

EPSON RC+ 7.0 选件

*GUI Builder 7.0*

Rev.8

SCM231S5541F

翻译版

EPSON RC+ 7.0 选件 GUI Builder 7.0 Rev.8

EPSON RC+ 7.0选件

# *GUI Builder 7.0*

Rev.8

©Seiko Epson Corporation 2013-2023

## 前言

感谢您购买我公司的机器人产品。本手册包含正确使用 EPSON RC+ 7.0 选件 GUI Builder 7.0 的须知信息。

使用前，请仔细阅读本手册及其他相关手册。

请将本手册放在随手可及之处，以便随时查看。

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。但是如果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。

## 商标

Microsoft、Windows、Windows 商标为美国 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。

## 本手册中的商标符号

Microsoft® Windows® 8 operating system

Microsoft® Windows® 10 operating system

Microsoft® Windows® 11 operating system

在本手册中，Windows 8、Windows 10 和 Windows 11 指的是上述各操作系统。在某些情况下，Windows 一般是指 Windows 8、Windows 10 和 Windows 11。

## 注意事项

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。

## 制造商

**SEIKO EPSON CORPORATION**

## 联系方式

有关咨询处的详细内容，请参阅下记手册序言中的“销售商”。

机器人系统 安全手册 请首先阅读本手册

<b>1. 简介</b>	<b>1</b>
1.1 功能 .....	1
<b>2. 安装</b>	<b>1</b>
<b>3. 操作入门</b>	<b>2</b>
3.1 GUI Builder教程 .....	2
<b>4. GUI Builder环境</b>	<b>7</b>
4.1 概述 .....	7
4.2 了解GUI Builder所需的基本概念 .....	7
4.3 打开GUI Builder窗口 .....	8
4.4 GUI Builder窗口的组成部分 .....	8
4.4.1 设计区域 .....	8
4.4.2 工具栏 .....	9
4.4.3 窗口资源管理器 .....	10
4.4.4 属性网格 .....	10
4.4.5 事件网格 .....	10
4.5 窗体和控件的使用 .....	10
4.5.1 创建窗体 .....	10
4.5.2 删除窗体 .....	10
4.5.3 打开和关闭窗口 .....	11
4.5.4 更改窗体尺寸 .....	11
4.5.5 编辑多个窗体 .....	11
4.5.6 创建控件 .....	11
4.5.7 删除控件 .....	11
4.5.8 缩放和移动控件 .....	12
4.5.9 控件的复制、剪切和粘贴 .....	12
4.5.10 编辑属性 .....	12
4.5.11 事件处理程序的使用 .....	13
4.5.12 更改选项卡顺序 .....	14
4.5.13 保存工作 .....	14
4.6 设置GUI Builder选项 .....	15
4.7 导入其他项目的窗体 .....	16
<b>5. GUI Builder组件</b>	<b>17</b>
5.1 窗体 .....	17
5.1.1 描述 .....	17
5.1.2 使用 .....	17
5.1.3 窗体属性 .....	19
5.1.4 窗体事件 .....	19
5.2 按钮控件 .....	20
5.2.1 描述 .....	20
5.2.2 使用 .....	20
5.2.3 按钮控件属性 .....	20
5.2.4 按钮控件事件 .....	21
5.3 标签控件 .....	22

5.3.1 描述 .....	22
5.3.2 使用 .....	22
5.3.3 标签控件属性 .....	22
5.3.4 标签控件事件 .....	23
5.4 文本框控件 .....	24
5.4.1 描述 .....	24
5.4.2 使用 .....	24
5.4.3 文本框控件属性 .....	24
5.4.4 文本框控件事件 .....	25
5.5 单选按钮控件 .....	26
5.5.1 描述 .....	26
5.5.2 使用 .....	26
5.5.3 单选按钮控件属性 .....	27
5.5.4 单选按钮控件事件 .....	28
5.6 复选框控件 .....	28
5.6.1 描述 .....	28
5.6.2 使用 .....	28
5.6.3 复选框控件属性 .....	28
5.6.4 复选框控件事件 .....	29
5.7 列表框控件 .....	30
5.7.1 描述 .....	30
5.7.2 使用 .....	30
5.7.3 列表框控件属性 .....	31
5.7.4 列表框控件事件 .....	32
5.8 组合框控件 .....	32
5.8.1 描述 .....	32
5.8.2 使用 .....	32
5.8.3 组合框控制属性 .....	33
5.8.4 组合框控件事件 .....	33
5.9 图片框控件 .....	34
5.9.1 描述 .....	34
5.9.2 使用 .....	34
5.9.3 图片框控件属性 .....	34
5.9.4 图片框控件事件 .....	35
5.10 分组框控件 .....	36
5.10.1 描述 .....	36
5.10.2 使用 .....	36
5.10.3 分组框控件属性 .....	36
5.10.4 分组框控件事件 .....	37
5.11 定时器控件 .....	38
5.11.1 描述 .....	38
5.11.2 使用 .....	38
5.11.3 定时器控件属性 .....	38
5.11.4 定时器控件事件 .....	38
5.12 视频框控件 .....	39
5.12.1 描述 .....	39
5.12.2 使用 .....	39
5.12.3 视频框控件属性 .....	39

5.12.4 视频框控件事件	40
5.13 LED控件	40
5.13.1 描述	40
5.13.2 使用	40
5.13.3 LED控件属性	40
5.13.4 LED控件事件	41
5.14 状态栏控件	42
5.14.1 描述	42
5.14.2 使用	42
5.14.3 状态栏控件属性	42
5.14.4 状态栏控件事件	43
5.15 进度条控件	44
5.15.1 描述	44
5.15.2 使用	44
5.15.3 进度条控件属性	44
5.15.4 进度条控件事件	45
5.16 跟踪栏控件	46
5.16.1 描述	46
5.16.2 使用	46
5.16.3 跟踪栏控件属性	46
5.16.4 跟踪栏控件事件	47
5.17 网格控件	48
5.17.1 描述	48
5.17.2 使用	48
5.17.3 网格控件属性	48
5.17.4 网格控件事件	49
5.17.5 网格编辑器	49
<b>6. 操作</b>	<b>50</b>
6.1 概述	50
6.2 编程模式下的GUI开发	50
6.2.1 设计GUI	50
6.2.2 调试	51
6.3 自动模式	51
6.4. 处理暂停与继续	52
6.5. 处理紧急停止	52
6.6 使用帮助文件	52
<b>7. GUI Builder参考</b>	<b>53</b>
7.1 概述	53
7.2 GUI Builder属性和事件格式描述	53
AcceptButton属性	54
AddItem属性	55
AddRow 属性	56
AppendText属性	57
AllowStateChange属性	58
BackColor属性	59
BackColorMode属性	60

BorderStyle属性	61
Camera属性	62
CancelButton属性	63
CellBackColor属性	64
CellChanged事件	65
CellForeColor属性	66
CellText属性	67
Checked属性	68
Click事件	69
Closed事件	70
ControlBox属性	71
Controls属性	72
Count属性	73
DbClick事件	74
DialogResult属性	75
Dock 属性	76
DropDownStyle属性	77
Enabled属性	78
EventTaskType属性	79
Font属性	80
FontBold属性	81
FontItalic属性	82
FontName属性	83
FontSize属性	84
ForeColor属性	85
FormBorderStyle属性	86
GClose语句	87
GGet语句	88
GraphicsEnabled属性	89
GridEditor属性	90
GSet语句	92
GShow语句	93
GShowDialog功能	94
GShowDialog语句	95
Height属性	96
HelpButton属性	97
HelpID属性	98
Image属性	99
ImageAlign属性	100
ImageOff属性	101
ImageOn属性	102
Interval属性	103
IOBit属性	104
IOType属性	105
KeyPress事件	106
LargeChange 属性	107
Left属性	108
List属性	109



ListCount属性 .....	110
Load事件 .....	111
MaximizeBox属性 .....	112
Maximum属性 .....	113
MinimizeBox属性 .....	114
Minimum 属性 .....	115
MultiLine属性 .....	116
Name属性 .....	117
Orientation 属性 .....	118
PasswordChar属性 .....	119
ProgressBarStyle属性 .....	120
ReadOnly属性 .....	121
RemoveRow 属性 .....	122
Resize事件 .....	123
RobotNumbe属性 .....	124
RowCount 属性 .....	125
Scroll事件 .....	126
ScrollBars属性 .....	127
SelectedIndex属性 .....	128
ShowDateTime 属性 .....	129
ShowEStop 属性 .....	130
ShowPrint 属性 .....	131
ShowRobot属性 .....	132
ShowSafeguard 属性 .....	133
SizeMode属性 .....	134
SmallChange属性 .....	135
Sorted属性 .....	136
StartPosition属性 .....	137
Step属性 .....	138
TabIndex属性 .....	139
Text属性 .....	140
TextAlign属性 .....	141
Tick事件 .....	142
TickFrequency属性 .....	143
TickStyle属性 .....	144
ToolTipText属性 .....	145
Top属性 .....	146
Type属性 .....	147
Update属性 .....	148
Value属性 .....	149
Variable属性 .....	150
VideoEnabled属性 .....	151
Visible属性 .....	152
Width属性 .....	153
WindowState属性 .....	154
WordWrap属性 .....	155



## 1. 简介

EPSON RC+ GUI Builder 7.0 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。



对于高级的 GUI 应用，您可能会想要考虑同时使用 EPSON RC+ 7.0 RC+ API 选件和 Visual Studio 或能够与 VB Guide .NET 库连接的其他工具。

### 1.1 功能

GUI Builder 7.0 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。

## 2. 安装

请遵守本章说明，以助于确保正确安装 GUI Builder 7.0 软件。

启动前，确保已关闭所有 Windows 应用。

安装 GUI Builder：

1. 安装 EPSON RC+ 7.0。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 7.0 的软件密钥。有关激活控制器选件的方式信息，请参阅 EPSON RC+ 7.0 用户指南。

这样便完成了 GUI Builder 7.0 的安装。

## 3. 操作入门

本章介绍了有关 GUI Builder 7.0 操作入门的信息。

继续前，确保已激活 GUI Builder 选件。有关详细信息，请参阅 安装 章节。

如果您之前从未使用过 EPSON RC+，则应阅读 EPSON RC+ 7.0 用户指南，以熟悉项目和程序的创建。

下一节将展示一些简单概念的说明教程。

### 3.1 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：

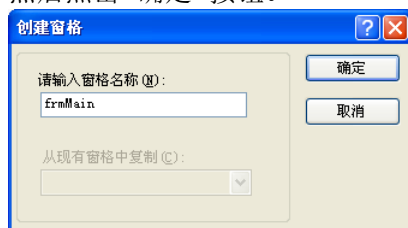
- 创建具有运行机器人循环功能的新 EPSON RC+ 项目。
- 创建具有<开始>和<停止>按钮的窗体，以运行机器人循环。这将展示如何创建窗体及添加按钮和事件来开始和停止 SPEL+ 任务。
- 在窗体中添加<暂停>和<继续>按钮。这将展示 EventTaskType 属性的使用。
- 添加设置窗体。此窗体将使用标签和文本框控件来允许用户更改机器人速度。
- 在主窗体上添加按钮，以显示设置窗体。这将展示 GShowDialog 语句和 DialogResult 属性的使用。

