

新版软件参考手册系列

PowerBuilder

6.0~6.5参考手册

(控件篇)

新世纪工作室 编著

62
/1

国防工业出版社

TP311.56-62

X.21

新版软件参考手册系列

PowerBuilder 6.0~6.5 参考手册

(控件篇)

新世纪工作室 编著



国防工业出版社

北京
053880

图书在版编目(CIP)数据

Power Builder 6.0~6.5 参考手册. 控件篇/新世纪
工作室编著. —北京:国防工业出版社,1999.8
(新版软件参考手册系列)
ISBN 7-118-02097-4

I . P… II . 新… III . 数据库管理系统-软件工具,
Power Builder 6.0~6.5 IV . TP311.56

中国版本图书馆 CIP 数据核字(1999)第 45970 号

JS336/22

国防工业出版社出版发行

(北京市海淀区紫竹院南路 23 号)

(邮政编码 100044)

北京怀柔新华印刷厂印刷

新华书店经售

*

开本 787×1092 1/16 印张 27¼ 637 千字

1999 年 8 月第 1 版 1999 年 8 月第 1 次印刷

印数:1-4000 册 定价:36.00 元

(本书如有印装错误,我社负责调换)

序

计算机软件的应用基本上可以分成两个层次。对于纯操作型软件,如 Word、Front Page、WPS 等,通常针对非专业计算机用户。有了这些软件,只需经过有限步骤的操作,就可以直接满足某些应用需求。而另一类软件,如 Visual Basic、Visual C++、Power-Builder 等,则要复杂得多。这些软件通常是供专业计算机用户(程序员)使用的,不仅要熟悉软件的环境与操作,还要掌握软件为应用开发所提供的编程元素,如特定的编程语言、内置的资源 and 工具等,这样才可能在其基础上开发出可以实现最终目标的应用系统。所以,对于后一类软件,专业用户或希望成为专业用户的初学者往往要投入大量时间和精力来查阅和尝试软件语言、资源以及工具的用法。

另外,软件更新换代的速度很快,新版软件总是会修订旧版软件的功能,并增添新的功能。为此,专业计算机用户还需要跟踪最新软件技术,并在很短时间内掌握最新软件工具的使用。

因此,根据这种需要,这类软件的相关书籍必须具备下列几个特点:

(1) 基于最新版本。新版软件往往兼容相应的旧版软件,因此新版软件的相关书籍也基本适用于旧版本。

(2) 内容全面。特定的应用开发往往涉及软件提供的各种资源,因此要求专业计算机用户对软件的掌握必须全面、深入。

(3) 结构合理,便于查阅。随着软件规模不断扩大,其复杂性也不断提高,合理的组织与编排可以为专业计算机用户节省大量时间。

目前市面上虽然出版了不少技术参考读物,但其内容基本上来源于相应的英文书籍或者软件的帮助文本。一方面,译者的英语水平和软件的掌握程度对书籍的质量影响极大;另一方面,对零碎的帮助文本组织得是否合理也是面临的一大问题。如果再考虑到软件的更新,以致质量很难有保证。因此,目前这类书籍的现状是质量参差不齐,上乘者极少。

针对这种情况,本着为软件开发和应用人员服务的原则,我们组织了一批经验丰富的软件工程师,编写了《新版软件参考手册系列》丛书。

这套丛书针对目前最新版本的流行软件,以软件开发环境、资源和工具为基础,主要以专业软件开发人员为读者对象,所涉及的软件基本包括 PowerBuilder 6.5、SQL Server 7.0、Novell Netware 5.0、Visual Basic 6.0 和 Visual C++ 6.0。这套丛书列举并举例说明了这些软件开发工具的组成元素,各条目间相关性小,但自成一体。全书强调全面、详尽、便于查阅,因为它是供专业软件开发人员在实际开发过程中参考查阅用的,非专业人员可能不需要这套丛书,但对于专业开发者,它却是必不可少的。

我们相信,丛书的出版必将为广大读者开辟跟踪、掌握、运用、进而创造新技术的最佳途径。

当然,尽管我们强调以质取胜,宁缺勿滥,但是,由于相应软件实在极其复杂,加上时间较紧,因此书中可能还难免有一些错误,请各位读者不吝指正。

前 言

PowerBuilder 是 Sybase 公司的子公司 PowerSoft 推出的一个用于设计企业级客户/服务器应用程序的开发工具,它支持多种平台,具备跨平台开发应用程序的能力。本书是 PowerBuilder 的控件参考手册,详细介绍了每个控件的属性、事件和函数。

