

#### 另见

Button

## 8.17.4 ProgressBarStyle属性

### 适用范围

ProgressBar

### 描述

获取或设置进度条控件的样式。

### 使用

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

Form

指定窗体名称或含有窗体名称的字符串变量。

Control

指定控件名称或含有控件名称的字符串。

var

指定要取得属性值的整数变量。

value

指定要在属性中设置的整数表达式。

### 数值

1 - Continuous

2 - Marquee

默认: Continuous

### 另见

ProgressBar

### 示例

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

## 8.18 R

### 8.18.1 ReadOnly属性

#### 适用范围

TextBox

#### 描述

获取或设置文本框控件是否为只读。

#### 使用

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

Form

指定窗体名称或含有窗体名称的字符串变量。

Control

指定控件名称或含有控件名称的字符串。控件必须位于指定窗体内。

var

指定要取得属性值的逻辑变量。

value

指定要在属性中设置的逻辑表达式。

#### 数值

False

True

默认: False

#### 备注

该属性设为 True 时，操作员将无法编辑文本框。

#### 另见

TextBox

#### 示例

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

## 8.18.2 RemoveRow 属性

### 适用范围

Grid

### 描述

从网格控件中删除行。

### 使用

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

Form

指定窗体名称或含有窗体名称的字符串变量。

Control

指定控件名称或含有控件名称的字符串。控件必须位于指定窗体内。

value

代表要删除的行的索引的整数值。

### 数值

代表要删除的行索引的整数值。

### 另见

AddRow, Grid, RowCount

### 示例

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

## 8.18.3 Resize事件

### 适用范围

Form

### 描述

重新调整窗体尺寸时发生。

### 使用

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

Sender\$

填入发生事件的窗体的名称。

### 另见

Form、Load、Closed

### 示例

```
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
End
```

## 8.18.4 RobotNumber属性

### 适用范围

StatusBar

### 描述

启用ShowRobot属性时，获取或设置要显示机器人信息的机器人编号。

### 使用

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

Form

指定窗体名称或含有窗体名称的字符串变量。

Control

指定控件名称或含有控件名称的字符串。控件必须位于指定窗体内。

var

取得属性值的整数变量。

value

在属性中设定的整数表达式。

### 数值

一个整数值，用于显示机器人的编号。

默认：1

### 另见

StatusBar, ShowRobot

### 示例

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

## 8.18.5 RowCount 属性

### 适用范围

Grid

### 描述

设置或获取网格控件的行数。

若设置的值大于当前行数，则在末尾添加新的行。

若设置的值小于当前行数，则将从末尾删除行。

### 使用

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

Form

指定窗体名称或含有窗体名称的字符串变量。

Control

指定控件名称或含有控件名称的字符串。控件必须位于指定窗体内。

var

指定包含控件行数的整数变量。

value

指定代表控件中总行数的整数表达式。

### 数值

指定网格行数的整数值。

### 另见

AddRow, Grid, RemoveRow, RowCount

### 示例

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

## 8.19 S

### 8.19.1 Scroll事件

#### 适用范围

TrackBar

#### 描述

滑块移动时，发生的事件。

#### 使用

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

Sender\$

被设置为引发事件的控件的名称。

#### 另见

TrackBar

#### 示例

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

## 8.19.2 ScrollBars属性

### 适用范围

TextBox、Grid

### 描述

获取或设置应出现在多行文本框控件上的滚动条。

### 使用

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

Form

指定窗体名称或含有窗体名称的字符串变量。

Control

指定控件名称或含有控件名称的字符串。控件必须位于指定窗体内。

var

取得属性值的整数变量。

value

在属性中设定的整数表达式。

### 数值

0 - None

1 - Horizontal

2 - Vertical

3 - Both

默认: None

### 备注

若要使用滚动条，MultiLine属性必须设为True。

若要显示水平滚动条，WordWrap属性必须设为False。

### 另见

TextBox、Multiline、WordWrap

### 示例

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

### 8.19.3 SelectedIndex属性

#### 适用范围

ListBox、ComboBox、Grid、TabControl

#### 描述

获取或设置SelectedIndex属性返回当前选定项目的索引。

#### 使用

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

Form

指定窗体名称或含有窗体名称的字符串变量。

Control

指定控件名称或含有控件名称的字符串。控件必须位于指定窗体内。

var

取得属性值的整数变量。

value

在属性中设定的整数表达式。

#### 数值

返回以0开始的选定项目索引。如果未选定项目，则会返回-1。

#### 备注

使用SelectedIndex可确定用户在列表框或组合框中选定的项目。通常，会在控件的Click事件中完成。

#### 另见

AddItem、ComboBox、List、ListBox、Grid

#### 示例

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

## 8.19.4 ShowDateTime 属性

### 适用范围

StatusBar

### 描述

获取或设置是否显示当前时刻。

### 使用

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

Form

指定窗体名称或含有窗体名称的字符串变量。

Control

指定控件名称或含有控件名称的字符串。控件必须位于指定窗体内。

var

指定要取得属性值的逻辑变量。

value

指定要在属性中设置的逻辑表达式。

### 数值

False

True

默认: False

### 另见

StatusBar, ShowEStop, ShowRobot, ShowSafeguard

### 示例

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

## 8.19.5 ShowEStop 属性

### 适用范围

StatusBar

### 描述

获取或设置是否显示Estop状态。

### 使用

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

Form

指定窗体名称或含有窗体名称的字符串变量。

Control

指定控件名称或含有控件名称的字符串。控件必须位于指定窗体内。

var

指定要取得属性值的逻辑变量。

value

指定要在属性中设置的逻辑表达式。

### 数值

False

True

默认: False

### 另见

StatusBar, ShowDateTime, ShowRobot, ShowSafeguard

### 示例

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

## 8.19.6 ShowPrint 属性

### 适用范围

TextBox

### 描述

设置是否获取或设置Print语句的输出。

### 使用

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

Form

指定窗体名称或含有窗体名称的字符串变量。

Control

指定控件名称或含有控件名称的字符串。控件必须位于指定窗体内。

var

指定要取得属性值的逻辑变量。

value

指定要在属性中设置的逻辑表达式。

### 数值

False

True

默认: False

### 备注

如果ShowPrint为True, 则由Spel任务执行的Print语句的所有输出, 将显示在文本框中。如果在多个文本框中将ShowPrint设置为True, 则打印输出将显示在所有文本框中。