遵循以下步骤：

- 1 创建新的 EPSON RC+ 7.0 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：

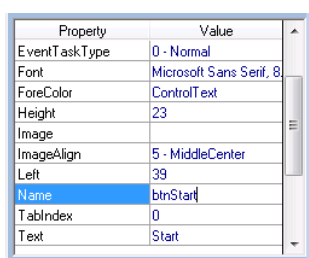
```
Function main
  Robot1
  Motor On
  Do
    Go P0
    Wait 0.5
    Go P1
    Wait 0.5
  Loop
End
```

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。

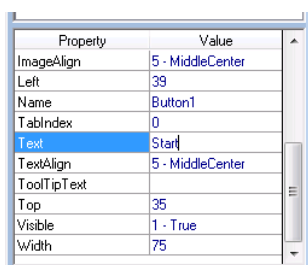


6. 点击GUI Builder窗口工具栏上的“新建按钮”按钮，然后点击窗体上的鼠标。将创建一个新按钮。

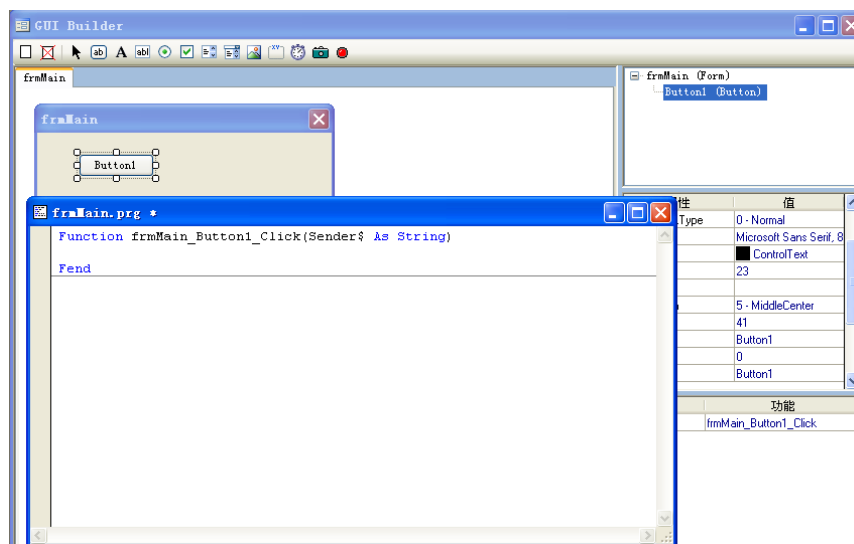
7. 在属性网格中，向下滚动至 Name 属性，然后将名称改为 btnStart 并按下 ENTER。



8. 在属性网格中，向下滚动至 Text 属性，然后将 Button1 文本改为 Start 并按下 ENTER。



9. 双击窗体上的<开始>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。

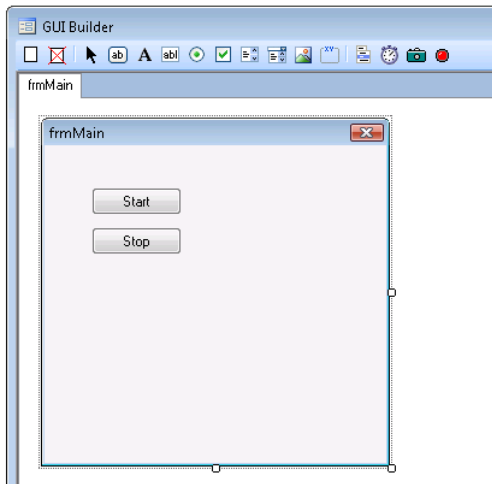


10. 更改 frmMain\_btnStart\_Click 函数，以启动 main 任务，如下所示。这样将在用户点击<开始>按钮时启动 main 函数。

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

11. 点击GUI Builder窗口工具栏上的“新建按钮”按钮，然后点击<开始>按钮下方的窗体创建另一按钮。

- 对于新按钮，将 Name 属性改为 btnStop，Text 属性改为 Stop。此时，窗体应如下所示：



- 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
```

- 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。
- 选择运行窗口上的<型号>单选按钮。
- 点击运行窗口上的<开始>按钮。
- 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。
- 点击窗体上的<Stop>按钮。机器人任务将停止。
- 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<暂停>和<继续>按钮。

- 点击 GUI Builder 工具栏按钮再次显示 GUI Builder 窗口。
- 点击 GUI Builder 窗口工具栏上的“新建按钮”按钮，然后点击<开始>按钮右侧的窗体创建新按钮。
- 对于新按钮，将 Name 属性改为 btnPause，Text 属性改为 Pause。
- 将<暂停>按钮的 EventTaskType 改为 l- 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。
- 双击<暂停>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
```

- 点击 GUI Builder 窗口功能栏上的“新建按钮”按钮，然后点击<停止>按钮右侧的窗体创建新按钮。
- 对于新按钮，将 Name 属性改为 “btnCont”，Text 属性改为 “Continue”。

27. 将<继续>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。
28. 双击<继续>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：

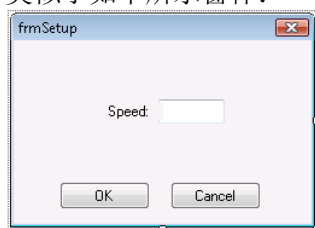
```
Function frmMain_btnCont_Click(Sender$ As String)
    Cont
End
```

29. 按 F5 构建项目并打开运行窗口。如果发生任何构建错误，则修改代码并再按 F5。
  30. 点击运行窗口上的<开始>按钮。将显示窗体。
  31. 点击窗体上的<开始>按钮。机器人循环将执行。
  32. 点击窗体上的<暂停>按钮。机器人循环将暂停。
  33. 点击窗体上的<继续>按钮。机器人循环将继续。
  34. 点击窗体上的<停止>按钮，然后点击右上角的 X 按钮关闭窗口。
- 现在我们将在 GUI 上添加设置窗体。

35. 点击GUI Builder窗体工具栏上的“新建窗体”按钮，并将新窗体命名为 frmSetup，然后点击<确定>。将出现显示新窗体的新选项卡页面。
36. 点击“新建标签”按钮并点击frmSetup创建标签。
37. 在属性网格中，将新标签名称改为“lblSpeed”。
38. 将Text属性改为“Speed:”。
39. 点击“新建文本框”按钮并点击Speed标签右侧的frmSetup。
40. 将Name属性改为“txtSpeed”，Text属性改为“10”。
41. 双击 frmSetup 为窗体创建 Load 事件处理程序函数并按如下所示更改代码：

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

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“btnOK”，并将Text属性设为“OK”。
43. 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：



44. 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
```

45. 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
```