控件是构造 PowerBuilder 应用程序界面的基本部件,应用程序利用控件完成输入、输出、装饰等任务。控件可分成以下几类:

(1) 编辑控件:编辑控件用于输入、编辑数据。PowerBuilder 提供了 5 种编辑控件,分别是单行编辑框、多行编辑框、超文本编辑框、编辑掩码控件和数据窗口控件。

(2) 显示输出控件:显示输出控件用于显示输出文本、统计图、图像。这类控件包括静态文本控件、图片框控件、统计图控件、用户对象、OLE 控件等。

(3) 选择操作控件:选择操作控件用于让用户从多种选项中选择一项或多项。这类控件包括单选钮、复选框、列表框、下拉列表框、下拉图片列表框、列表视图、树形视图、标签控件等。

(4) 装饰控件:装饰控件用于美化用户界面,将程序功能分组。这类控件包括直线控件、椭圆控件、矩形控件、圆角矩形控件、组框等。

(5) 按钮控件:按钮控件用于接收用户的操作,以完成程序指定的功能。这类控件包括命令按钮、图片按钮、水平滚动条、垂直滚动条等。

上述分类方法只是按主要功能对控件进行的粗略划分,实际上,有很多控件同时属于多种类型。具体地说,PowerBuilder 中的控件包括 27 种:单行编辑框、多行编辑框、编辑掩码控件、超文本编辑框、静态文本控件、图片框控件、统计图控件、用户对象控件、OLE 控件、单选钮、复选框、列表框、下拉列表框、下拉图片列表框、列表视图控件、树形视图控件、标签控件、直线控件、椭圆控件、矩形控件、圆角矩形控件、组框控件、命令按钮、图片按钮、水平滚动条控件、垂直滚动条控件、数据窗口控件。这些控件基本上有三方面的特征:属性、事件和函数。本书将简要介绍各控件属性的含义和事件的触发时机,并详细介绍各控件的函数,同时给出了可直接在应用程序中运行的大量示例,供用户编写应用程序时参考。另外,对于功能、语法、用法完全相同的函数,在附录中单独介绍,而对存在某些细微区别的函数,依然作为单独条目在各控件中介绍。

本书适合 PowerBuilder 开发人员阅读,是开发 PowerBuilder 应用程序不可缺少的参考书。如果与本书的姊妹篇《PowerBuilder 6.0~6.5 参考手册(语言、函数、对象篇)》搭配使用,则效果会更好。

本书第一章到第十章由裘实、边天正、刘力、徐洁编写;第十一章到第十七章由刘秀芳、刘松海、吴清、李梅、王小兰编写;第十八章到第二十七章由张启忠、魏宁、赵亦农、刘勇、王海宏编写,对以上同志表示感谢。

目 录

第一章 单行编辑框	1	第九章 OLE 控件	121
1.1 单行编辑框控件的属性	1	9.1 OLE 控件的属性	121
1.2 单行编辑框控件的事件	3	9.2 OLE 控件的事件	123
1.3 单行编辑框控件的函数	3	9.3 OLE 控件的函数	124
第二章 多行编辑框	10	第十章 单选钮	138
2.1 多行编辑框的属性	10	10.1 单项钮的属性	138
2.2 多行编辑框的事件	12	10.2 单选钮的事件	140
2.3 多行编辑框的函数	12	10.3 单选钮的函数	140
第三章 编辑掩码控件	21	第十一章 复选框	142
3.1 编辑掩码控件的属性	21	11.1 复选框的属性	142
3.2 编辑掩码控件的事件	24	11.2 复选框的事件	144
3.3 编辑掩码控件的函数	24	11.3 复选框的函数	145
第四章 超文本编辑框	33	第十二章 列表框	146
4.1 超文本编辑框的属性	33	12.1 列表框的属性	146
4.2 超文本编辑框的事件	36	12.2 列表框的事件	148
4.3 超文本编辑框的函数	37	12.3 列表框的函数	149
第五章 静态文本控件	70	第十三章 下拉列表框	160
5.1 静态文本控件的属性	70	13.1 下拉列表框的属性	160
5.2 静态文本控件的事件	72	13.2 下拉列表框的事件	162
5.3 静态文本控件的函数	72	13.3 下拉列表框的函数	163
第六章 图片框控件	74	第十四章 下拉图片列表框	174
6.1 图片框控件的属性	74	14.1 下拉图片列表框的属性	174
6.2 图片框控件的事件	75	14.2 下拉图片列表框的事件	177
6.3 图片框控件的函数	76	14.3 下拉图片列表框的函数	177
第七章 统计图控件	80	第十五章 列表视图控件	190
7.1 统计图控件的属性	80	15.1 列表视图控件的属性	190
7.2 统计图控件的事件	82	15.2 列表视图控件的事件	193
7.3 统计图控件的函数	82	15.3 列表视图控件的函数	195
第八章 用户对象控件	117	第十六章 树形视图控件	214
8.1 用户对象控件的属性	117	16.1 树形视图控件的属性	214
8.2 用户对象控件的事件	119	16.2 树形视图控件的事件	217
8.3 用户对象控件的函数	119	16.3 树形视图控件的函数	218