### 另见

TextBox、ScrollBars

### 示例

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

## 8.19.7 ShowRobot属性

### 适用范围

StatusBar

### 描述

获取或设置是否显示RobotNumber属性中指定的机器人。

### 使用

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

Form

指定窗体名称或含有窗体名称的字符串变量。

Control

指定控件名称或含有控件名称的字符串。控件必须位于指定窗体内。

var

指定要取得属性值的逻辑变量。

value

指定要在属性中设置的逻辑表达式。

### 数值

False

True

默认: False

### 另见

StatusBar, RobotNumber, ShowDateTime, ShowEStop, ShowSafeguard

### 示例

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

## 8.19.8 ShowSafeguard 属性

### 适用范围

StatusBar

### 描述

获取或设置是否显示安全门状态。

### 使用

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

Form

指定窗体名称或含有窗体名称的字符串变量。

Control

指定控件名称或含有控件名称的字符串。控件必须位于指定窗体内。

var

指定要取得属性值的逻辑变量。

value

指定要在属性中设置的逻辑表达式。

### 数值

False

True

默认: False

### 另见

StatusBar, ShowDateTime, ShowEStop, ShowRobot

### 示例

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

## 8.19.9 SizeMode属性

### 适用范围

PictureBox

### 描述

表示图像显示的方式。

### 使用

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

Form

指定窗体名称或含有窗体名称的字符串变量。

Control

指定控件名称或含有控件名称的字符串。控件必须位于指定窗体内。

var

取得属性值的整数变量。

value

在属性中设定的整数表达式。

### 数值

0 - Normal

1 - StretchImage

2 - AutoSize

3 - CenterImage

4 - Zoom

默认: Normal

### 备注

默认情况下为Normal模式，图像置于图片框的左上角，并会剪掉图像对于图片框过大的所有部分。

使用StretchImage值可拉伸图像，以适合图片框。

使用AutoSize值可调整控件尺寸，使其始终适合图像。

使用CenterImage值可使图像居中于图片框。

使用Zoom值可拉伸图像，使其适合保持相同屏幕比例的图片框。

### 另见

PictureBox、Image、ImageAlign

### 示例

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

## 8.19.10 SmallChange属性

### 适用范围

TrackBar

### 描述

获取或设置使用键盘(方向键)输入时, 滑块移动的位置数。

### 使用

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

Form

指定窗体名称或含有窗体名称的字符串变量。

Control

指定控件名称或含有控件名称的字符串。

var

取得属性值的整数变量。

value

在属性中设定的整数表达式。

### 数值

一个整数值, 用于表示滑块移动的位置数。

### 另见

TrackBar, LargeChange

### 示例

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

## 8.19.11 Sorted属性

### 适用范围

ComboBox、ListBox

### 描述

获取或设置表示是否按字母顺序对组合框或列表框中的项目排序的值。

排序顺序：数字字符（不论是单字节或双字节）、字母符号、片假名、平假名、日语字符

### 使用

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