46. 单击 frmMain 选项卡，重新与 frmMain 一起工作。
47. 单击 GUI Builder 窗口工具栏上的“新建按钮”按钮，然后单击窗体以添加按钮。将此按钮命名为“btnSetup”，并将 Text 属性设为“Setup”。
48. 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
End
```

49. 按 F5 构建项目并打开运行窗口。
50. 单击运行窗口上的<开始>按钮。将显示主窗体。
51. 单击主窗体上的<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. 输入新速度并单击<确定>。
53. 单击主窗体上的<开始>按钮，以新速度运行机器人循环。  
如果点击了<停止>按钮且机器人停止，则动作速度将重置为默认速度。
54. 至此便完成了教程。



## 4. GUI Builder 环境

### 4.1 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

### 4.2 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。

### 4.3 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 EPSON RC+ 的主工具栏中，您应该会看到 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。

### 4.4 GUI Builder 窗口的组成部分

GUI Builder 窗口如下所示。后续章节将分别介绍各所示部分。



#### 4.4.1 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。













### 4.4.2 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：



GUI Builder 工具栏

下文所示为每个 GUI Builder 工具栏按钮的一般描述。

按钮	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 选件显示视频。
	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。

### 4.4.3 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。

双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。

### 4.4.4 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** 为属性名称。**Value** 为可编辑的当前值。

### 4.4.5 事件网格

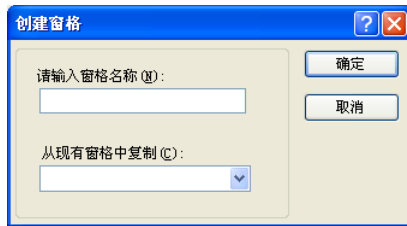
事件网格用于显示和更改相关窗体或控件的事件。每个事件均具有一个在发生事件时会调用的用户函数。

## 4.5 窗体和控件的使用

### 4.5.1 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。您可能想要使用以“frm”为前缀的名称。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。

### 4.5.2 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，然后选择<删除>。

### 4.5.3 打开和关闭窗口

若要打开窗体，双击窗体资源管理器中的窗体。

若要关闭窗口，右击窗体资源管理器中的窗体并选择<关闭>，或者右击窗体选项卡并选择<关闭>。

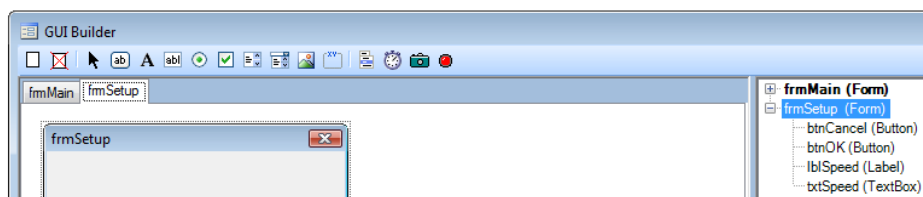
### 4.5.4 更改窗体尺寸

可采用两种方式更改窗体尺寸：

- 使用鼠标拖拽 GUI Builder 窗口设计区域内的一个窗体尺寸句柄。
- 设计时在属性网格中更改 Height 和 Width 属性或在运行时使用 GSet。

### 4.5.5 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。

### 4.5.6 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。

### 4.5.7 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。

### 4.5.8 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。
- 设计时在属性网格中更改 Height 和 Width 属性或在运行时使用 GSet。

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。

### 4.5.9 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。选定控件将以白色（第一个控件）和黑色尺寸句柄指示。

复制选定控件

若要复制选定控件，按下 Ctrl+C，或点击主工具栏上的复制按钮，或从编辑菜单中选择复制。

剪切选定控件

若要剪切选定控件，按下 Ctrl+X，或点击主工具栏上的剪切按钮，或从编辑菜单中选择剪切。

粘贴控件

若要粘贴已复制或剪切的控件，按下 Ctrl+V，或点击主工具栏上的粘贴按钮，或从编辑菜单中选择粘贴。

### 4.5.10 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。点击值右侧的椭圆按钮。从对话框中选择新值。选择之后或在对话框中点击<确定>之后便会应用更改。

### 4.5.11 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：

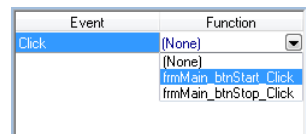
-双击窗体或控件创建默认事件处理程序。例如，如果双击按钮控件，则会创建点击事件处理程序函数。

-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。

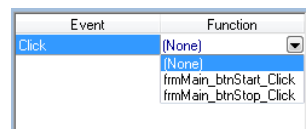
要更改事件处理程序

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，在事件值下拉列表中选择 (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。

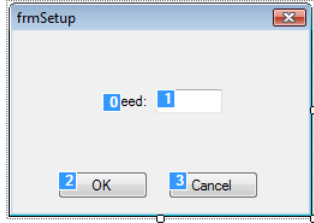
例如，如果按钮点击事件处理程序将执行Pause或Cont，则EventTaskType必须设为1 - NoPause。或者如果事件处理程序在紧急停止情况下必须执行Reset，则EventTaskType必须设为2 - NoEmgAbort。

### 4.5.12 更改选项卡顺序

在窗体中添加控件之后，您可能会想更改选项卡顺序。即用户在运行时输入选项卡时设置的焦点顺序。另外重要的是，在标识控件之前使用热键将标签的选项卡顺序设为一个选项卡索引。

要更改选项卡顺序

1. 打开想要更改选项卡顺序的窗体。
2. 按下 Tab 键。将显示如下所示的选项卡顺序：



3. 以您想要的顺序点击每个选项卡顺序值。
4. 再按下 Tab 键隐藏选项卡顺序。

### 4.5.13 保存工作

在 GUI Builder 窗口中进行更改之后，您可以通过两种方式保存工作：

- 从文件菜单中选择保存或按下 Ctrl+S。
- 点击 EPSON RC+ 主工具栏上的项目保存工具栏按钮。



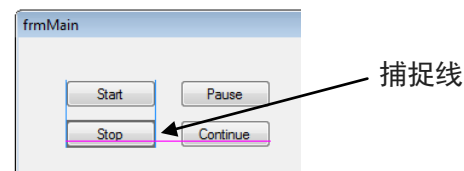
## 4.6 设置 GUI Builder 选项

您可以通过设置 GUI Builder 选项改变 GUI Builder 窗口设计区域中窗体上的控件管理方式。

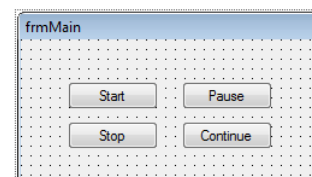
选择设置菜单|选项|GUI 创建器。



如果布局模式设为捕捉线，当您在窗体中移动控件时，GUI Builder 会在控件周围显示线段。



如果布局模式设为对齐网格，各窗体上会显示网格，当您移动窗体上的控件时，控件会对齐到各网格位置。



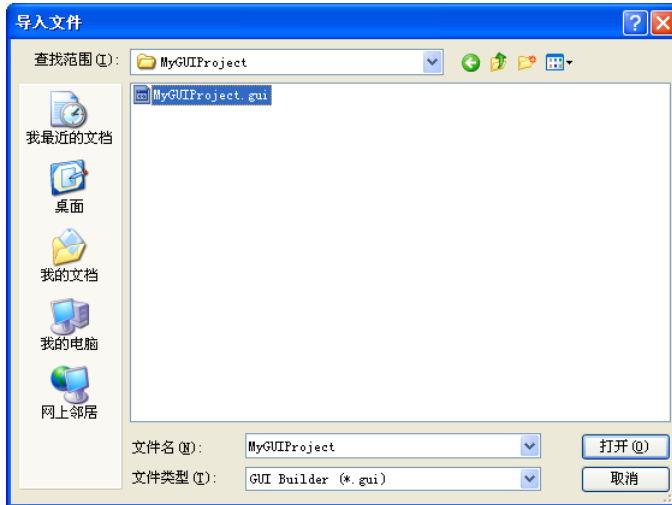
若要更改网格尺寸，可更改<网格宽度>和<网格高度>值并点击<应用>。

当您更改 GUI Builder 选项并点击<应用>时，如果 GUI Builder 窗口打开，则会保存当前更改并重新打开窗口以使用新选项。

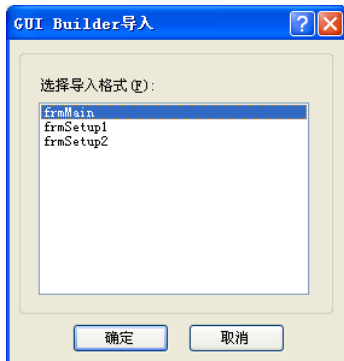
### 4.7 导入其他项目的窗体

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。
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。

## 5. GUI Builder 组件

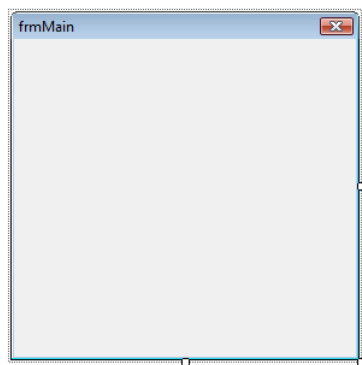
本章将介绍窗体以及窗体上可用的所有控件。每个组件章节将提供与组件相关的所有属性和事件的使用信息。

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](#)。

### 5.1 窗体

#### 5.1.1 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。

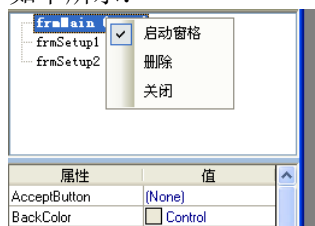


#### 5.1.2 使用

显示窗体

窗体可通过三种方式显示：

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
```

### 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 6.6 使用帮助文件。

## 5.1.3 窗体属性

属性	描述
<b>AcceptButton</b>	设置用户键入 Enter 时执行点击事件的按钮控件。 默认: None
<b>BackColor</b>	设置窗体的背景颜色。 默认: Control
<b>CancelButton</b>	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
<b>ControlBox</b>	设置是否显示标题栏控件框项目。 默认: 1 - True
<b>Controls</b>	窗体上的控件数组。
<b>Count</b>	获取控件数组中的控件数量。
<b>Dialog Result</b>	设置窗体关闭时的返回值 (仅运行时)。
<b>EventTaskType</b>	设置用于事件的任务类型。 默认: 0 - Normal
<b>FormBorderStyle</b>	设置用于窗体的边框类型。 默认: 3 - FixedDialog
<b>Height</b>	设置控件高度, 单位为像素。
<b>HelpButton</b>	设置是否显示标题栏帮助按钮。 默认: 0 - False
<b>HelpID</b>	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
<b>Left</b>	设置窗体的左坐标, 单位为像素。
<b>MaximizeBox</b>	设置是否显示标题栏最大化按钮。 默认: 0 - False
<b>MinimizeBox</b>	设置是否显示标题栏最小化按钮。 默认: 0 - False
<b>Name</b>	设置控件名称。
<b>StartPosition</b>	设置窗体的启动位置。 默认: 1 - CenterScreen
<b>Text</b>	用于设置控件文本。 默认: 窗体名称
<b>Top</b>	获取或设置窗体的顶坐标, 单位为像素。
<b>Type</b>	获取控件的类型名称。
<b>Width</b>	设置窗体宽度, 单位为像素。
<b>WindowState</b>	设置窗体的默认窗口状态。 默认: 0 - Normal

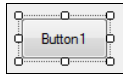
## 5.1.4 窗体事件

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

## 5.2 按钮控件

### 5.2.1 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。



### 5.2.2 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。

### 5.2.3 按钮控件属性

属性	描述
<b>BackColor</b>	控件的背景颜色。 默认: Control
<b>BackColorMode</b>	控件的背景颜色模式。 默认: 0 – Visual Style
<b>Enabled</b>	设置是否在运行时启用控件。 默认: 1 – True
<b>EventTaskType</b>	用于事件的任务类型。 默认: 0 – Normal
<b>Font</b>	控件文本的字体（仅设计时）。 默认: Microsoft Sans Serif 8.25 pt
<b>FontBold</b>	当前字体的粗体属性（仅运行时）。 默认: 0 - False
<b>FontItalic</b>	当前字体的斜体属性（仅运行时）。 默认: 0 - False
<b>FontName</b>	当前字体的名称（仅运行时）。 默认: Microsoft Sans Serif
<b>FontSize</b>	点中当前字体的大小（仅运行时）。 默认: 8.25
<b>ForeColor</b>	控件文本的前景颜色。 默认: ControlText
<b>Height</b>	控件高度，单位为像素。
<b>Image</b>	控件图片。若要删除，可按下<删除>键。 默认: 空
<b>ImageAlign</b>	控件的图片对齐。 默认: 5 – MiddleCenter
<b>Left</b>	控件的左坐标，单位为像素。
<b>Name</b>	控件名称。 默认: ButtonXX

属性	描述
<b>TabIndex</b>	控件的选项卡索引。
<b>Text</b>	控件文本。 默认：控件名称
<b>TextAlign</b>	选择文本对齐。 默认：5 – MiddleCenter
<b>ToolTipText</b>	控件提示信息中使用的文本。 默认：空
<b>Top</b>	控件的顶坐标，单位为像素。
<b>Visible</b>	控件在运行时是否可见。 默认：1 – True
<b>Width</b>	控件宽度，单位为像素。

#### 5.2.4 按钮控件事件

事件	描述
<b>Click</b>	用户使用鼠标单击控件时执行。

## 5.3 标签控件

### 5.3.1 描述

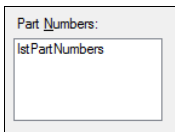
标签控件用于显示用户无法编辑的文本或图像。还用于识别窗体上的对象，以描述点击某个控件会执行的动作，例如，根据运行时间事件或应用中的过程显示信息。由于标签控件无法接收焦点，还可用于创建其他控件的快捷键。



### 5.3.2 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

使用 BorderStyle、ForeColor、BackColor、Font、Image、ImageAlign 和 TextAlign 属性可更改标签外观。

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。请参阅章节 [更改选项卡顺序](#)。

### 5.3.3 标签控件属性

属性	描述
<b>BackColor</b>	控件的背景颜色。 默认：Control
<b>BorderStyle</b>	用于控件的边框。 默认：0 – None
<b>Enabled</b>	设置是否在运行时启用控件。 默认：1 – True
<b>EventTaskType</b>	用于事件的任务类型。 默认：0 – Normal
<b>Font</b>	控件文本的字体（仅设计时）。 默认：Microsoft Sans Serif 8.25 pt



属性	描述
<b>FontBold</b>	当前字体的粗体属性（仅运行时）。 默认：0 – False
<b>FontItalic</b>	当前字体的斜体属性（仅运行时）。 默认：0 – False
<b>FontName</b>	当前字体的名称（仅运行时）。 默认：Microsoft Sans Serif
<b>FontSize</b>	点中当前字体的大小（仅运行时）。 默认：8.25
<b>ForeColor</b>	控件文本的前景颜色。 默认：ControlText
<b>Height</b>	控件高度，单位为像素。
<b>Image</b>	控件图片。若要删除，可按下<删除>键。 默认：空
<b>ImageAlign</b>	控件的图片对齐。 默认：5 – MiddleCenter
<b>Left</b>	控件的左坐标，单位为像素。
<b>Name</b>	控件名称。 默认：LabelXX
<b>TabIndex</b>	控件的选项卡索引。
<b>Text</b>	控件文本。 默认：控件名称
<b>TextAlign</b>	选择文本对齐。 默认：1 – TopLeft
<b>ToolTipText</b>	控件提示信息中使用的文本。 默认：空
<b>Top</b>	控件的顶坐标，单位为像素。
<b>Visible</b>	控件在运行时是否可见。 默认：1 – True
<b>Width</b>	控件宽度，单位为像素。