第十七章 标签控件	238	23.1 命令按钮的属性	260
17.1 标签控件的属性	238	23.2 命令按钮的事件	261
17.2 标签控件的事件	241	23.3 命令按钮的函数	262
17.3 标签控件的函数	242	第二十四章 图片按钮	263
第十八章 直线控件	249	24.1 图片按钮的属性	263
18.1 直线控件的属性	249	24.2 图片按钮的事件	265
18.2 直线控件的函数	250	24.3 图片按钮的函数	265
第十九章 椭圆控件	251	第二十五章 水平滚动条控件	266
19.1 椭圆控件的属性	251	25.1 水平滚动条控件的属性	266
19.2 椭圆控件的函数	252	25.2 水平滚动条控件的事件	267
第二十章 矩形控件	253	25.3 水平滚动条控件的函数	268
20.1 矩形控件的属性	253	第二十六章 垂直滚动条控件	269
20.2 矩形控件的函数	254	26.1 垂直滚动条控件的属性	269
第二十一章 圆角矩形控件	255	26.2 垂直滚动条控件的事件	270
21.1 圆角矩形控件的属性	255	26.3 垂直滚动条控件的函数	271
21.2 圆角矩形控件的函数	256	第二十七章 数据窗口控件	272
第二十二章 组框控件	257	27.1 数据窗口控件的属性	272
22.1 组框控件的属性	257	27.2 数据窗口控件的事件	274
22.2 组框控件的函数	259	27.3 数据窗口控件的函数	276
第二十三章 命令按钮	260	附录 控件的公共函数	414

第一章 单行编辑框

单行编辑框(SingleLineEdit)控件可以输入、编辑、显示一行文本,通常用于输入较少的数据,比如让用户输入口令等。

1.1 单行编辑框控件的属性

单行编辑框控件的属性如表 1.1 所列。

表 1.1 单行编辑框控件的属性

属性	数据类型	意 义
Accelerator	Integer	指定该控件的快捷键,使用相应键的 ASCII 码表示
AutoHScroll	Boolean	指定随着数据的输入或删除该控件是否自动水平滚动数据。有效取值为: True——自动水平滚动;False——不自动水平滚动
BackColor	Long	指定该控件的背景颜色,可能取值在 -2 到 16 777 215 之间
Border	Boolean	指定该控件是否有边框。有效取值为: True——有边框;False——无边框
BorderStyle	BorderStyle	指定该控件的边框风格,有效取值为: StyleBox!、StyleLowered!、StyleRaised!、StyleShadowBox!
BringToTop	Boolean	指定 PowerBuilder 是否把该控件移动到其他控件的前面。有效取值为: True——移动;False——不移动
DisplayOnly	Boolean	指定该控件是否处于只读方式,处于只读方式时,用户不能修改该控件中的文本。有效取值为: True——只读方式;False——用户可以修改数据
DragAuto	Boolean	指定 PowerBuilder 是否自动把该控件置入拖曳模式(Drag Mode)。有效取值为: True——当该控件被单击时,该控件自动进入拖曳模式;False——当该控件被单击时,该控件不自动进入拖曳模式。此时,如果需要让该控件进入拖曳模式,需要在代码中使用 Drag() 函数手工设置
DragIcon	String	指定拖曳图标的名称,该名称可以是系统预定义图标名(比如 exclamation!、information! 等),也可以是图标文件名(.ICO)。缺省时系统使用控件大小的方框作为拖曳图标。在用户拖动控件过程中,当被拖动控件进入该控件可以被放下的区域时,系统显示拖动图标
Enabled	Boolean	指定该控件是否能用(即用户是否能操作),有效取值为: True——用户能够操作该控件;False——用户不能够操作该控件
FaceName	String	指定字体名称,该控件使用指定的字体显示文字,例如 ARIAL、COURIER、宋体、楷体等