Form

指定窗体名称或含有窗体名称的字符串变量。

Control

指定控件名称或含有控件名称的字符串。控件必须位于指定窗体内。

var

指定要取得属性值的逻辑变量。

value

指定要在属性中设置的逻辑表达式。

### 数值

False

True

默认：False

### 另见

ComboBox、ListBox

### 示例

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

## 8.19.12 StartPosition属性

### 适用范围

Form

### 描述

获取或设置窗体运行时的启动位置。

### 使用

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

Form

指定窗体名称或含有窗体名称的字符串变量。

var

取得属性值的整数变量。

value

在属性中设定的整数表达式。

### 数值

0 - Manual

1 - CenterScreen

2 - CenterParent

默认: CenterScreen

### 另见

Form、WindowState

### 示例

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

## 8. 20 T

### 8. 20. 1 TabAlignment属性

#### 适用范围

TabControl

#### 说明

获取或设置控件上显示的选项卡的布局。

#### 使用

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

Form

指定窗体名称或含有窗体名称的字符串变量。

Control

指定控件名称或含有控件名称的字符串。

var

取得属性值的整数变量。

value

在属性中设定的整数表达式。

本属性仅在连接RC800控制器时在GGet、GSet中可以使用。

#### 值

0 - Left

1 - Top

2 - Right

3 - Bottom

默认: TOP

#### 另见

TabControl

#### 示例

```
' 获取当前设置的选项卡的布局  
GGet frmMain.tabControl1.TabAlignment, alignment  
' 将选项卡的布局设置为0 (Left)。  
GSet frmMain.tabControl1.TabAlignment, 0
```

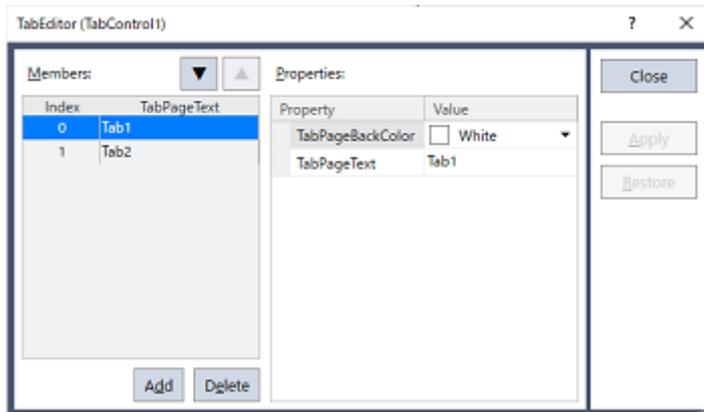
## 8.20.2 TabEditor属性

### 适用范围

TabControl

### 说明

打开选项卡编辑器，以设置选项卡控件。



### 使用

TabEditor属性仅能在设计时使用。可以进行选项卡页面的添加和删除，设置选项卡页面的背景颜色、选项卡显示文字。

设置	说明
添加按钮	在末尾添加选项卡页面。
删除按钮	删除选中的选项卡页面。
TabPageBackColor	设置选中的选项卡页面背景颜色。
TabText	设置选中的选项卡页面的选项卡显示文字。

### 数值

0 - Left  
 1 - Top  
 2 - Right  
 3 - Bottom  
 默认: TOP

### 另见

TabControl

### 8.20.3 TabIndex属性

#### 适用范围

Button、Label、TextBox、RadioButton、CheckBox、ListBox、ComboBox、PictureBox、GroupBox、VideoBox、LED、ProgressBar、TrackBar、StatusBar、Grid

#### 描述

获取或设置容器内控件的选项卡顺序。

#### 使用

**GGet** Form.Control.TabPageBackColor (Index), var

**GSet** Form.Control.TabPageBackColor (Index), value

Form

指定窗体名称或含有窗体名称的字符串变量。

Control

指定控件名称或含有控件名称的字符串。

var

取得属性值的整数变量。

value

在属性中设定的整数表达式。

#### 数值

控件选项卡顺序的索引。

#### 另见

Button、Label、TextBox、RadioButton、Checkbox、ListBox、ComboBox、PictureBox、GroupBox、VideoBox、LED、ProgressBar、TrackBar、StatusBar