### 5.3.4 标签控件事件

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

## 5.4 文本框控件

### 5.4.1 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。



### 5.4.2 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。

设置文本框外观

使用 **BorderStyle**、**ForeColor**、**BackColor**、**Font** 和 **TextAlign** 属性可更改文本框的外观。

显示 SPEL+ 全局变量状态

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 EPSEL+全局变量未在 **Variable** 属性设置中显示，请构建项目。

### 5.4.3 文本框控件属性

属性	描述
<b>AppendText</b>	运行时附加文本。
<b>BackColor</b>	控件的背景颜色。 默认：Window
<b>BorderStyle</b>	用于控件的边框。 默认：2 - Fixed3D
<b>Enabled</b>	设置是否在运行时启用控件。 默认：1 - True
<b>EventTaskType</b>	用于事件的任务类型。 默认：0 - Normal
<b>Font</b>	控件文本的字体（仅设计时）。 默认：Microsoft Sans Serif 8.25 pt
<b>FontBold</b>	当前字体的粗体属性（仅运行时）。 默认：0 - False
<b>FontItalic</b>	当前字体的斜体属性（仅运行时）。 默认：0 - False

属性	描述
<b>FontName</b>	当前字体的名称（仅运行时）。 默认：Microsoft Sans Serif
<b>FontSize</b>	当前字体的大小，单位为点（仅运行时）。 默认：8.25
<b>ForeColor</b>	控件文本的前景颜色。 默认：WindowText
<b>Height</b>	控件高度，单位为像素。
<b>Left</b>	控件的左坐标，单位为像素。
<b>Multiline</b>	设置是否显示一行或多行。 默认：0 – False
<b>Name</b>	控件名称。 默认：TextBoxXX
<b>PasswordChar</b>	设置用于隐藏输入的各字符的字符。 默认：空
<b>ReadOnly</b>	设置用户是否可以编辑文本。 默认：0 – False
<b>ScrollBars</b>	设置如何显示控件的滚动条。 默认：0 – None
<b>ShowPrint</b>	设置是否显示 Print 语句的输出。 默认：0 - False
<b>TabIndex</b>	控件的选项卡索引。
<b>Text</b>	控件文本。 默认：空
<b>TextAlign</b>	选择文本对齐。 默认：1 – Left
<b>ToolTipText</b>	控件提示信息中使用的文本。 默认：空
<b>Top</b>	控件的顶坐标，单位为像素。
<b>Update</b>	更新文本控件的显示（仅运行时）。
<b>Variable</b>	这是一个可选的 SPEL+ 全局变量。 默认：None
<b>Visible</b>	控件在运行时是否可见。 默认：1 – True
<b>Width</b>	控件宽度，单位为像素。
<b>WordWrap</b>	设置文字是否换行。 默认：1 – True

#### 5.4.4 文本框控件事件

事件	描述
<b>KeyPress</b>	当控件存在焦点并按下按键时，会发生此事件。

## 5.5 单选按钮控件

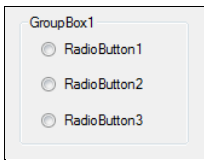
### 5.5.1 描述

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。



### 5.5.2 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

使用 **BorderStyle**、**ForeColor**、**BackColor**、**Font**、**Image**、**ImageAlign** 和 **TextAlign** 属性可更改单选按钮的外观。

## 5.5.3 单选按钮控件属性

属性	描述
<b>BackColor</b>	控件的背景颜色。 默认: Control
<b>Checked</b>	设置是否在运行时选择控件。 默认: 0 – False
<b>Enabled</b>	设置是否在运行时启用控件。 默认: 1 – True
<b>EventTaskType</b>	用于事件的任务类型。 默认: 0 – Normal
<b>Font</b>	控件文本的字体 (仅设计时)。 默认: Microsoft Sans Serif 8.25 pt
<b>FontBold</b>	当前字体的粗体属性 (仅运行时)。 默认: 0 – False
<b>FontItalic</b>	当前字体的斜体属性 (仅运行时)。 默认: 0 – False
<b>FontName</b>	当前字体的名称 (仅运行时)。 默认: Microsoft Sans Serif
<b>FontSize</b>	点中当前字体的大小 (仅运行时)。 默认: 8.25
<b>ForeColor</b>	控件文本的前景颜色。 默认: ControlText
<b>Height</b>	控件高度, 单位为像素。
<b>Image</b>	控件图片。若要删除, 可按下<删除>键。 默认: 空
<b>ImageAlign</b>	控件的图片对齐。 默认: 5 – MiddleCenter
<b>Left</b>	控件的左坐标, 单位为像素。
<b>Name</b>	控件名称。 默认: RadioButtonXX
<b>TabIndex</b>	控件的选项卡索引。
<b>Text</b>	控件文本。 默认: 控件名称
<b>TextAlign</b>	选择文本对齐。 默认: 4 – Middle Left
<b>ToolTipText</b>	控件提示信息中使用的文本。 默认: 空
<b>Top</b>	控件的顶坐标, 单位为像素。
<b>Visible</b>	控件在运行时是否可见。 默认: 1 – True
<b>Width</b>	控件宽度, 单位为像素。

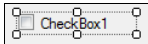
### 5.5.4 单选按钮控件事件

事件	描述
Click	用户使用鼠标单击控件时执行。

## 5.6 复选框控件

### 5.6.1 描述

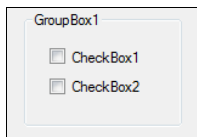
复选框控件表示是否打开或关闭某一特定条件。通常用于为用户提供是/否或 True/False 的选择。您可以成组使用复选框显示用户可从中选择一个或多个选项的诸多选项。复选框类似于单选按钮，但可选择任意数量的分组复选框控件。



### 5.6.2 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

使用 **BorderStyle**、**ForeColor**、**BackColor**、**Font**、**Image**、**ImageAlign** 和 **TextAlign** 属性可更改复选框的外观。

### 5.6.3 复选框控件属性

属性	描述
<b>BackColor</b>	控件的背景颜色。 默认: Control
<b>Checked</b>	设置是否在运行时选中控件。 默认: 0 – False
<b>Enabled</b>	设置是否在运行时启用控件。 默认: 1 – True
<b>EventTaskType</b>	用于事件的任务类型。 默认: 0 – Normal
<b>Font</b>	控件文本的字体（仅设计时）。 默认: Microsoft Sans Serif 8.25 pt
<b>FontBold</b>	当前字体的粗体属性（仅运行时）。 默认: 0 – False
<b>FontItalic</b>	当前字体的斜体属性（仅运行时）。 默认: 0 – False

属性	描述
<b>FontName</b>	当前字体的名称（仅运行时）。 默认：Microsoft Sans Serif
<b>FontSize</b>	点中当前字体的大小（仅运行时）。 默认：8.25
<b>ForeColor</b>	控件文本的前景颜色。 默认：ControlText
<b>Height</b>	控件高度，单位为像素。
<b>Image</b>	控件图片。若要删除，可按下<删除>键。 默认：空
<b>ImageAlign</b>	控件的图片对齐。 默认：5 – MiddleCenter
<b>Left</b>	控件的左坐标，单位为像素。
<b>Name</b>	控件名称。 默认：CheckBoxXX
<b>TabIndex</b>	控件的选项卡索引。
<b>Text</b>	控件文本。 默认：控件名称
<b>TextAlign</b>	选择文本对齐。 默认：4 – MiddleLeft
<b>ToolTipText</b>	控件提示信息中使用的文本。 默认：空
<b>Top</b>	控件的顶坐标，单位为像素。
<b>Visible</b>	控件在运行时是否可见。 默认：1 – True
<b>Width</b>	控件宽度，单位为像素。

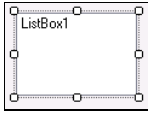
#### 5.6.4 复选框控件事件

事件	描述
<b>Click</b>	用户使用鼠标单击控件时执行。

### 5.7 列表框控件

#### 5.7.1 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。



#### 5.7.2 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。



## 5.7.3 列表框控件属性

属性	描述
<b>AddItem</b>	运行时在列表中添加项目。
<b>BackColor</b>	控件的背景颜色。 默认: Window
<b>BorderStyle</b>	用于控件的边框。 默认: 2 – Fixed3D
<b>Enabled</b>	设置是否在运行时启用控件。 默认: 1 – True
<b>EventTaskType</b>	用于事件的任务类型。 默认: 0 – Normal
<b>Font</b>	控件文本的字体 (仅设计时)。 默认: Microsoft Sans Serif 8.25 pt
<b>FontBold</b>	当前字体的粗体属性 (仅运行时)。 默认: 0 – False
<b>FontItalic</b>	当前字体的斜体属性 (仅运行时)。 默认: 0 – False
<b>FontName</b>	当前字体的名称 (仅运行时)。 默认: Microsoft Sans Serif
<b>FontSize</b>	点中当前字体的大小 (仅运行时)。 默认: 8.25
<b>ForeColor</b>	控件文本的前景颜色。 默认: WindowText
<b>Height</b>	控件高度, 单位为像素。
<b>Left</b>	控件的左坐标, 单位为像素。
<b>List</b>	用于在运行时访问列表中的项目。
<b>ListCount</b>	用于在运行时获取列表中的项目数量。
<b>Name</b>	控件名称。 默认: ListBoxXX
<b>SelectedIndex</b>	用于在运行时确定操作员选择的项目。
<b>Sorted</b>	设置是否对列表中的项目分类。 默认: 0 – False
<b>TabIndex</b>	控件的选项卡索引。
<b>ToolTipText</b>	控件提示信息中使用的文本。 默认: 空
<b>Top</b>	控件的顶坐标, 单位为像素。
<b>Visible</b>	控件在运行时是否可见。 默认: 1 – True
<b>Width</b>	控件宽度, 单位为像素。

### 5.7.4 列表框控件事件

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

## 5.8 组合框控件

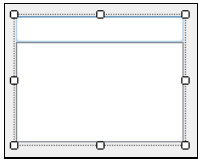
### 5.8.1 描述

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项目列表的列表框。

### 5.8.2 使用

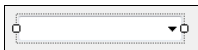
根据 `DropDownStyle` 属性的值，组合框控件具有三种不同的使用方法。

