Raize Components for Delphi

TRzToolbar Component

Unit RzPanel **Inherits from** TRzCustomPanel

Description
TRzToolbar is a specialized panel component for creating toolbars which automatically adjusts its appearance according to the operating system. Of course, the default appearance can be overridden using the BorderInner, BorderOuter, BorderSides, and FrameSides properties. Under Windows 95 or Windows NT 4.0, the divider line is on by default. The divider line is a groove that separates the toolbar from the main menu or from other toolbars. Turn off the divider using the ShowDivider property.

Populating a toolbar with speed buttons is a snap using the custom component editor. Simply invoke the editor and select the buttons to be created from a palette of 45 standard buttons. The selected button is created on the toolbar and is automatically aligned and positioned. In addition, the same bitmap from the palette is used in the new button.

Properties

About
Align
BevelWidth
BorderColor
BorderInner
BorderOuter
BorderSides
BorderWidth
Color
Ctl3D
Cursor
DragCursor
DragMode
Enabled
Font
FrameSides
Height
HelpContext
Hint
Left
Locked
Name
ParentColor
ParentCtl3D
ParentFont
ParentShowHint
PopupMenu
ShowHint
ShowDivider
TabOrder
TabStop
Tag
Top
Visible
Width

Methods

Create
Destroy

Events

OnClick
OnDblClick
OnDragDrop
OnDragOver
OnEndDrag
OnEnter
OnExit
OnMouseDown
OnMouseEnter
OnMouseLeave
OnMouseMove
OnMouseUp
OnResize
OnStartDrag

Raize Components for Delphi Contents

Class Hierarchy

The following is the class hierarchy for Raize Components. Notice that standard Delphi classes have also been included for easy reference.

```
TObject
|__ TPersistent
|   |__ TComponent
|       |__ TControl
|       |   |__ TGraphicControl
|       |   |   |__ TCustomLabel
|       |   |   |   |__ TRzLabel
|       |   |   |   |       |__ TRzDBLabel
|       |   |   |__ TRzBorder
|       |   |   |__ TRzCustomStatusPane
|       |   |   |   |__ TRzStatusPane
|       |   |   |   |__ TRzDBStatusPane
|       |   |   |   |__ TRzCustomGlyphStatus
|       |   |   |   |   |__ TRzGlyphStatus
|       |   |   |   |   |__ TRzDBStateStatus
|       |   |   |   |__ TRzPollingStatus
|       |   |   |   |       |__ TRzClockStatus
|       |   |   |   |       |__ TRzKeyStatus
|       |   |   |   |       |__ TRzResourceStatus
|       |   |   |__ TRzCustomProgressBar
|       |   |   |   |__ TRzProgressBar
|       |   |   |   |__ TRzDBProgressBar
|       |   |   |__ TSpeedButton
|       |   |       |__ TRzRapidFireButton
|       |   |__ TWinControl
|       |       |__ TCustomComboBox
|       |       |   |__ TRzCustomComboBox
|       |       |   |   |__ TRzComboBox
|       |       |   |   |__ TRzColorComboBox
|       |       |   |   |__ TRzFontComboBox
|       |       |   |__ TDBComboBox
|       |       |   |   |__ TRzDBComboBox
|       |       |   |__ TDriveComboBox
|       |       |       |__ TRzDriveComboBox
|       |       |__ TCustomControl
|       |       |   |__ TCustomPanel
|       |       |   |   |__ TRzCustomPanel
|       |       |   |       |__ TRzCustomRadioGroup
|       |       |   |       |   |__ TRzRadioGroup
|       |       |   |       |   |__ TRzDBRadioGroup
|       |       |   |       |__ TRzPanel
|       |       |   |       |__ TRzSplitter
|       |       |   |       |__ TRzStatusBar
|       |       |   |       |__ TRzToolbar
|       |       |   |__ TRzTrackBar
|       |       |       |__ TRzDBTrackBar
|       |       |__ TCustomEdit
|       |       |   |__ TRzLineEdit
|       |       |   |__ TCustomMaskEdit
|       |       |       |__ TDBEdit
|       |       |           |__ TRzDBLineEdit
|       |       |__ TCustomListBox
|       |       |   |__ TRzCustomListBox
|       |       |   |   |__ TRzListBox
|       |       |   |       |__ TRzTabbedListBox
|       |       |   |__ TDBListBox
|       |       |   |   |__ TRzDBListBox
|       |       |   |__ TRzCheckList
|       |       |   |__ TDirectoryListBox
|       |       |   |   |__ TRzDirectoryListBox
|       |       |   |__ TFileListBox
|       |       |       |__ TRzFileListBox
|       |       |__ TRzButtonEdit
|       |       |   |__ TRzDBButtonEdit
|       |       |__ TRzSpinEdit
|       |           |__ TRzDBSpinEdit
|       |__ TRzDBLookupDialog
|       |__ TRzLauncher
|       |__ TRzLookupDialog
|       |__ TRzSendMessage
|       |__ TRzSelDirDialog
|
|__ Exception
    |__ EAbort
    |   |__ EMapiUserAbort
    |__ EMapiError
    |__ ELaunchError
    |__ EInvalidSearchError
    |__ EInvalidDataset
```