#### 示例

```
GSet frmMain.txtStatus.TabIndex, 3
```

## 8. 20. 4 TabPageBackColor属性

### 适用范围

TabControl

### 说明

设置要选项卡控件上显示的选项卡页面的背景颜色。

### 使用

**GGet** Form.Control.TabPageBackColor (index), var

**GSet** Form.Control.TabPageBackColor (index), value

Form

指定窗体名称或含有窗体名称的字符串变量。

Control

指定控件名称或含有控件名称的字符串。

Index

指定选项卡控件的索引编号。

var

指定要取得属性值的字符串变量。

value

指定要在属性中设置的字符串表达式。

本属性仅在连接RC800控制器时在GGet、GSet中可以使用。

### 数值

在选项卡页面背景颜色中指定颜色的名称。

默认: Control

### 另见

TabControl, TabEditor

### 示例

· 获取当前设置的索引编号2选项卡页面的背景颜色

```
GGet frmMain.tabControl1.TabPageBackColor(2), backColor
```

· 将索引编号0选项卡页面的背景颜色设置为“Red”。

```
GSet frmMain.tabControl1.TabPageBackColor(0), "Red"
```

## 8.20.5 TabPageFont 属性

此属性仅在设计GUI上可用。

### 适用范围

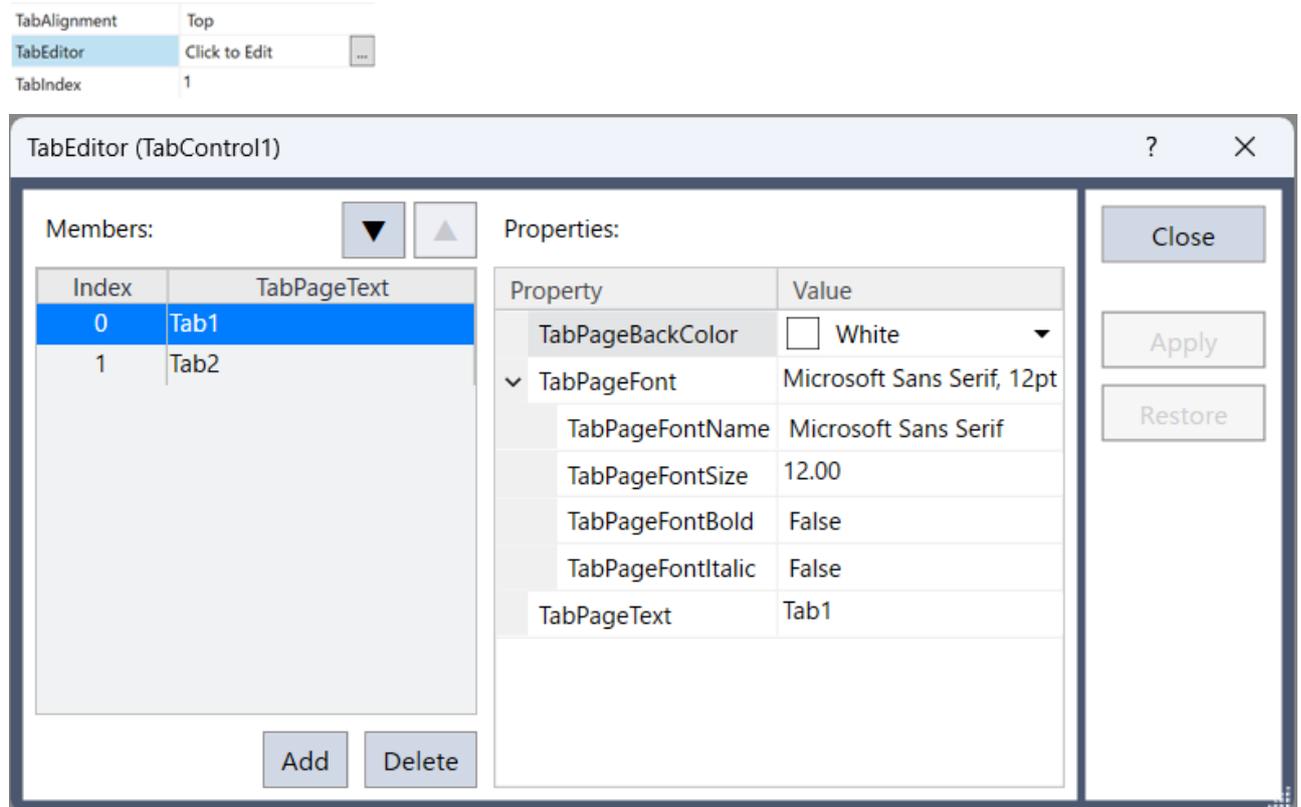
TabControl

### 说明

设计窗体时，可变更TabControl选项卡页面的文本字体。（窗体名、类型、大小）

### 使用

按下GUI Builder属性网格中显示的TabEditor按钮，显示选项卡编辑器。可以通过利用选项卡编辑器更改TabPageFont属性的值，更改显示字符的字体。



更改字体设置并保存项目后，设置将被保存。

### 另见

TabControl

## 8.20.6 TabFontBold 属性

### 适用范围

TabControl

### 说明

设计窗体时，设置或获取TabControl选项卡页面的文本字体为“标准”或“粗体”。

### 使用

**GGet** Form.Control.TabPageFontBold(index), var

**GSet** Form.Control.TabPageFontBold(index), value

Form

指定窗体名称或含有窗体名称的字符串变量。

Control

指定控件名称或含有控件名称的字符串。

Index

指定选项卡控件的索引编号。

var

指定要取得属性值的字符串变量。

value

指定要在属性中设置的字符串表达式。

本属性仅在连接RC800控制器时在GGet、GSet中可以使用。

**数值** 0 - False

1 - True

默认: 0 - False

### 另见

TabControl, TabPageFontName, TabPageFontItalic, TabPageFontSize

### 示例

```