`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。

## 5.8.3 组合框控制属性

属性	描述
<b>AddItem</b>	运行时在列表中添加项目。
<b>BackColor</b>	控件的背景颜色。 默认: Window
<b>DropDownStyle</b>	指定组合框的类型。 默认: 1 – DropDown
<b>Enabled</b>	设置是否在运行时启用控件。 默认: 1 – True
<b>EventTaskType</b>	用于事件的任务类型。 默认: 0 – Normal
<b>Font</b>	控件文本的字体 (仅设计时)。 默认: Microsoft Sans Serif 8.25 pt
<b>FontBold</b>	当前字体的粗体属性 (仅运行时)。 默认: 0 – False
<b>FontItalic</b>	当前字体的斜体属性 (仅运行时)。 默认: 0 – False
<b>FontName</b>	当前字体的名称 (仅运行时)。 默认: Microsoft Sans Serif
<b>FontSize</b>	点中当前字体的大小 (仅运行时)。 默认: 8.25
<b>ForeColor</b>	控件文本的前景颜色。 默认: WindowText
<b>Height</b>	控件高度, 单位为像素。
<b>Left</b>	控件的左坐标, 单位为像素。
<b>List</b>	用于在运行时访问列表中的项目。
<b>ListCount</b>	用于在运行时获取列表中的项目数量。
<b>Name</b>	控件名称。 默认: ComboBoxXX
<b>SelectedIndex</b>	用于在运行时确定操作员选择的项目。
<b>Sorted</b>	设置是否对列表中的项目分类。 默认: 0 – False
<b>TabIndex</b>	控件的选项卡索引。
<b>Text</b>	控件文本。 默认: 空
<b>ToolTipText</b>	控件提示信息中使用的文本。 默认: 空
<b>Top</b>	控件的顶坐标, 单位为像素。
<b>Visible</b>	控件在运行时是否可见。 默认: 1 – True
<b>Width</b>	控件宽度, 单位为像素。

## 5.8.4 组合框控件事件

事件	描述
<b>Click</b>	用户使用鼠标单击控件时执行。

## 5.9 图片框控件

### 5.9.1 描述

图片框控件用于显示位图、GIF、JPEG、PNG、WMF、图元文件或图标格式的图形。

### 5.9.2 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，可使用 **Image** 属性浏览图像文件。

运行时，可使用 **Image** 属性指定图像文件的完整路径。

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。

### 5.9.3 图片框控件属性

属性	描述
<b>BackColor</b>	控件的背景颜色。 默认: Control
<b>BorderStyle</b>	用于控件的边框。 默认: 0 – None
<b>Enabled</b>	设置是否在运行时启用控件。 默认: 1 – True
<b>EventTaskType</b>	用于事件的任务类型。 默认: 0 – Normal
<b>Height</b>	控件高度，单位为像素。
<b>Image</b>	指定待显示的图像。若要删除，可按下<删除>键。
<b>Left</b>	控件的左坐标，单位为像素。
<b>Name</b>	控件名称。 默认: PictureBoxXX
<b>SizeMode</b>	指定图像尺寸。 默认: 0 – Normal
<b>TabIndex</b>	控件的选项卡索引。
<b>ToolTipText</b>	控件提示信息中使用的文本。 默认: 空
<b>Top</b>	控件的顶坐标，单位为像素。
<b>Visible</b>	控件在运行时是否可见。 默认: 1 – True
<b>Width</b>	控件宽度，单位为像素。

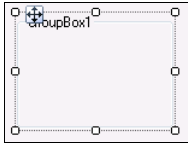
#### 5.9.4 图片框控件事件

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

## 5.10 分组框控件

### 5.10.1 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。例如，您可能设有指定机器人速度选项的设置窗体。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。



### 5.10.2 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。

### 5.10.3 分组框控件属性

属性	描述
<b>BackColor</b>	控件的背景颜色。 默认: Control
<b>Enabled</b>	设置是否在运行时启用控件。 默认: 1 – True
<b>EventTaskType</b>	用于事件的任务类型。 默认: 0 – Normal
<b>Font</b>	控件文本的字体（仅设计时）。 默认: Microsoft Sans Serif 8.25 pt
<b>ForeColor</b>	控件文本的前景颜色。 默认: ControlText
<b>FontBold</b>	当前字体的粗体属性（仅运行时）。 默认: 0 – False
<b>FontItalic</b>	当前字体的斜体属性（仅运行时）。 默认: 0 – False
<b>FontName</b>	当前字体的名称（仅运行时）。 默认: Microsoft Sans Serif
<b>FontSize</b>	点中当前字体的大小（仅运行时）。 默认: 8.25

属性	描述
<b>ForeColor</b>	控件文本的前景颜色。 默认: ControlText
<b>Height</b>	控件高度, 单位为像素。
<b>Left</b>	控件的左坐标, 单位为像素。
<b>Name</b>	控件名称。 默认: GroupBoxXX
<b>TabIndex</b>	控件的选项卡索引。 默认:
<b>Text</b>	控件文本。 默认: 控件名称
<b>ToolTipText</b>	控件提示信息中使用的文本。 默认: 空
<b>Top</b>	控件的顶坐标, 单位为像素。
<b>Visible</b>	控件在运行时是否可见。 默认: 1 – True
<b>Width</b>	控件宽度, 单位为像素。

#### 5.10.4 分组框控件事件

事件	描述
<b>Click</b>	用户使用鼠标单击控件时执行。

### 5.11 定时器控件

#### 5.11.1 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。



#### 5.11.2 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。

NOTE



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。

#### 5.11.3 定时器控件属性

属性	描述
<b>Enabled</b>	设置是否启用控件。 默认：0 – False
<b>Interval</b>	设置时间间隔，单位为毫秒。 默认：100
<b>Name</b>	控件名称。 默认：TimerXX

#### 5.11.4 定时器控件事件

事件	描述
<b>Tick</b>	定时器启用且达到间隔时发生。



## 5.12 视频框控件

### 5.12.1 描述

视频框控件可实现在窗体内显示 Vision Guide 选件的视频。

### 5.12.2 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。

### 5.12.3 视频框控件属性

属性	描述
<b>BorderStyle</b>	用于控件的边框。 默认：0 – None
<b>Camera</b>	选择将显示视频的相机。 默认：0
<b>Enabled</b>	设置是否在运行时启用控件。 默认：1 – True
<b>EventTaskType</b>	用于事件的任务类型。 默认：0 – Normal
<b>GraphicsEnabled</b>	设置是否显示视觉图形。 默认：0 – False
<b>Height</b>	控件高度，单位为像素。
<b>Left</b>	控件的左坐标，单位为像素。
<b>Name</b>	控件名称。 默认：VideoBoxXX
<b>TabIndex</b>	控件的选项卡索引。
<b>ToolTipText</b>	控件提示信息中使用的文本。 默认：空
<b>Top</b>	控件的顶坐标，单位为像素。
<b>VideoEnabled</b>	设置是否显示视频。 默认：0 - False
<b>Visible</b>	控件在运行时是否可见。

	默认: 1 – True
<b>Width</b>	控件宽度, 单位为像素。

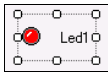
### 5.12.4 视频框控件事件

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

## 5.13 LED 控件

### 5.13.1 描述

LED 控件用于显示 I/O 状态。



### 5.13.2 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

使用 BackColor、BorderStyle、ForeColor、Font、ImageAlign 和 TextAlign 属性可更改 LED 的外观。

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。

### 5.13.3 LED 控件属性

属性	描述
<b>AllowStateChange</b>	允许操作员切换输出状态。 默认: 0 – False
<b>BackColor</b>	控件的背景颜色。 默认: Control
<b>BorderStyle</b>	用于控件的边框。 默认: 0 – None
<b>Enabled</b>	设置是否在运行时启用控件。 默认: 1 – True
<b>EventTaskType</b>	用于事件的任务类型。 默认: 0 – Normal
<b>Font</b>	控件文本的字体 (仅设计时)。 默认: Microsoft Sans Serif 8.25 pt
<b>FontBold</b>	当前字体的粗体属性 (仅运行时)。 默认: 0 – False

<b>FontItalic</b>	当前字体的斜体属性（仅运行时）。 默认：0 – False
<b>FontName</b>	当前字体的名称（仅运行时）。 默认：Microsoft Sans Serif
<b>FontSize</b>	点中当前字体的大小（仅运行时）。 默认：8.25
<b>ForeColor</b>	控件文本的前景颜色。 默认：ControlText
<b>Height</b>	控件高度，单位为像素。
<b>ImageAlign</b>	控件的图片对齐。 默认：4 - MiddleLeft
<b>ImageOff</b>	设置 I/O 状态关闭时显示的图像。 默认：LedOff.ico
<b>ImageOn</b>	设置 I/O 状态打开时显示的图像。 默认：LedRed.ico
<b>IOBit</b>	范围：0 至 9999 默认：0
<b>IOType</b>	设置待监测 I/O 的类型。 默认：0 – Input
<b>Left</b>	控件的左坐标，单位为像素。
<b>Name</b>	控件名称。 默认：LedXX
<b>TabIndex</b>	控件的选项卡索引。
<b>Text</b>	控件文本。 默认：控件名称
<b>TextAlign</b>	选择文本对齐。 默认：6 – MiddleRight
<b>ToolTipText</b>	控件提示信息中使用的文本。 默认：空
<b>Top</b>	控件的顶坐标，单位为像素。
<b>Visible</b>	控件在运行时是否可见。 默认：1 – True
<b>Width</b>	控件宽度，单位为像素。

#### 5.13.4 LED 控件事件

事件	描述
<b>DbtClick</b>	用户使用鼠标双击控件时执行。

## 5.14 状态栏控件

### 5.14.1 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)。

### 5.14.2 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。

### 5.14.3 状态栏控件属性

属性	描述
<b>BackColor</b>	控件的背景颜色。 默认: Control
<b>BorderStyle</b>	设置控件的边框样式。 默认: 0 – None
<b>Dock</b>	设置控件的停放位置。 默认: 2 –Bottom
<b>Enabled</b>	设置是否启用控件。 默认: 1 – True
<b>EventTaskType</b>	设置用于事件的任务类型。 默认: 0 – Normal
<b>Font</b>	设置控件文本的字体。 (仅设计时候有效) 默认: Microsoft Sans Serif 8.25 pt
<b>ForeColor</b>	设置控件的文本颜色。 默认: ControlText
<b>Height</b>	设置控件的高度，以像素为单位。
<b>Left</b>	设置控件显示位置的左坐标，以像素为单位。
<b>Name</b>	设置控件的名称。 默认: StatusBarXX
<b>RobotNumber</b>	设置用于 ShowRobot 的机器人编号。
<b>ShowDateTime</b>	设置是否显示当前日期和时间。 默认: 0 – False
<b>ShowEStop</b>	设置是否显示 Estop 状态。 默认: 0 – False
<b>ShowRobot</b>	设置是否显示由 RobotNumber 指定的机器人。 默认: 0 – False
<b>ShowSafeguard</b>	设置是否显示安全门状态。 默认: 0 – False
<b>TabIndex</b>	设置控件的选项卡索引。

属性	描述
<b>Text</b>	设置控件的显示文本。 默认: 控制器名称
<b>ToolTipText</b>	设置控制器的工具提示中显示的文本。 默认: 空白
<b>Top</b>	设置控件显示位置的上坐标，以像素为单位。
<b>Visible</b>	控件在运行时是否可见。 默认: 1 – True
<b>Width</b>	设置控件的宽度，以像素为单位。