TRzBorder Component

Unit RzBorder **Inherits from** TGraphicControl

Description
This component is very similar in appearance to a TRzPanel component. The difference is that the TRzBorder component does not use a window handle and therefore cannot be used as a container for other components. There are many ways to customize how the border is displayed. The BorderInner and BorderOuter properties support many frame styles, including Windows 95 styles. The area between the inner and outer borders can be filled by specifying a BorderColor. The BorderSides property can be used to selectively display only some of the sides of the border making it easy to create separator lines (e.g. grooves and bumps).

Properties

About
Align
BorderColor
BorderInner
BorderOuter
BorderSides
BorderWidth
Cursor
Height
Hint
Left
Name
ParentShowHint
ShowHint
Tag
Top
Width

Methods

Create
Destroy

Events

OnMouseEnter
OnMouseLeave

TRzSplitter Component

Unit RzSplit **Inherits from** TRzCustomPanel

Description

TRzSplitter is a specialized panel component designed for splitting two regions of a window. The TRzSplitter component consists of three areas: the upper-left pane, the lower-right pane, and the splitter bar. The two panes serve as place holders for other components such as list boxes and tree views, and are automatically resized when the user moves the splitter bar. By default, when the user moves the splitter bar, a mask is displayed showing the new position of the bar, and when the user releases the mouse, the bar is repositioned. When the RealTimeDrag property is set to True, the bar is repositioned as the user moves the mouse.

The splitter bar can be oriented either horizontally or vertically via the Orientation property. The appearance of the splitter can be altered using the BorderInner, BorderOuter, SplitterStyle, and SplitterWidth properties. The UpperLeft and LowerRight properties provide design-time access to the underlying splitter panes. Both of these properties are class properties encapsulating several properties that define the appear of the splitter panes. For example, if the UpperLeft.Visible property is set to False, the upper left pane is removed from view. To access the actual pane object, which is a descendant of TCustomPanel, use the Pane subproperty (e.g. LowerRight.Pane).

Since there are two panes that can serve as containers for other components, pasting controls previously copied to the clipboard requires selecting the destination pane before pasting the controls to the splitter component. The currently selected pane has a dashed line around its perimeter. To change the selected pane, use the SelectedPane property or select the desired pane from the component's popup menu

The custom component editor provides quick access to many of the splitter's properties, including the the splitter panes, and provides instant visual feedback.