' 获取当前设置的索引编号2选项卡页面的文本字体类型（标准、粗体）
GGet frmMain.tabControl1.TabPageFontBold(2), fontBold
' 将索引编号0选项卡页面的文本字体类型设为粗体
GSet frmMain.tabControl1.TabPageFontBold(0), True
```

## 8.20.7 TabPageFontItalic 属性

### 适用范围

TabControl

### 描述

设计窗体时，设置或获取TabControl选项卡页面的文本字体为“标准”或“斜体”。

### 使用

**GGet** Form.Control.TabPageFontItalic(index), var

**GSet** Form.Control.TabPageFontItalic(index), value

Form

指定窗体名称或含有窗体名称的字符串变量。

Control

指定控件名称或含有控件名称的字符串。

Index

指定选项卡控件的索引编号。

var

指定要取得属性值的逻辑变量。

value

指定要在属性中设置的逻辑表达式。

### 数值

0 - False 1 - True

默认: 0 - False

### 另见

TabControl, TabPageFontName, TabPageFontBold, TabPageFontSize

### 示例

```
' 获取当前设置的索引编号2选项卡页面的文本字体类型（标准、斜体）
GGet frmMain.tabControl1.TabPageFontItalic(2), fontItalic
' 将索引编号0选项卡页面的文本字体类型设为斜体
GSet frmMain.tabControl1.TabPageFontItalic(0), True
```

## 8.20.8 TabPageFontName 属性

### 适用范围

TabControl

### 说明

设置或获取TabControl选项卡页面的文本字体名称。

### 使用

**GGet** Form.Control.TabPageFontName(index), var

**GSet** Form.Control.TabPageFontName(index), value

Form

指定窗体名称或含有窗体名称的字符串变量。

Control

指定控件名称或含有控件名称的字符串。

Index

指定选项卡控件的索引编号。

var

指定要取得属性值的字符串变量。

value

指定要在属性中设置的字符串表达式。

### 数值

含有字体名称的字符串。

默认: Microsoft Sans Serif

### 另见

TabControl, TabPageFontSize, TabPageFontItalic, TabPageFontBold

### 示例

' 获取当前设置的索引编号2选项卡页面的文本字体名称

```
GGet frmMain.tabControl1.TabPageFontName(2), fontName
```

' 将索引编号0选项卡页面的文本字体名称设为"Courier New"

```
GSet frmMain.tabControl1.TabPageFontName(0), "Courier New"
```