#### 5.14.4 状态栏控件事件

事件	描述
<b>Click</b>	单击控件时执行。

### 5.15 进度条控件

#### 5.15.1 描述

使用进度条控件显示任务的处理进度。

#### 5.15.2 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

通过设置 Variable 属性自动显示 SPEL+全局变量的值。

注意

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

#### 5.15.3 进度条控件属性

属性	描述
<b>BackColor</b>	控件的背景颜色。 默认: Control
<b>BorderStyle</b>	设置控件的边框样式。 默认: 0 – None
<b>Dock</b>	设置控件的停放位置。 默认: 0 – None
<b>Enabled</b>	设置是否启用控件。 默认: 1 – True
<b>EventTaskType</b>	设置用于事件的任务类型。 默认: 0 – Normal
<b>ForeColor</b>	设置控件的文本颜色。 默认: ControlText
<b>Height</b>	设置控件的高度，以像素为单位。
<b>Left</b>	设置控件显示位置的左坐标，以像素为单位。
<b>Maximum</b>	设置控件正在处理的范围的上限。 默认: 100
<b>Minimum</b>	设置控件正在处理的范围的下限。 默认: 0
<b>Name</b>	设置控件的名称。 默认: ProgressBarXX
<b>Orientation</b>	设置控件的方向。 默认: 0 – Horizontal
<b>ProgressBarStyle</b>	设置控件的样式。 默认: 0 – Blocks
<b>Step</b>	设置控件的当前值增加的数量。 默认: 10
<b>TabIndex</b>	设置控件的选项卡索引。

属性	描述
<b>ToolTipText</b>	设置控件的工具提示中显示的文本。 默认: 空白
<b>Top</b>	设置控件显示位置的上坐标，以像素为单位。
<b>Value</b>	设置显示控件当前位置的值。 默认: 50
<b>Variable</b>	设置要显示的 SPEL+ 的全局变量。 默认: None
<b>Visible</b>	设置显示或隐藏控件。 默认: 1 – True
<b>Width</b>	设置控件的宽度，以像素为单位。

#### 5.15.4 进度条控件事件

事件	描述
<b>Click</b>	单击控件时执行。
<b>DbClick</b>	双击控件时执行。

### 5.16 跟踪栏控件

#### 5.16.1 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。

#### 5.16.2 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

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 EPSEL+的全局变量未在 Variable 属性中更新，请重新创建项目。

#### 5.16.3 跟踪栏控件属性

属性	描述
<b>BackColor</b>	控件的背景颜色。 默认: Control
<b>Enabled</b>	设置控件的边框样式。 默认: 1 – True
<b>EventTaskType</b>	设置用于事件的任务类型。 默认: 0 – Normal
<b>Height</b>	设置控件的高度，以像素为单位。
<b>LargeChange</b>	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
<b>Left</b>	设置控件显示位置的左坐标，以像素为单位。
<b>Maximum</b>	设置控件正在处理的范围的上限。 默认: 10
<b>Minimum</b>	设置控件正在处理的范围的下限。 默认: 0
<b>Name</b>	设置控件的名称。 默认: TrackBarXX
<b>Orientation</b>	设置控件的方向。 默认: 0 – Horizontal
<b>SmallChange</b>	设置使用键盘输入(方向键)时，滑块移动的位置数。 默认: 1
<b>TabIndex</b>	设置控件的选项卡索引。
<b>TickFrequency</b>	设置刻度线之间的位置数。 默认: 1
<b>TickStyle</b>	设置在跟踪栏上显示刻度线的位置。 默认: 2 - BottomRight



属性	描述
<b>ToolTipText</b>	设置控件的工具提示中的显示文本。 默认: 空白
<b>Top</b>	设置控件显示位置的上坐标，以像素为单位。
<b>Value</b>	设置显示控件当前位置的值。 默认: 0
<b>Variable</b>	设置要显示的 SPEL+ 的全局变数。 默认: None
<b>Visible</b>	设置显示或隐藏控件。 默认: 1 – True
<b>Width</b>	设置控件的宽度，以像素为单位。

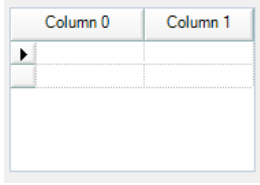
#### 5.16.4 跟踪栏控件事件

事件	描述
<b>Scroll</b>	跟踪栏的滑块移动时，执行滚动。

## 5.17 网格控件

### 5.17.1 描述

网格控件用于以 Excel 格式，显示和编辑数据。



### 5.17.2 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。

### 5.17.3 网格控件属性

属性	描述
<b>AddRow</b>	在执行时添加行。
<b>BorderStyle</b>	设置控件的背景颜色。 默认: Control
<b>CellBackColor</b>	设置或获取单元格的背景颜色。
<b>CellForeColor</b>	设置或获取单元格的 ForeColor。
<b>CellText</b>	设置或获取单元格中的文本。
<b>Enabled</b>	设置是否启用控件。 默认: 1 - True
<b>EventTaskType</b>	设置是否在事件中激活任务类型。 默认: 0 - Normal
<b>GridEditor</b>	打开[GridEditor]对话框，设置网格控件。
<b>Height</b>	以像素为单位，设置控件的高度。
<b>Left</b>	以像素为单位，设置控件显示位置的左侧的坐标。
<b>Name</b>	设置控件名称。 默认: GridXX
<b>RemoveRow</b>	删除行。
属性	描述

<b>RowCount</b>	设置或获取行数。
<b>ScrollBars</b>	显示滚动条时设置。 默认: 0 - None
<b>SelectedIndex</b>	设置或获取选定的行。
<b>Top</b>	以像素为单位设置控件显示位置的上方坐标。
<b>Visible</b>	设置是否显示控件。 默认: 1 - True
<b>Width</b>	以像素为单位，设置控件的宽度。

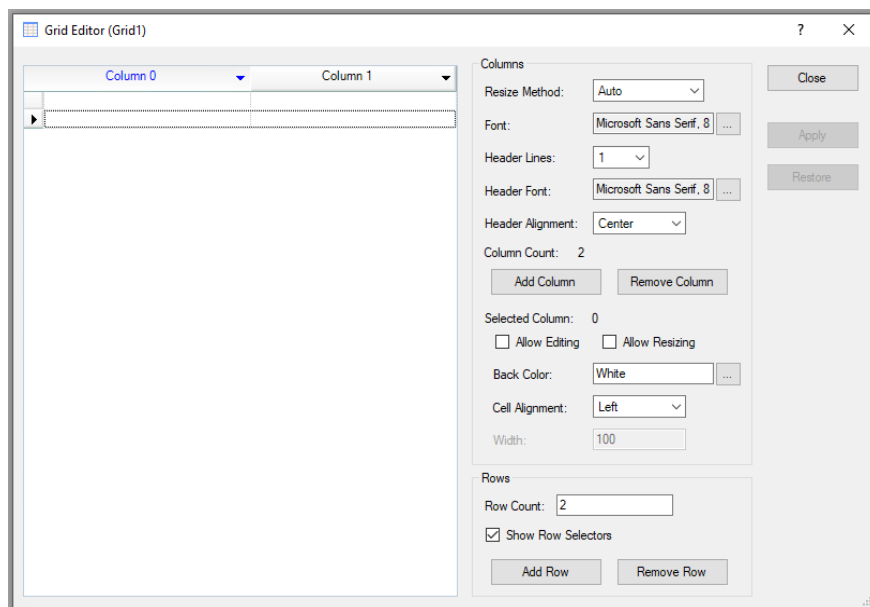
#### 5.17.4 网格控件事件

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

#### 5.17.5 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. 操作

## 6.1 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

## 6.2 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+ 7.0。

### 6.2.1 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。

有关使用窗体和控件的详细信息，请参阅章节窗体和控件的使用。

### 6.2.2 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。

## 6.3 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|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。

## 6.4. 处理暂停与继续

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
End

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

## 6.5. 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
End
Fend
```

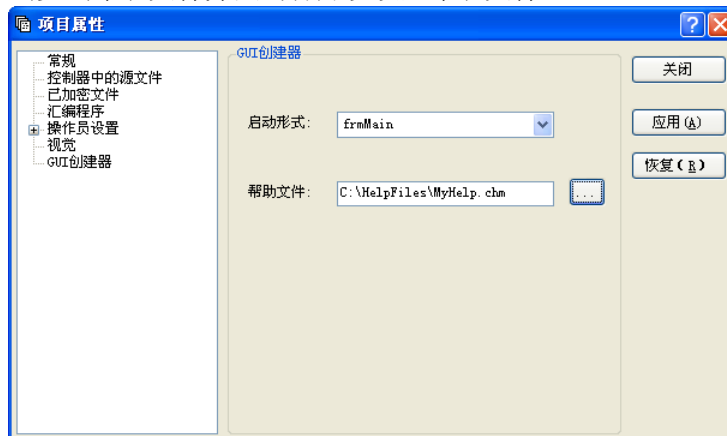
## 6.6 使用帮助文件

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

## 7. GUI Builder 参考

### 7.1 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。

### 7.2 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：

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。（不包括 Button, Label, Checkbox） 如果属性或事件用于窗体，则本节将介绍 Forms 一词。
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。

## 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* 窗体名称或含有窗体名称的字符串变量。

*var* 取得属性值的字符串变量。

*value* 在属性中设定的字符串表达式。

### 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"
```



## 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* 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。控件必须位于指定窗体内。

*value* 在属性中设定的字符串表达式。

### 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"
```

## 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* 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。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
```

## 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* 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。控件必须位于指定窗体内。

*value* 在属性中设定的字符串表达式。

### 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"
```

## 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* 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。  
控件必须位于指定窗体内。

*var* 取得属性值的 Boolean 变量。

*value* 在属性中设定的 Boolean 表达式。

### 数值

0 – False

1 – True

默认：0 – False

### 备注

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
```

## 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* 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* 取得属性值的字符串变量。

*value* 在属性中设定的字符串表达式。

### 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"
```

## 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* 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* 获取属性值的整数变量。

*value* 在属性中设置的整数表达式。

### 数值

0 – VisualStyle.使用系统指定的 BackColor。

1 – User.使用指定的 BackColor。

### 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。

默认：0 – Visual Style

### 另见

BackColor、Button

### 示例

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

## 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* 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* 获取属性值的整数变量。

*value* 在属性中设置的整数表达式。

### 数值

0 - None

1 - FixedSingle

2 - Fixed3D

默认: 0 - None (Label、PictureBox、VideoBox、LED)  
2 - 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
```

## 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* 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* 获取属性值的整数变量。

*value* 在属性中设置的整数表达式。

### 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。若要显示视频，需将 VideoEnabled 属性设为“1 - True”。

### 另见

VideoBox、VideoEnabled

### 示例

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



## 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* 窗体名称或含有窗体名称的字符串变量。

*var* 取得属性值的字符串变量。

*value* 在属性中设定的字符串表达式。

### 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"
```

## 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* 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。

*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* 取得属性值的字符串变量。

*value* 在属性中设定的字符串表达式。

### 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"
```

## 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
```

## 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* 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。

*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* 取得属性值的字符串变量。

*value* 在属性中设定的字符串表达式。

### 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"
```

## 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* 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。

*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* 取得属性值的字符串变量。

*value* 在属性中设定的字符串表达式。

### 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"
```

## 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* 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 **Boolean** 变量。

*value* 在属性中设定的 **Boolean** 表达式。

### 数值

0 - False

1 - True

默认: 0 - 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
```