(续)

属性	数据类型	意 义
FontCharSet	FontCharSet (枚举)	指定显示文字的字体字符集。有效取值为:ANSI!、ChineseBig5!、Default-CharSet!、Hangeul!、OEM!、ShiftJIS!、Symbol!
FontFamily	FontFamily (枚举)	指定显示文字的字体风格。有效取值为:AnyFont!、Decorative!、Modern!、Roman!、Script!、Swiss!
FontPitch	FontPitch (枚举)	指定字体间距方式,有效取值为:Default!、Fixed!、Variable!
Height	Integer	指定该控件的高度(以 PBU 为单位)
HideSelection	Boolean	指定当输入焦点不在该控件上时,选中文本是否依然保持加亮显示。有效取值为:True——不加亮显示;False——加亮显示
Italic	Boolean	指定控件上显示的文字是否使用斜体。有效取值为:True——斜体文字;False——非斜体文字
Limit	Integer	指定该控件中能够输入的最多字符个数,其值在 0 到 32 767 之间,其中 0 表示没有个数限制
Password	Boolean	指定该单行编辑框是否用于保密字输入。有效取值为:True——用于保密字输入,此时,用户输入的所有字符均显示为星号(*),星号个数就是用户输入的字符个数;False——不用于保密字输入,此时,按用户的输入显示文字
Pointer	String	指定当用户把鼠标指针移动到该控件上时系统显示的指针图像。既可以把系统预定义的鼠标指针名(Arrow!、Beam!、Cross!、HourGlass!、SizeNESW!、SizeNS!、SizeNWSE!、SizeWE!、UpArrow!)赋给该属性,也可以指定一个位图作为鼠标指针。缺省值为 Arrow!
RightToLeft	Boolean	指定该控件上显示的文本是否采用自右向左方式。要使该属性起作用,不仅需要操作系统支持自右向左显示方式,而且应用程序有 Hebrew 或 Arabic 版本的 PowerBuilder 动态库支持。有效取值为:True——文字采用自右向左显示方式;False——文字采用自左向右显示方式
TabOrder	Integer	指定该控件在窗口中的跳转次序(即按 Tab 键移动焦点的顺序)值(0 表示用户不能通过按 Tab 键把输入焦点移动到该控件上(但依然可以使用鼠标进行操作))
Tag	String	指定与该控件相关联的任意字符串,字符串的意义由应用程序确定(最简单的用法是用该属性保存微帮助信息)
Text	String	指定显示在该控件中的文本,同时用户输入的文本也保存在该属性中
TextCase	TextCase	指定以何种方式显示用户的输入。有效取值为:AnyCase! ——输入什么显示什么;Lower! ——以小写格式显示;Upper! ——以大写格式显示
TextColor	Long	指定文字的显示颜色。有效值在 -2 到 16 777 215 之间
TextSize	Integer	指定显示在该控件上的文字的大小(以点为单位)。为了向前兼容,大小值采用负值表示,例如,10 点大小时该属性保存为 -10

(续)

属性	数据类型	意 义
Underline	Boolean	指定显示文字上是否加下划线。有效取值为: True——加下划线; False——不加下划线
Visible	Boolean	指定该控件是否可见(即是否显示在窗口上),有效取值为: True——该控件可见; False——该控件不可见
Weight	Integer	指定显示文字的磅值,例如,400 为正常值,700 为粗体
Width	Integer	指定该控件的宽度(以 PBU 为单位)
X	Integer	指定控件在窗口上的 X 位置(即离窗口左边缘的距离),以 PBU 为单位
Y	Integer	指定控件在窗口上的 Y 位置(即离窗口顶部的距离),以 PBU 为单位