Properties

About
Align
BorderInner
BorderOuter
BorderWidth
Color
Cursor
Enabled
Height
HelpContext
Hint
Left
Locked
LowerRight
Name
Orientation
ParentColor
ParentShowHint
PopupMenu
Position
RealTimeDrag
SelectedPane
ShowHint
SplitterStyle
SplitterWidth
TabOrder
TabStop
Tag
Top
UpperLeft
Visible
Width

Methods

Create
Destroy

Events

OnChange
OnChanging
OnEnter
OnExit
OnResize

TRzStatusBar Component

Unit RzPanel **Inherits from** TRzCustomPanel

Description

TRzStatusBar is a specialized panel component for displaying status information which automatically adjusts its appearance according to the operating system. Of course, the default appearance can be overridden using the BorderInner, BorderOuter, BorderSides, and FrameSides properties. By default, the size grip is visible in the lower right corner. To turn off the size grip, use the ShowSizeGrip property.

The TRzStatusBar component makes a great holder for the Raize status components (i.e. TRzStatusPane, TRzGlyphStatus, TRzClockStatus, TRzDBStatusPane, TRzDBStateStatus, TRzResourceStatus, TRzKeyStatus, and TRzProgressBar). Plus, the custom component editor for TRzStatusBar makes it a snap to create status components automatically positioned and aligned within the status bar. Simply right-click on the status bar to display a list of status components.

Properties

- About
- Align
- BevelWidth
- BorderColor
- BorderInner
- BorderOuter
- BorderSides
- BorderWidth
- Color
- Ctl3D
- Cursor
- DragCursor
- DragMode
- Enabled
- Font
- FrameSides
- Height
- HelpContext
- Hint
- Left
- Locked
- Name
- ParentColor
- ParentCtl3D
- ParentFont
- ParentShowHint
- PopupMenu
- ShowHint
- ShowSizeGrip
- TabOrder
- TabStop
- Tag
- Top
- Visible
- Width

Methods

- Create
- Destroy

Events

- OnClick
- OnDblClick
- OnDragDrop
- OnDragOver
- OnEndDrag
- OnEnter
- OnExit
- OnMouseDown
- OnMouseEnter
- OnMouseLeave
- OnMouseMove
- OnMouseUp
- OnResize
- 32 OnStartDrag

TRzPanel Component

Unit RzPanel **Inherits from** TRzCustomPanel

Description
This component is an enhanced panel component that provides many ways to customize how the panel is displayed. The BorderInner and BorderOuter properties support many frame styles, including Windows 95 styles. The area between the inner and outer borders can be filled by specifying a BorderColor. The BorderSides and FrameSides properties can be used to selectively display only some of the sides of the panel (well suited for creating toolbars, status bars, and splitter bars). The AlignmentVertical property along with the standard Alignment property allows the caption to be placed in one of nine different locations on the panel.

The custom component editor provides quick access to many of the panel's properties and provides instant visual feedback.

Properties

About
Align
Alignment
AlignmentVertical
BevelWidth
BorderColor
BorderInner
BorderOuter
BorderSides
BorderWidth
Caption
Color
Ctl3D
Cursor
DragCursor
DragMode
Enabled
Font
FrameSides
Height
HelpContext
Hint
Left
Locked
Name
ParentColor
ParentCtl3D
ParentFont
ParentShowHint
PopupMenu
ShowHint
TabOrder
TabStop
Tag
Top
Visible
Width

Methods

Create
Destroy

Events

OnClick
OnDblClick
OnDragDrop
OnDragOver
OnEndDrag
OnEnter
OnExit
OnMouseDown
OnMouseEnter
OnMouseLeave
OnMouseMove
OnMouseUp
OnResize
32 OnStartDrag

ShowSizeGrip Property

Applies to
TRzStatusBar

Declaration

```
property ShowSizeGrip : Boolean;
```

Description
This property determines whether the size grip of the status bar is visible. The size grip is located in the lower right corner of the status bar and functions like a sizing border. Click on the sizing grip and drag it to resize the parent window.

ShadowColor Property

Applies to
TRzLabel, TRzDBLabel

Declaration
```
property ShadowColor : TColor;
```

Description
Use this property to specify the color of the shadow.   This property is only applicable when TextStyle is set to tsShadow.

TRzLabel Component

Unit RzLabel **Inherits from** TCustomLabel

Description
Descendant of the TLabel. This component provides additional properties to enable this component to display its caption in several different three-dimensional styles using the TextStyle property. In addition, the label can be rotated to any angle using the Angle property. The custom component editor provides quick access to many of the label's properties and provides instant visual feedback.