## 8.20.9 TabFontSize 属性

### 适用范围

TabControl

### 说明

设置或获取TabControl选项卡页面的文本字体大小。

### 使用

**GGet** Form.Control.TabPageFontSize(index), var

**GSet** Form.Control.TabPageFontSize(index), value

Form

指定窗体名称或含有窗体名称的字符串变量。

Control

指定控件名称或含有控件名称的字符串。

var

指定要取得属性值的实数变量。

value

指定要在属性中设置的实数表达式。

本属性仅在连接RC800控制器时在GGet、GSet中可以使用。

### 数值

字体大小值，单位为点。

默认：12

### 另见

TabControl, TabPageFontName, TabPageFontItalic, TabPageFontBold

### 示例

```
' 现获取当前设置的索引编号2选项卡页面的文本字体大小
GGet frmMain.tabControl1.TabPageFontSize(2), fontSize
' 将索引编号0选项卡页面的文本字体大小设为20
GSet frmMain.tabControl1.TabPageFontSize(0), 20
```

## 8.20.10 TabPageText 属性

### 适用范围

TabControl

### 说明

设置要在选项卡控件上显示的选项卡的文本。

### 使用

**GGet** Form.Control.TabPageText(Index), var

**GSet** Form.Control.TabPageText(Index), value

Form

指定窗体名称或含有窗体名称的字符串变量。

Control

指定控件名称或含有控件名称的字符串。

Index

指定选项卡控件的索引编号。

var

指定要取得属性值的字符串变量。

value

指定要在属性中设置的字符串表达式。

本属性仅在连接RC800控制器时在GGet、GSet中可以使用。

### 数值

要在选项卡中指定的文本。

默认: TabXX

### 另见

TabControl, TabEditor

### 示例

’获取当前设置的索引编号2选项卡的文本

GGet frmMain.tabControl1.TabPageText(2), backColor ’将索引编号0选项卡的文本设置为“监视器”。

GSet frmMain.tabControl1.TabPageText(0), “监视器”

## 8.20.11 Text属性

### 适用范围

Form、Button、Label、TextBox、RadioButton、CheckBox、ComboBox、GroupBox、LED、StatusBar

### 描述

获取或设置与控件相关的文本。

### 使用

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

Form

指定窗体名称或含有窗体名称的字符串变量。

Control

指定控件名称或含有控件名称的字符串。

var

指定要取得属性值的字符串变量

value

指定要在属性中设置的字符串表达式。

### 数值

含有窗体或控件文本的字符串。

### 另见

Form、Button、Label、TextBox、RadioButton、Checkbox、ComboBox、GroupBox、LED、StatusBar、TextAlign

### 示例

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

## 8.20.12 TextAlign属性

### 适用范围

Button、Label、TextBox、RadioButton、CheckBox、LED

### 描述

获取或设置控件中的文本对齐。

### 使用

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

Form

指定窗体名称或含有窗体名称的字符串变量。

Control

指定控件名称或含有控件名称的字符串。

var

取得属性值的整数变量。

value

在属性中设定的整数表达式。

### 数值

- 对于按钮、标签、单选按钮、复选框控件
  - 1 - TopLeft
  - 2 - TopCenter
  - 3 - TopRight
  - 4 - MiddleLeft
  - 5 - MiddleCenter
  - 6 - MiddleRight
  - 7 - BottomLeft
  - 8 - BottomCenter
  - 9 - BottomRight

默认: 1 - TopLeft (Label 1)  
4 - MiddleLeft (RadioButton, CheckBox)  
5 - MiddleCenter (Button)  
6 - MiddleRight (LED)
- 文本框控件
  - 1 - Left
  - 2 - Center
  - 3 - Right