1.2 单行编辑框控件的事件

单行编辑框控件的事件及触发时机如表 1.2 所列。

表 1.2 单行编辑框控件的事件及触发时机

事件	触 发 时 机
Constructor	在窗口的 Open 事件发生之前触发
Destructor	在窗口的 Close 事件发生之后触发
DragDrop	当某个被拖曳控件放置到该控件上时触发
DragEnter	当某个被拖曳控件进入该控件时触发
DragLeave	当某个被拖曳控件离开该控件时触发
DragWithin	当某个被拖曳控件处于该控件内部时触发
GetFocus	在该控件得到输入焦点前触发(这个事件发生在该控件被选中并变为活动控件之前)
LoseFocus	当该控件失去输入焦点(变为不活动时)时触发
Modified	当该控件的内容被修改并失去焦点时触发
Other	当发生了某个 Windows 消息,而该消息又不对应于该控件的某个 PowerBuilder 事件时触发
RbuttonDown	当用户右击该控件时触发

1.3 单行编辑框控件的函数

单行编辑框的对象函数如表 1.3 所列。

表 1.3 单行编辑框的对象函数

函数	返回值	功 能
CanUndo	Boolean	检测最近一次编辑操作能否被撤消。返回值为 True 时能够撤消,为 False 时不能撤消
ClassName	String	返回单行编辑框的名称
Clear	Integer	清除该控件中所选文本(但并不将其放置到剪贴板上)
Copy	Integer	将该控件中所选文本复制(而不是删除)到系统剪贴板上
Cut	Integer	将该控件中所选文本剪贴到系统剪贴板上
Drag	Integer	开始或结束单行编辑框的拖动
GetContextService	Integer	创建指定服务(包括 ContextInformation、Internet 以及 Keyword)的上下文相关实例
GetParent	PowerObject	返回父对象的名称
Hide	Integer	隐藏单行编辑框
Move	Integer	将该控件移动到指定位置
Paste	Integer	将系统剪贴板的内容插入到该控件中插入点所在位置
PointerX	Integer	返回当前鼠标指针离该控件左边缘的距离
PointerY	Integer	返回当前鼠标指针离该控件上边缘的距离
Position	Integer	返回控件中插入点的位置
PostEvent	Boolean	将某个事件添加到单行编辑框控件消息队列的尾部
Print	Integer	打印单行编辑框
ReplaceText	Integer	用指定的字符串替换单行编辑框中选定的文本。如果没有文本被选定,则 ReplaceText 将指定字符串插入到插入点位置
Resize	Integer	修改单行编辑框的大小
SelectedLength	Integer	返回单行编辑框中所选文本的长度
SelectedStart	Integer	返回单行编辑框中所选文本的起始位置
SelectedText	String	返回单行编辑框中所选文本
SelectText	Integer	在单行编辑框中选某段文本
SetFocus	Integer	将焦点设置在该控件上
SetPosition	Integer	指定该控件在窗口上的前后次序
SetRedraw	Integer	控制在每次修改该控件的属性后是否自动重绘该控件
Show	Integer	显示该控件
TriggerEvent	Integer	触发单行编辑框的指定事件并执行该事件的事件处理程序
TypeOf	Object	返回该控件的类型
Undo	Integer	取消用户对单行编辑框的前一次编辑,恢复编辑前的内容

其中,ClassName()、Drag()、GetContextService()、GetParent()、Hide()、Move()、PointerX()、PointerY()、PostEvent()、Print()、Resize()、SetFocus()、SetPosition()、SetRedraw()、Show()、TriggerEvent()、TypeOf()函数的意义与其他控件同名函数相同,请参阅附录。下面介绍该控件特有的函数。

1. CanUndo()

功能

检测最近的编辑操作能否被撤消。如果能够使用 Undo()函数取消用户在该控件中的上次操作时,CanUndo()函数返回 TRUE,否则返回 FALSE。

语法

singlelineeditname.CanUndo()

参数

singlelineeditname:单行编辑框名称。

返回值

Boolean。如果最近一次操作可以被取消,该函数返回 TRUE,否则返回 FALSE。如果 singlelineeditname 为 NULL,则 CanUndo()函数返回 NULL。

2. Clear()

功能

删除该控件中所选内容。

语法

singlelineeditname.Clear()

参数

singlelineeditname:单行编辑框名称。

返回值

Integer。函数执行成功时返回删除的字符个数。如果在单行编辑框中未选择任何内容,则不删除任何字符,函数返回 0。如果函数执行出错,则返回 -1。如果 singlelineeditname 的值为 NULL,则 Clear()函数返回 NULL。