Raised
Recessed
Shadow
On Angle

Properties

- About
- Align
- Alignment
- Angle
- AutoSize
- Caption
- Color
- Cursor
- DragCursor
- DragMode
- Enabled
- FocusControl
- Font
- Height
- Hint
- Left
- Name
- ParentColor
- ParentFont
- ParentShowHint
- PopupMenu
- ShadowColor
- ShadowDepth
- ShowAccelChar
- ShowHint
- Tag
- TextStyle
- Top
- Transparent
- Visible
- Width
- WordWrap

Methods

- Create
- Destroy

Events

- OnClick
- OnDblClick
- OnDragDrop
- OnDragOver
- OnEndDrag
- OnMouseDown
- OnMouseEnter
- OnMouseLeave
- OnMouseMove
- OnMouseUp
- OnStartDrag

TRzDBLabel Component

Unit RzDBLbl **Inherits from** TRzLabel

Description
Data-aware version of TRzLabel. Provides standard DataField and DataSource properties for connecting to a column in a database table.

Properties

- About
- Align
- Alignment
- Angle
- AutoSize
- Caption
- Color
- Cursor
- DataField
- DataSource
- DragCursor
- DragMode
- Enabled
- FocusControl
- Font
- Height
- Hint
- Left
- Name
- ParentColor
- ParentFont
- ParentShowHint
- PopupMenu
- ShadowColor
- ShadowDepth
- ShowAccelChar
- ShowHint
- Tag
- TextStyle
- Top
- Transparent
- Visible
- Width
- WordWrap

Methods

- Create
- Destroy

Events

- OnClick
- OnDblClick
- OnDragDrop
- OnDragOver
- OnEndDrag
- OnMouseDown
- OnMouseEnter
- OnMouseLeave
- OnMouseMove
- OnMouseUp
- 32 OnStartDrag

TRzStatusPane Component

Unit RzStatus **Inherits from** TRzCustomStatusPane

Description

This direct descendant of TRzCustomStatusControl. simply publishes the Alignment, Caption, and CaptionOffset properties. The parent class provides the ability to display a status border, which by default is fsStatus, and the TRzStatusPane class provides the ability to display a caption within the border. The style of the border is controlled by the FrameStyle property. By default the frame is drawn two pixels away from the edge of the control. This distance can be changed by setting the BorderWidth property to a new value.

Properties

About
Align
Alignment
BorderWidth
Caption
CaptionOffset
Cursor
DragCursor
DragMode
Enabled
FillColor
Font
FrameStyle
Height
Hint
Left
Name
ParentColor
ParentFont
ParentShowHint
PopupMenu
ShowHint
Tag
Top
Visible
Width

Methods

Create
Destroy

Events

OnClick
OnDblClick
OnDragDrop
OnDragOver
OnEndDrag
OnMouseDown
OnMouseEnter
OnMouseLeave
OnMouseMove
OnMouseUp
OnStartDrag

TRzDBStatusPane Component

Unit RzDBStat **Inherits from** TRzCustomStatusPane

Description

This status control descendant can be connected to a field in a dataset by setting the DataSource and DataField properties. Whenever the data in the corresponding dataset changes, the contents of the status pane are updated. The FieldLabel property can be used to specify a static text string to prefix the data stored in the dataset. In addition, the Hint property is automatically populated with the name of the data field connected to the component.