默认: Left

### 另见

Button、Label、TextBox、RadioButton、Checkbox、LED、Text

### 示例

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

## 8.20.13 Tick事件

### 适用范围

Timer

### 描述

定时器控件达到间隔值时发生。

### 使用

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

Sender\$

发生事件的控件的名称。

### 另见

Timer、Interval、Enabled

### 示例

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

## 8. 20. 14 TickFrequency属性

### 适用范围

TrackBar

### 描述

获取或设置刻度线之间的位置数。

### 使用

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

Form

指定窗体名称或含有窗体名称的字符串变量。

Control

指定控件名称或含有控件名称的字符串。

var

取得属性值的整数变量。

value

在属性中设定的整数表达式。

### 数值

一个整数值，代表刻度线之间的位置数。

### 另见

TrackBar

### 示例

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

## 8.20.15 TickStyle属性

### 适用范围

TrackBar

### 描述

获取或设置刻度线在跟踪栏上显示的位置。

### 使用

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

Form

指定窗体名称或含有窗体名称的字符串变量。

Control

指定控件名称或含有控件名称的字符串。

var

取得属性值的整数变量。

value

在属性中设定的整数表达式。

### 数值

0 - None

1 - TopLeft

2 - BottomRight

3 - Both

默认: BottomRight

### 另见

TrackBar

### 示例

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

## 8.20.16 ToolTipText属性

### 适用范围

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

### 描述

用于在用户将鼠标指针放在控件上方时显示具有控件用途简介的小方形弹出窗口。

### 使用

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

#### Form

指定窗体名称或含有窗体名称的字符串变量。

#### Control

指定控件名称或含有控件名称的字符串。

#### var

指定要取得属性值的字符串变量。

#### value

指定要在属性中设置的字符串表达式。

### 数值

含有控件工具提示文本的字符串。

默认：空

### 另见

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

### 示例

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

## 8. 20. 17 Top属性

### 适用范围

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

### 描述

获取或设置窗体或控件的顶坐标，单位为像素。

### 使用

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

Form

指定窗体名称或含有窗体名称的字符串变量。

Control

指定控件名称或含有控件名称的字符串。

var

取得属性值的整数变量。

value

在属性中设定的整数表达式。

### 数值

表示顶坐标的整数，单位为像素。

### 另见

Form, Button, Label, TextBox, RadioButton, Checkbox, ListBox, ComboBox, PictureBox, GroupBox, VideoBox, LED, ProgressBar, TrackBar, StatusBar, Left, Height, Width , TabControl

### 示例

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

## 8.20.18 Type属性

### 适用范围

All Controls

### 描述

获取控件的类型名称。

### 使用

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

Form

指定窗体名称或含有窗体名称的字符串变量。

Control

指定控件名称或含有控件名称的字符串。

var

指定要取得属性值的字符串变量。

### 数值

含有控件类型名称的字符串。

### 另见

Controls

### 示例

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

## 8.21 U

### 8.21.1 Update属性

#### 适用范围

TextBox

#### 描述

获取或设置当文本框设为显示全局变量时是否自动更新文本框的值。

#### 使用

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

Form

指定窗体名称或含有窗体名称的字符串变量。

Control

指定控件名称或含有控件名称的字符串。

var

指定要取得属性值的逻辑变量。

value

指定要在属性中设置的逻辑。

#### 数值

False

True

默认: True

#### 备注

若要在文本框上显示SPEL+ 全局变量(数组除外), 设置Variable属性中的全局变量名称。如果Update属性设为False, 则不会更新设置的全局变量的值。

若将TextBox Variable属性设置为全局变量, 在默认设置下执行RC+时, TextBox Text值会自动更新为变量值。Update属性的默认值为True。如需使用用户可更改相同的变量值, 需将Update设置为False。这将阻止自动更新并允许用户输入新的值。

当Update设置从False更改为True时, RC+将变量值设置为用户在文本框中输入的新值。其中一种方法是, “更改值”或在有类似名称的文本框上方添加复选框。在复选框的Click事件中, 使用GGet获取复选框的选中状态, 然后用GSet获取TextBox Update属性。如果选中该复选框, 请将Update设置为False。

#### 另见

TextBox、Variable

#### 示例

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

## 8.22 V

### 8.22.1 Value属性

#### 适用范围

ProgressBar, TrackBar

#### 描述

获取或设置显示控件当前位置的值。

#### 使用

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

Form

指定窗体名称或含有窗体名称的字符串变量。

Control

指定控件名称或含有控件名称的字符串。控件必须位于指定窗体内。

var

指定要取得属性值的整数变量。

value

指定要在属性中设置的整数表达式。

#### 数值

一个整数值，显示当前位置的值。

#### 另见

ProgressBar, TrackBar, Maximum, Minimum, Variable

#### 示例

```
GSet form1.ProgressBar1.Value, 75
```

## 8.22.2 Variable属性

### 适用范围

TextBox、ProgressBar、TrackBar

### 描述

获取或设置用于文本框显示的SPEL+ 全局变量(数组除外)名称。

### 使用

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

#### Form

指定窗体名称或含有窗体名称的字符串变量。

#### Control

指定控件名称或含有控件名称的字符串。控件必须位于指定窗体内。

#### var

指定要取得属性值的字符串变量。

#### value

指定要在属性中设置的字符串表达式。

### 数值

当前项目中SPEL+ 全局变量(数组除外)的名称。

默认: None

### 备注

当Variable属性设为SPEL+ 全局变量(数组除外)时, 将在运行时自动更新Text值, 以显示变量值。

若将TextBox Variable属性设置为全局变量, 在默认设置下执行RC+时, TextBox Text值会自动更新为变量值。Update属性的默认值为True。如需使用用户可更改相同的变量值, 需将Update设置为False。