用法

除了程序中使用 Clear()函数删除单行编辑框中所选内容外,用户使用键盘(按 Del 键)或鼠标同样可以删除所选内容。代码中可以使用该控件的对象函数 SelectText()选择某段文本。

如果希望删除所选文本并将被删文本放置到系统剪贴板上,那么请使用该控件的对象函数 Cut()。

3. Copy()

功能

将该控件中所选文本复制到系统剪贴板上。Copy()函数并不删除所选文本。

语法

singlelineeditname.Copy ()

参数

singlelineeditname: 单行编辑框名称。

返回值

integer。函数执行成功时返回复制到剪贴板的字符个数。如果在单行编辑框中未选择任何内容,则不复制任何字符,函数返回 0。如果函数执行出错,则返回 -1。

4. Cut()

功能

将该控件中所选文本剪切到系统剪贴板上。

语法

singlelineeditname.Cut ()

参数

singlelineeditname: 单行编辑框名称。

返回值

Integer。函数执行成功时返回剪切到剪贴板的字符个数。如果在该控件中未选择任何内容,则不剪切任何字符,函数返回 0。如果函数执行出错,则返回 -1。

5. Paste()

功能

将系统剪贴板的内容插入到该控件中光标所在位置。

语法

singlelineeditname.Paste ()

参数

singlelineeditname: 单行编辑框名称。

返回值

Integer。函数执行成功时,返回粘贴到该控件的字符个数。如果 singlelineeditname 的值为 NULL,则 Paste()函数返回 NULL。

6. Position()

功能

返回该控件中插入点的位置。

语法

singlelineeditname.Position()

参数

singlelineeditname: 单行编辑框名称。

返回值

Integer。函数执行成功时,返回插入点位置,函数执行出错时返回 -1。如果 singlelineeditname 的值为 NULL,则 Position()函数返回 NULL。

用法

Position()返回插入点后第一个字符的位置,例如,如果光标位于编辑框的起始位置,则 Position()函数返回 1。如果某段文本被选中,则 Position()函数返回所选文本第一个字符的位置。

示例 如果单行编辑框 sle_1 的内容为“春风又绿江南岸”,鼠标指针位于“又”字的后面,并且未选中任何文本,则下面的代码返回 7(每个汉字占两个位置):

```
long ll_rc
ll_rc = sle.Position( )
```

7. ReplaceText()

功能

用指定的字符串替换单行编辑框中选定的文本。

语法

```
singlelineeditname.ReplaceText( string )
```

参数

singlelineeditname:单行编辑框名称。

string:string 类型,要取代所选文本的字符串。

返回值

Integer。函数执行成功时返回 string 参数的字符个数,发生错误时返回 -1。如果任何参数的值为 NULL,则 ReplaceText()函数返回 NULL。

用法

如果没有文本被选中,ReplaceText 将 string 参数的内容插入到插入点所在位置。

要插入剪贴板的内容,请使用 Paste()函数。

示例 下面的代码用“北京”取代单行编辑框中所选内容。如果没有所选内容,则在插入点位置插入“北京”:

```
sle_1.ReplaceText("北京")
```

8. SelectedLength()

功能

返回该控件中所选文本的长度。

语法

```
singlelineeditname.SelectedLength( )
```

参数

singlelineeditname:单行编辑框名称。

返回值

Integer。函数执行成功时返回该控件中所选文本的字符个数。如果没有文本被选中,SelectedLength()函数返回 0。如果 singlelineeditname 的值为 NULL,则 SelectedLength()函数返回 NULL。

9. SelectedStart()

功能

返回该控件中所选文本的起始位置。

语法

singlelineeditname.SelectedStart()

参数

singlelineeditname: 单行编辑框名称。

返回值

Integer。函数执行成功时返回该控件中所选文本的起始位置。如果没有文本被选中,SelectedStart()函数返回插入点后第一个字符的位置。如果发生错误,则函数返回-1。如果 singlelineeditname 的值为 NULL,则 SelectedStart()函数返回 NULL。

10. SelectedText()

功能

返回该控件中所选文本。

语法

singlelineeditname.SelectedText()

参数

singlelineeditname: 单行编辑框名称。

返回值

String。函数执行成功时返回该控件中所选文本。如果没有文本被选中或发生错误,则函数返回空字符串("")。如果 singlelineeditname 的值为 NULL,则 SelectedText()函数返回 NULL。