Properties

- About
- Align
- Alignment
- BorderWidth
- Caption
- CaptionOffset
- Cursor
- DataField
- DataSource
- DragCursor
- DragMode
- Enabled
- FieldLabel
- FillColor
- Font
- FrameStyle
- Height
- Hint
- Left
- Name
- ParentColor
- ParentFont
- ParentShowHint
- PopupMenu
- ShowHint
- Tag
- Top
- Visible
- Width

Methods

- Create
- Destroy

Events

- OnClick
- OnDblClick
- OnDragDrop
- OnDragOver
- OnEndDrag
- OnMouseDown
- OnMouseEnter
- OnMouseLeave
- OnMouseMove
- OnMouseUp
- OnStartDrag

TRzGlyphStatus Component

Unit RzStatus

Inherits from TRzCustomGlyphStatus

Description

This status component is basically a traditional status control with the additional feature of being able to display a glyph within the status frame. The bitmap displayed is controlled by the Glyph property. The glyph can be position either to the right or left of the status control by using the GlyphAlignment property. Its distance from the edge is controlled by the GlyphOffset property.

Properties

About
Align
Alignment
BorderWidth
Caption
CaptionOffset
Cursor
DragCursor
DragMode
Enabled
FillColor
Font
FrameStyle
Glyph
GlyphAlignment
GlyphOffset
Height
Hint
Left
Name
NumGlyphs
ParentColor
ParentFont
ParentShowHint
PopupMenu
ShowHint
Tag
Top
Visible
Width

Methods

Create
Destroy

Events

OnClick
OnDblClick
OnDragDrop
OnDragOver
OnEndDrag
OnMouseDown
OnMouseEnter
OnMouseLeave
OnMouseMove
OnMouseUp
32 OnStartDrag

TRzDBStateStatus Component

Unit RzDBStat **Inherits from** TRzCustomGlyphStatus

Description

Use this component to display the current state of a dataset. Once the DataSource property is set to a data source, the component displays the state of the dataset connected to the data source. By default, both a glyph and caption are used to indicate the state. Like the TRzGlyphStatus, the position of the glyph can be modified using the GlyphAlignment and GlyphOffset properties. The caption can be repositioned using the CaptionOffset property or turned off using the ShowCaption property.

Properties

- About
- Align
- Alignment
- BorderWidth
- CaptionOffset
- Cursor
- DatasetState
- DataSource
- DragCursor
- DragMode
- Enabled
- FillColor
- Font
- FrameStyle
- GlyphAlignment
- GlyphOffset
- Height
- Hint
- Left
- Name
- ParentColor
- ParentFont
- ParentShowHint
- PopupMenu
- ShowCaption
- ShowHint
- Tag
- Top
- Visible
- Width

Methods

- Create
- Destroy

Events

- OnClick
- OnDblClick
- OnDragDrop
- OnDragOver
- OnEndDrag
- OnMouseDown
- OnMouseEnter
- OnMouseLeave
- OnMouseMove
- OnMouseUp
- OnStartDrag

OnTimerExpired Event

Applies to
TRzPollingStatus, TRzClockStatus, TRzKeyStatus, TRzResourceStatus

Declaration
```
property OnTimerExpired : TNotifyEvent;
```

Description
This event is generated whenever the timer used by polling status controls expires.   The frequency of the timer is controlled by the Interval and Active properties.

TRzClockStatus Component

Unit RzStatus **Inherits from** TRzPollingStatus

Description

This status component displays the current date and time in a status pane. Use the Format property to control how the date and time is displayed.

Properties

- About
- Active
- Align
- Alignment
- BorderWidth
- CaptionOffset
- Cursor
- DragCursor
- DragMode
- Enabled
- FillColor
- Font
- Format
- FrameStyle
- Height
- Hint
- Interval
- Left
- Name
- ParentColor
- ParentFont
- ParentShowHint
- PopupMenu
- ShowHint
- Tag
- Top
- Visible
- Width

Methods

- Create
- Destroy

Events

- OnClick
- OnDblClick
- OnDragDrop
- OnDragOver
- OnEndDrag
- OnMouseDown
- OnMouseEnter
- OnMouseLeave
- OnMouseMove
- OnMouseUp
- OnStartDrag
- OnTimerExpired

TRzKeyStatus Component

Unit RzStatus **Inherits from** TRzPollingStatus

Description
This status component displays the current toggle state of either the CapsLock, NumLock, or ScrollLock keyboard keys. The Key property is used to specify which key to monitor, and the State property is a read only property which is used to query the current state of the selected key.