## 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\$* 发送事件的控件的名称。

### 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。

### 另见

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

### 示例

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

## 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\$ 发送事件的窗体的名称。

### 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
```



## 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* 窗体名称或含有窗体名称的字符串变量。

*var* 取得属性值的 Boolean 变量。

*value* 在属性中设定的 Boolean 表达式。

### 数值

0 - False

1 - True

默认: True

### 备注

如果 **ControlBox** 属性设为 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。

### 另见

Form、MaximizeBox、MinimizeBox

### 示例

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

## 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* 窗体名称或含有窗体名称的字符串变量。

*Index* 含有窗体上控件索引的整数表达式。

*Property* 待访问的控件属性名称。

*var* 将含有属性值的变量。数据类型取决于指定的属性。

*value* 新属性值的表达式。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
```

## 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* 窗体名称或含有窗体名称的字符串变量。

*var* 获取属性值的整数变量。

### 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
```

## DbClick 事件

### 适用范围

Label、ListBox、Led、PictureBox、VideoBox、LED、ProgressBar、Grid

### 描述

双击控件时发生。

### 使用

*Form\_Control\_DbClick* (*Sender\$* As String)

*Sender\$* 发送事件的控件的名称。

### 备注

**DbClick** 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_DbClick(Sender$ As String)
    Integer index

    GGet frmMain.lstModels.SelectedIndex, index
    GGet frmMain.lstModels.List(index), g_CurrModel$
End
```

## 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* 窗体名称或含有窗体名称的字符串变量。

*var* 获取属性值的整数变量。

*value* 在属性中设置的整数表达式。

### 数值

0 – None

1 – OK

2 – Cancel

默认：2 – 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
GClose frmSetSpeed
Fend

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

## Dock 属性

### 适用范围

ProgressBar, StausBar

### 描述

获取或设置状态栏的停放操作。

### 使用

**GGet** *Form.Control.Dock*, *var*

**GSet** *Form.Control.Dock*, *value*

*Form* 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* 获取属性值的整数变量。

*value* 在属性中设置的整数表达式。

### 数值

0 – None

1 – Top

2 – Bottom

3 – Left

4 – Right

5 – Fill

默认: ProgressBar: 0 – None, StatusBar: 2 - Bottom

### 另见

ProgressBar, StatusBar

### 示例

```
GSet frmStatus.ProgressBar1.Dock, DOCK_TOP
```

## 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* 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* 获取属性值的整数变量。

*value* 在属性中设置的整数表达式。

### 数值

0 – Simple

1 – DropDown

2 – DropDownList

默认：1 – DropDown

### 备注

**DropDownStyle** 属性可控制为用户展示的界面。您可以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，或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
```

## Enabled 属性

### 适用范围

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

### 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* 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 **Boolean** 变量。

*value* 在属性中设定的 **Boolean** 表达式。

### 数值

0 – False

1 – True

默认： 1 – True（适用于 Timer 以外）  
0 – False（适用于 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，可将 Enabled 设为 0 – False，使<暂停>按钮变暗。

### 另见

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

### 示例

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



## EventTaskType 属性

### 适用范围

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

### 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* 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* 获取属性值的整数变量。

### 数值

0 – Normal

1 – NoPause

2 – NoEmgAbort

默认: 0 – 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

### 示例

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

## Font 属性

### 适用范围

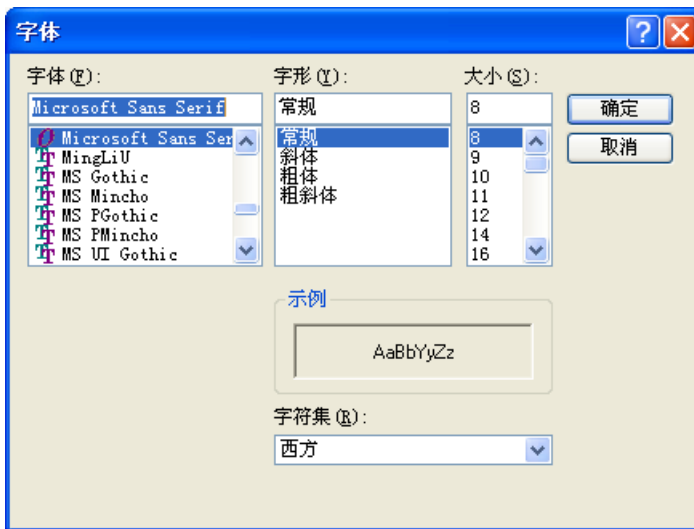
Button、Label、TextBox、RadioButton、CheckBox、ListBox、ComboBox、GroupBox、LED、StatusBar

### 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

在 GUI Builder 窗口属性网格中，选择窗体上的控件并单击 Font 属性值内的按钮。将显示以下对话框。



更改设置并单击<确定>保存。

### 另见

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

## 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* 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 **Boolean** 变量。

*value* 在属性中设定的 **Boolean** 表达式。

### 数值

0 – False

1 – True

默认：0 – False

### 另见

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

### 示例

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

## 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* 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 Boolean 变量。

*value* 在属性中设定的 Boolean 表达式。

### 数值

0 – False

1 – True

默认: 0 – False

### 另见

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

### 示例

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

## 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* 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* 取得属性值的字符串变量。

*value* 在属性中设定的字符串表达式。

### 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"
```

## 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* 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* 将含有属性值的实际变量。

*value* 新属性值的实际表达式。

### 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
```

## 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* 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* 取得属性值的字符串变量。

*value* 在属性中设定的字符串表达式。

### 数值

表示控件前景颜色的字符串。

默认： Control Text (Button、Label、RadioButton、CheckBox、GroupBox、LED)  
Window Text (TextBox、ListBox、ComboBox)

### 另见

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

### 示例

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

## 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* 窗体名称或含有窗体名称的字符串变量。

*var* 获取属性值的整数变量。

*value* 在属性中设置的整数表达式。

### 数值

0 – None

1 – Fixed Single

2 – Fixed3D

3 – FixedDialog

4 – Sizeable

默认: 3 – FixedDialog

### 另见

Form

### 示例

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



## GClose 语句

### 适用范围

Form

### 描述

关闭窗口体。

### 使用

#### **GClose** *Form*

*Form* 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
```

## 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* 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。控件必须位于指定窗体内。

*Property* 待设置或返回数值的属性名称。

*var* 将含有返回数值的变量。

### 备注

GGet 用于在运行时检索属性值。

### 另见

GSet

## 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* 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 Boolean 变量。

*value* 在属性中设定的 Boolean 表达式。

### 数值

0 – False

1 – True

默认：0 – False

### 另见

VideoBox、VideoEnabled

### 示例

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

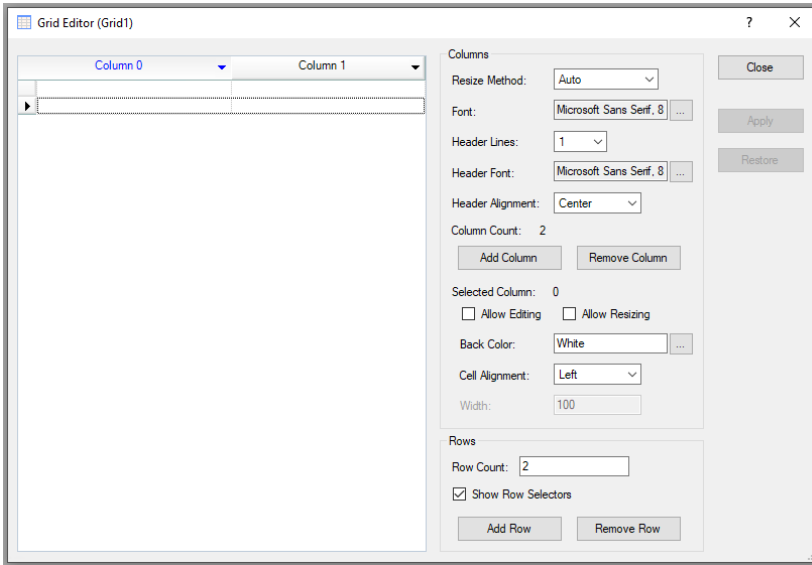
## GridEditor 属性

### 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。 Left: 单元格文本与列的左侧对齐。 Center: 单元格文本在列的居中对齐。 Right: 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。 Left: 列标题与列的左侧对齐。 Center: 列标题在列的居中对齐。 Right: 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

设置	描述
Remove Column	删除选中的列。
Remove Row	删除选中的行。
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。 Auto: 自动调整列的大小以适合网格控件。 Free: 将每列更改为各自的列宽。 Synchronized: 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，然后单击<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。

双击列标题可重新命名列名称。

使用鼠标右键菜单和快捷方式，可以剪切、复制、粘贴、撤销和重试。

*Show Row Selectors* 设置为 True 时，可以选择多行。

### 另见

Grid

## 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* 窗体名称或含有窗体名称的字符串变量。

*Control* 控件名称或含有对象名称的字符串变量。控件必须位于指定窗体内。

*Property* 待设置数值的属性名称。

*value* 含有待设置数据的表达式。

### 备注

GSet 用于在运行时设置窗体或控件属性的值。

### 另见

GGet

## 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* 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
```

## 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* 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
```



## 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* 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
```

## Height 属性

### 适用范围

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

### 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* 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* 获取属性值的整数变量。

*value* 在属性中设置的整数表达式。

### 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

### 示例

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

## 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* 窗体名称或含有窗体名称的字符串变量。

*var* 取得属性值的 Boolean 变量。

*value* 在属性中设定的 Boolean 表达式。

### 数值

0 – False

1 – True

默认：0 – False

### 备注

该属性设为 1 – True 时，<关闭>按钮左侧的标题栏中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
```

## 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* 窗体名称或含有窗体名称的字符串变量。

*var* 获取属性值的整数变量。

*value* 在属性中设置的整数表达式。

### 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
```

## 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* 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* 取得属性值的字符串变量。

*value* 在属性中设定的字符串表达式。

### 数值

含有图像文件完整路径的字符串。

默认：空白

### 注

使用删除键可删除已设置的图像文件。

### 另见

Button、Label、RadioButton、Checkbox、PictureBox、ImageAlign

### 示例

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

## 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* 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* 获取属性值的整数变量。

*value* 在属性中设置的整数表达式。

### 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
```

## 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* 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* 取得属性值的字符串变量。

*value* 在属性中设定的字符串表达式。

### 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、wmf 或 png。

默认: LedOff.ico

### 另见

LED、ImageOn

### 示例

```
GSet frmMain.Led1.ImageOff, "c:\EpsonRC70\GUI\Icons\LedOff.ico"
```

## 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* 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* 取得属性值的字符串变量。

*value* 在属性中设定的字符串表达式。

### 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、wmf 或 png。

默认: LedRed.ico

### 另见

LED、ImageOff

### 示例

```
GSet frmMain.Led1.ImageOn, "c:\EpsonRC70\GUI\Icons\LedRed.ico"
```



## 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* 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* 获取属性值的整数变量。

*value* 在属性中设置的整数表达式。

### 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
```

## 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* 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* 获取属性值的整数变量。

*value* 在属性中设置的整数表达式。

### 数值

整数值范围为 0–9999。

默认：0

### 另见

LED、IOType

### 示例

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

## 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* 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* 获取属性值的整数变量。

*value* 在属性中设置的整数表达式。

### 数值

0 – Input

1 – Output

2 – Memory

默认: 0 – Input

### 另见

LED、IOBit

### 示例

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

## 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* 发送事件的控件的名称。

*Key* 操作员按下的按键。

### 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
```