11. SelectText()

功能

在该控件中选中指定的文本。

语法

singlelineeditname.SelectText(start, length)

参数

singlelineeditname: 单行编辑框名称。

start: long 类型,指定要选择文本的开始位置。

length: long 类型,指定要选择文本的长度。如果 length 参数的值为 0,则没有文本被选中,但 PowerBuilder 将把插入点移动到 start 参数指定的位置。

返回值

Integer。函数执行成功时返回所选文本的字符个数,发生错误时返回-1。如果任何参数的值为 NULL,则 SelectText()函数返回 NULL。

用法

如果在调用 SelectText()函数时该控件未得到焦点,那么在该控件得到焦点后所选

文本才会加亮显示。要把焦点设置到某个控件上,请调用该控件的 `SetFocus()` 函数。

如果想选中该控件中的所有内容,那么可以使用 `Len()` 函数得到该控件中文本的长度,然后把以 `start` 参数为 1、`length` 参数为得到的长度来调用 `SelectText()` 函数。

示例 1 下面的代码将插入点设置在单行编辑框的尾部:

```
sle_name.SelectText(Len(sle_name.Text), 0)
```

示例 2 下面的代码选中单行编辑框的所有内容:

```
sle_name.SelectText(1, Len(sle_name.Text))
```

12. `Undo()`

功能

取消在该控件中所做的最后一次编辑。

语法

```
singlelineeditname.Undo()
```

参数

`singlelineeditname`: 单行编辑框名称。

返回值

Integer。函数执行成功时返回 1, 发生错误时函数返回 -1。如果 `singlelineeditname` 的值为 NULL, 则 `Undo()` 函数返回 NULL。

用法

如果想知道上次修改能否取消, 请使用 `CanUndo()` 函数。

第二章 多行编辑框

多行编辑框(MultiLineEdit)可以输入、编辑、显示多行文本,通常用于用户输入。

2.1 多行编辑框的属性

多行编辑框的属性如表 2.1 所列。

表 2.1 多行编辑框的属性

属性	类型	意 义
Accelerator	Integer	指定该控件的快捷键,使用相应键的 ASCII 码表示
Alignment	Alignment (枚举)	指定该控件中文本对齐方式。有效取值为:Center! ——居中对齐、Justify! ——均匀排列;Left! ——左对齐;Right! ——右对齐
AutoHScroll	Boolean	指定随着数据的输入或删除该控件是否自动水平滚动数据。有效取值为:True——自动水平滚动;False——不自动水平滚动
AutoVScroll	Boolean	指定随着数据的输入或删除该控件是否自动垂直滚动数据。有效取值为:True——自动垂直滚动;False——不自动垂直滚动
BackColor	Long	指定该控件的背景颜色,可能取值在-2 到 16 777 215 之间
Border	Boolean	指定该控件是否有边框。有效取值为:True——有边框;False——无边框
BorderStyle	BorderStyle (枚举)	指定该控件的边框风格,有效取值为:StyleBox!、StyleLowered!、StyleRaised!、StyleShadowBox!
BringToTop	Boolean	指定 PowerBuilder 是否把该控件移动到其他控件的前面。有效取值为:True——移动;False——不移动
DisplayOnly	Boolean	指定该控件是否处于只读方式,处于只读方式时,用户不能修改该控件中的文本。有效取值为:True——只读方式;False——用户可以修改数据
DragAuto	Boolean	指定 PowerBuilder 是否自动把该控件置入拖曳模式(Drag Mode)。有效取值为:True——当该控件被单击时,该控件自动进入拖曳模式;False——当该控件被单击时,该控件不自动进入拖曳模式。此时,如果需要让该控件进入拖曳模式,需要在代码中使用 Drag() 函数手工设置
DragIcon	String	指定拖曳图标的名称,该名称可以是系统预定义图标名(比如 exclamation!、information! 等),也可以是图标文件名(.ICO)。缺省时系统使用控件大小的方框作为拖曳图标。在用户拖动控件过程中,当被拖动控件进入该控件可以被放下的区域时,系统显示拖动图标
Enabled	Boolean	指定该控件是否能用(即用户是否能操作),有效取值为:True——用户能够操作该控件;False——用户不能够操作该控件