Properties

- About
- Active
- Align
- BorderWidth
- Cursor
- DragCursor
- DragMode
- Enabled
- FillColor
- Font
- FrameStyle
- Height
- Hint
- Interval
- Key
- Left
- Name
- ParentColor
- ParentFont
- ParentShowHint
- PopupMenu
- ShowHint
- State
- Tag
- Top
- Visible
- Width

Methods

- Create
- Destroy

Events

- OnClick
- OnDblClick
- OnDragDrop
- OnDragOver
- OnEndDrag
- OnMouseDown
- OnMouseEnter
- OnMouseLeave
- OnMouseMove
- OnMouseUp
- OnStartDrag
- OnTimerExpired

TRzResourceStatus Component

Unit RzStatus **Inherits from** TRzPollingStatus

Description
This status component is responsible for displaying information pertaining the current state of resource usage. The ResourceType property determines which resource to monitor: System, GDI, User, or Memory. The DisplayStyle property determines whether a progress bar is used to display the status or just simple text.

Properties

About
Active
Align
Alignment
BackColor
BarColor
BarStyle
BorderWidth
CaptionOffset
Cursor
DisplayStyle
DragCursor
DragMode
Enabled
FillColor
Font
FrameStyle
Height
Hint
Interval
Left
Name
NumSegments
Orientation
ParentColor
ParentFont
ParentShowHint
PopupMenu
ResourceType
ShowHint
ShowPercent
Tag
Top
Visible
Width

Methods

Create
Destroy

Events

OnClick
OnDblClick
OnDragDrop
OnDragOver
OnEndDrag
OnMouseDown
OnMouseEnter
OnMouseLeave
OnMouseMove
OnMouseUp
32 OnStartDrag
OnTimerExpired

TRzTrackBar Component

Unit RzTrkBar **Inherits from** TCustomControl

Description

The TRzTrackBar component is a slider control that mimics the behavior of the Windows 95 TrackBar control. The track bar consists of a thumb that is dragged along a track. The thumb rests at periodic positions (tick marks) along the track. The thumb cannot be placed in between these tick marks.

Many of the properties control how the track and thumb are drawn. For example, the ThumbStyle property can be used to change the style of the thumb. Standard choices include pointer, box, and mixer. TRzTrackBar even supports custom thumbs through the CustomThumb bitmap property.

The track width and color are controlled by the TrackWidth and TrackColor properties, respectively. Along the edge of the track tick marks can be displayed for each position by setting the ShowTicks to True. Custom tick marks can even be drawn by setting the TickStyle property to tkOwnerDraw and writing an event handler for the OnDrawTick event.

Properties

About
Color
Cursor
CustomThumb
DragCursor
DragMode
Enabled
Height
HelpContext
Hint
Left
Max
Min
Name
Orientation
PageSize
ParentShowHint
PointerType
PopupMenu
Position
ShowHint
ShowTicks
TabOrder
TabStop
Tag
ThumbStyle
TickStyle
Top
TrackColor
TrackOffset
TrackWidth
Visible
Width

Methods

Create
Destroy

Events

OnChange
OnChanging
OnClick
OnDragDrop
OnDragOver
OnDrawTick
OnEndDrag
OnEnter
OnExit
OnKeyDown
OnKeyPress
OnKeyUp
OnMouseDown
OnMouseMove
OnMouseUp
32 OnStartDrag

IncPartsByOne Method

Applies to
TRzProgressBar, TRzDBProgressBar

Declaration

```
procedure IncPartsByOne;
```

Description
Increments the PartsComplete property by 1.

TRzDBTrackBar Component

Unit RzDBTrak **Inherits from** TRzTrackBar

Description
Data-aware version of TRzTrackBar. Provides standard DataField and DataSource properties for connecting to a column in a database table. In addition, this component provides the Values string list property which is populated with the values to be written to the database table. After a set of values have been entered, the Min and Max positions of the track bar are updated according to the number of items in the list box. At run-time, as the user moves the thumb of the track bar, the value from the Values list corresponding to the selected position is stored in the database field.