## 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* 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* 获取属性值的整数变量。

*value* 在属性中设置的整数表达式。

### 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
```

## Left 属性

### 适用范围

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

### 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* 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* 获取属性值的整数变量。

*value* 在属性中设置的整数表达式。

### 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

### 示例

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

## 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* 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。控件必须位于指定窗体内。

*index* 以 0 开始的列表数组索引的整数表达式。

*var* 获取属性值的整数变量。

*value* 在属性中设定的字符串表达式。

### 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$
```

## 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* 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。控件必须位于指定窗体内。

*var* 获取属性值的整数变量。

*value* 在属性中设定的字符串表达式。

### 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
```



## 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* 发送事件的窗体的名称。

### 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
```

## 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* 窗体名称或含有窗体名称的字符串变量。

*var* 取得属性值的 Boolean 变量。

*value* 在属性中设定的 Boolean 表达式。

### 数值

0 – False

1 – True

默认：0 – 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
```

## 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* 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。控件必须位于指定窗体内。

*var* 取得属性值的整数变量。

*value* 在属性中设定的字符串表达式。

### 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
```

## 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* 窗体名称或含有窗体名称的字符串变量。

*var* 取得属性值的 Boolean 变量。

*value* 在属性中设定的 Boolean 表达式。

### 数值

0 – False

1 – True

默认：0 – 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
```

## 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* 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。控件必须位于指定窗体内。

*var* 取得属性值的整数变量。

*value* 在属性中设定的字符串表达式。

### 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
```

## 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* 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。控件必须位于指定窗体内。

*var* 取得属性值的 Boolean 变量。

*value* 在属性中设定的 Boolean 表达式。

### 数值

0 – False

1 – True

默认: 0 – 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 1 – True。

### 另见

TextBox、ScrollBars

### 示例

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

## Name 属性

### 适用范围

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

### 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* 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

### 示例

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

## 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* 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* 取得属性值的字符串变量。

*value* 在属性中设定的字符串表达式。

### 数值

0 – Horizontal

1 – Vertical

默认: 0 – Horizontal

### 另见

ProgressBar, TrackBar

### 示例

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



## 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* 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。控件必须位于指定窗体内。

*var* 取得属性值的字符串变量。

*value* 在属性中设定的字符串表达式。

### 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**。注意 TextBox 控件的 **Multiline** 属性必须设为 0-False。

### 另见

TextBox

### 示例

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

## 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* 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* 取得属性值的字符串变量。

*value* 在属性中设定的字符串表达式。

### 数值

0 – Blocks

1 – Continuous

2 – Marquee

默认: 0 - Blocks

### 另见

ProgressBar

### 示例

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

## 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* 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。  
控件必须位于指定窗体内。

*var* 取得属性值的 Boolean 变量。

*value* 在属性中设定的 Boolean 表达式。

### 数值

0 – False

1 – True

默认: 0 – False

### 备注

该属性设为 1-True 时, 操作员将无法编辑文本框。

### 另见

TextBox

### 示例

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

## 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* 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。  
控件必须位于指定窗体内。

*value* 代表要删除的行索引的整数值。

### 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
```

## 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* \$ 发送事件的窗体的名称。

### 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
```

## RobotNumbe 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* 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。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
```

## 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* 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。  
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
```

## 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\$* 设置导致事件发生的控件的名称。

### 另见

TrackBar

### 示例

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



## 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* 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。控件必须位于指定窗体内。

*var* 获取属性值的整数变量。

*value* 在属性中设置的整数表达式。

### 数值

0 – None

1 – Horizontal

2 – Vertical

3 – Both

默认：0 – None

### 备注

若要使用 ScrollBars，文本框控件的 Multiline 属性必须设为 1 – True。

若要显示水平滚动条，需将 WordWrap 属性设为 0 – False。

### 另见

TextBox、Multiline、WordWrap

### 示例

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

## SelectedIndex 属性

### 适用范围

ListBox、ComboBox、Grid

### 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* 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。控件必须位于指定窗体内。

*var* 获取属性值的整数变量。

*value* 在属性中设置的整数表达式。

### 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
```

## 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* 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。  
控件必须位于指定窗体内。

*var* 取得属性值的 Boolean 变量。

*value* 在属性中设定的 Boolean 表达式。

### 数值

0 – False

1 – True

默认: 0 – False

### 另见

StatusBar, ShowEStop, ShowRobot, ShowSafeguard

### 示例

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

## 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* 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。  
控件必须位于指定窗体内。

*var* 取得属性值的 Boolean 变量。

*value* 在属性中设定的 Boolean 表达式。

### 数值

0 – False

1 – True

默认: 0 – False

### 另见

StatusBar, ShowDateTime, ShowRobot, ShowSafeguard

### 示例

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

## 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* 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。控件必须位于指定窗体内。

*var* 获取属性值的整数变量。

*value* 在属性中设置的整数表达式。

### 数值

0 – False

1 – True

默认: 0 – 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
```

## 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* 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。  
控件必须位于指定窗体内。

*var* 取得属性值的 Boolean 变量。

*value* 在属性中设定的 Boolean 表达式。

### 数值

0 – False

1 – True

默认: 0 – False

### 另见

StatusBar, RobotNumber, ShowDateTime, ShowEStop, ShowSafeguard

### 示例

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

## 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* 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。  
控件必须位于指定窗体内。

*var* 取得属性值的 **Boolean** 变量。

*value* 在属性中设定的 **Boolean** 表达式。

### 数值

0 – False

1 – True

默认: 0 – False

### 另见

StatusBar, ShowDateTime, ShowEStop, ShowRobot

### 示例

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

## 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* 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。控件必须位于指定窗体内。

*var* 获取属性值的整数变量。

*value* 在属性中设置的整数表达式。

### 数值

0 – Normal

1 – StretchImage

2 – AutoSize

3 – CenterImage

4 – Zoom

默认：0 – 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
```



## 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* 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。控件必须位于指定窗体内。

*var* 获取属性值的整数变量。

*value* 在属性中设置的整数表达式。

### 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
```

## 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* 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。控件必须位于指定窗体内。

*var* 取得属性值的 **Boolean** 变量。

*value* 在属性中设定的 **Boolean** 表达式。

### 数值

0 – False

1 – True

默认：0 – False

### 另见

ComboBox、ListBox

### 示例

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

## 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* 窗体名称或含有窗体名称的字符串变量。

*var* 获取属性值的整数变量。

*value* 在属性中设置的整数表达式。

### 数值

0 – Manual

1 – CenterScreen

2 – CenterParent

默认：1 – CenterScreen

### 另见

Form、WindowState

### 示例

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

## Step 属性

### 适用范围

ProgressBar

### 描述

获取或设置进度条控件的当前值增加的数量。

### 使用

**GGet** *Form.Control.Step, var*

**GSet** *Form.Control.Step, value*

*Form* 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* 获取属性值的整数变量。

*value* 在属性中设置的整数表达式。

### 数值

一个整数值，用于显示当前值增加的数量。

### 另见

ProgressBar

### 示例

```
GSet frmMain.ProgressBar1.Step, 5
```

## 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.TabIndex, var*

**GSet** *Form.Control.TabIndex, value*

*Form* 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* 获取属性值的整数变量。

*value* 在属性中设置的整数表达式。

### 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
```

## 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* 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* 取得属性值的字符串变量。

*value* 在属性中设定的字符串表达式。

### 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: "
```

## 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* 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* 获取属性值的整数变量。

*value* 在属性中设置的整数表达式。

### 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)

#### 对于文本框

1 – Left

2 – Center

3 – Right

默认: 1 – Left

### 另见

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

### 示例

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

## 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\$ 发送事件的控件的名称。

### 另见

Timer、Interval、Enabled

### 示例

```
Function frmMain_Timer1_Tick(Sender$ As String)

    GSet frmMain.lblDateTime.Text, Date$ + " " + Time$
End
```



## 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* 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* 获取属性值的整数变量。

*value* 在属性中设置的整数表达式。

### 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
```

## 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* 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* 获取属性值的整数变量。

*value* 在属性中设置的整数表达式。

### 数值

0 – None

1 – TopLeft

2 – BottomRight

3 – Both

默认: 2 - BottomRight

### 另见

TrackBar

### 示例

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

## ToolTipText 属性

### 适用范围

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

### 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* 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* 取得属性值的字符串变量。

*value* 在属性中设定的字符串表达式。

### 数值

含有控件工具提示文本的字符串。

默认：空字符串

### 另见

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

### 示例

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

## Top 属性

### 适用范围

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

### 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* 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* 获取属性值的整数变量。

*value* 在属性中设置的整数表达式。

### 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

### 示例

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

## 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* 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$
```

## 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* 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 Boolean 变量。

*value* 在属性中设定的 Boolean 表达式。

### 数值

0 – False

1 – True

默认: 1 – True

### 备注

若要在文本框上显示 SPEL+ 全局变量，设置 Variable 属性中的全局变量名称。如果 Update 属性设为 0–False，则不会更新全局变量值。

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 GSset 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
```

## 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* 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。控件必须位于指定窗体内。

*var* 取得属性值的字符串变量。

*value* 在属性中设定的字符串表达式。

### 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
```

## 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+ 全局变量名称。

### 使用

**GGet** *Form.Control.Variable, var*

**GSet** *Form.Control.Variable, value*

*Form* 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。控件必须位于指定窗体内。

*var* 取得属性值的字符串变量。

*value* 在属性中设定的字符串表达式。

### 数值

当前项目中 SPEL+ 全局变量的名称。

默认: None

### 备注

当 Variable 属性设为 SPEL+ 全局变量时，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 GSset 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$"
```



## 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* 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 Boolean 变量。

*value* 在属性中设定的 Boolean 表达式。

### 数值

0 – False

1 – True

默认：0 – False

### 另见

VideoBox、GraphicsEnabled

### 示例

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

## Visible 属性

### 适用范围

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

### 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* 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 Boolean 变量。

*value* 在属性中设定的 Boolean 表达式。

### 数值

1 – True 显示控件

0 – False 不显示控件

默认：1 – 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

### 示例

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

## Width 属性

### 适用范围

Form、Button、Label、TextBox、RadioButton、CheckBox、ListBox、ComboBox、PictureBox、GroupBox1、VideoBox、LED、ProgressBar、TrackBar、StatusBar、Grid

### 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* 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* 获取属性值的整数变量。

*value* 在属性中设置的整数表达式。

### 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

### 示例

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

## 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* 窗体名称或含有窗体名称的字符串变量。

*var* 获取属性值的整数变量。

*value* 在属性中设置的整数表达式。

### 数值

0 – Normal

1 – Minimized

2 – Maximized

默认: 0 – Normal

### 另见

Form

### 示例

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

## 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* 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。控件必须位于指定窗体内。

*var* 取得属性值的 **Boolean** 变量。

*value* 在属性中设定的 **Boolean** 表达式。

### 数值

1 – **True** : 多行文本框控件换行

0 – **False** : 当用户的输入超过控件右侧边缘时, 文本框控件自动水平滚动。

默认: 1 – **True**

### 另见

TextBox、Multiline、ScrollBars

### 示例

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