当Update设置从False更改为True时, RC+将变量值设置为用户在文本框中输入的新值。

其中一种方法是, “更改值” 或在有类似名称的文本框上方添加复选框。在复选框的Click事件中, 使用GGet获取复选框的选中状态, 然后用GSet获取TextBox Update属性。如果选中该复选框, 请将Update设置为False。若将ProgressBar Variable属性设置为全局变量, 进度条将显示变量值。若将TrackBar Variable属性设置为全局变量, 则变量值由轨迹栏设置。

### 另见

TextBox、ProgressBar、TrackBar、Value

### 示例

```
GSet frmMain.txtStatus.Variable, "g_Status$"
```

### 8.22.3 VideoEnabled属性

#### 适用范围

VideoBox

#### 描述

获取或设置VideoBox是否显示视频。

#### 使用

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

Form

指定窗体名称或含有窗体名称的字符串变量。

Control

指定控件名称或含有控件名称的字符串。

var

指定要取得属性值的逻辑变量。

value

指定要在属性中设置的逻辑表达式。

#### 数值

False

True

默认: False

#### 另见

VideoBox、GraphicsEnabled

#### 示例

```
GSet frmMain.VideoBox1.VideoEnabled, True
```

## 8.22.4 Visible属性

### 适用范围

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

### 描述

获取或设置表示是否在运行时显示控件的值。

### 使用

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

#### Form

指定窗体名称或含有窗体名称的字符串变量。

#### Control

指定控件名称或含有控件名称的字符串。

#### var

指定要取得属性值的逻辑变量。

#### value

指定要在属性中设置的逻辑表达式。

### 数值

True 显示控件。

False 不显示控件。

默认: True

### 备注

如果不在使用GShow或GShowDialog加载表单的情况下执行GGet, 则将返回False。

### 另见

Button, Label, TextBox, RadioButton, Checkbox, ListBox, ComboBox, PictureBox, GroupBox, VideoBox, LED, ProgressBar, StatusBar, TrackBar, Enabled, TabControl

### 示例

```
GSet frmMain.txtStatus.Visible, True
```

## 8.23 W

### 8.23.1 Width属性

#### 适用范围

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

#### 描述

获取或设置窗体或控件的宽度，单位为像素。

#### 使用

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

Form

指定窗体名称或含有窗体名称的字符串变量。

Control

指定控件名称或含有控件名称的字符串。

var

取得属性值的整数变量。

value

在属性中设定的整数表达式。

#### 数值

表示宽度的整数，单位为像素。

#### 另见

Form, Button, Label, TextBox, RadioButton, Checkbox, ListBox, ComboBox, PictureBox, GroupBox, VideoBox, LED, ProgressBar, TrackBar, StatusBar, Left, Top, Height, TabControl

#### 示例

```
GSet frmMain.txtStatus.Width, 300
```

## 8.23.2 WindowState属性

### 适用范围

Form

### 描述

获取或设置窗体的窗口状态。

### 使用

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

Form

指定窗体名称或含有窗体名称的字符串变量。

var

取得属性值的整数变量。

value

在属性中设定的整数表达式。

### 数值

0 - Normal

1 - Minimized

2 - Maximized

默认: Normal

### 另见

Form

### 示例

```
GSet frmMain.WindowState, WINDOWSTATE_MAXIMIZED
```

### 8.23.3 WordWrap属性

#### 适用范围

TextBox

#### 描述

表示多行文本框控件是否在必要时自动换行至下一行的开始。

#### 使用

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

Form

指定窗体名称或含有窗体名称的字符串变量。

Control

指定控件名称或含有控件名称的字符串。控件必须位于指定窗体内。

var

指定要取得属性值的逻辑变量。

value

指定要在属性中设置的逻辑表达式。

#### 数值

True : 多行文本框时, 对一行放不下的单次进行换行。

False : 当用户的输入超过文本框右端时, 文本框控件自动水平滚动。

默认: True

#### 另见

TextBox、Multiline、ScrollBars

#### 示例

```
GSet frmMain.txtStatus.WordWrap, True
```