Properties

- About
- Color
- Cursor
- CustomThumb
- DataField
- DataSource
- DragCursor
- DragMode
- Enabled
- Field
- Value
- Height
- HelpContext
- Hint
- Left
- Max
- Min
- Name
- Orientation
- PageSize
- ParentShowHint
- PointerType
- PopupMenu
- Position
- ReadOnly
- ShowHint
- ShowTicks
- TabOrder
- TabStop
- Tag
- ThumbStyle
- TickStyle
- Top
- TrackColor
- TrackOffset
- TrackWidth
- Values
- Visible
- Width

Methods

- Create
- Destroy

Events

- OnChange
- OnChanging
- OnClick
- OnDragDrop
- OnDragOver
- OnDrawTick
- OnEndDrag
- OnEnter
- OnExit
- OnKeyDown
- OnKeyPress
- OnKeyUp
- OnMouseDown
- OnMouseMove
- OnMouseUp
- OnStartDrag

TRzProgressBar Component

Unit RzPrgres **Inherits from** TGraphicControl

Description

This component is used to display a progress indicator which displays a solid bar extending either to the right or to the top of the control depending the Orientation setting. The progress bar is filled using the color specified in the BarColor property. If the ShowPercent property is True, the Percent property value is displayed in the center of the progress bar. However, ShowPercent has no effect if the BarStyle property is set to bsLED. In this case, the bar is filled with segments, up to NumSegments, instead of a solid bar.

The current progress level can be controlled by setting the Percent property directly or by using the TotalParts and PartsComplete properties. If the latter is used, the IncParts and IncPartsByOne methods can be used to easily increment the PartsComplete property.

Properties

- About
- Align
- BackColor
- BarColor
- BarStyle
- BevelWidth
- BorderColor
- BorderInner
- BorderOuter
- BorderWidth
- Cursor
- Font
- Height
- Hint
- InteriorOffset
- Left
- Name
- NumSegments
- Orientation
- ParentFont
- ParentShowHint
- PartsComplete
- Percent
- PopupMenu
- ShowHint
- ShowPercent
- Tag
- Top
- TotalParts
- Visible
- Width

Methods

- Create
- Destroy
- IncParts
- IncPartsByOne

Events

- OnChange
- OnClick
- OnDblClick
- OnDblClick
- OnMouseDown
- OnMouseMove
- OnMouseUp

TRzDBProgressBar Component

Unit RzDBProg **Inherits from** TRzProgressBar

Description
Data-aware version of TRzProgressBar. Provides standard DataField and DataSource properties for connecting to a column in a database table. However, the TRzDBProgressBar component is slightly different than its non-data-aware counterpart in that the percentage value displayed can be automatically calculated from the data in the table.

There are three ways to display a percentage value.

1. Set the DataField to the desired table column containing a percentage value
2. Set the DataField to one column and the BaseField to another column. The percentage is calculated between the two columns.
   Percent := DataField / BaseField * 100
3. Set the DataField to a column in the table and set the BaseValue property to a static value. The percentage is calculated using the BaseValue.

Properties

About
Align
BackColor
BarColor
BarStyle
BaseField
BaseValue
BevelWidth
BorderColor
BorderInner
BorderOuter
BorderWidth
Cursor
DataField
DataSource
Font
Height
Hint
InteriorOffset
Left
Name
NumSegments
Orientation
ParentFont
ParentShowHint
Percent
PopupMenu
ShowHint
ShowPercent
Tag
Top
Visible
Width

Methods

Create
Destroy

Events

OnChange
OnClick
OnDblClick
OnDblClick
OnMouseDown
OnMouseMove
OnMouseUp

TRzListBox Component

Unit RzLstBox **Inherits from** TRzCustomListBox

Description

TRzListBox provides all of the features of a regular TListBox plus adds intuitive keyboard searching. That is, as the user types a series of characters the selection bar is moved to the item in the list that most closely matches the characters entered. The read-only SearchString property records the current search string. If the BeepOnInvalidKey property is True, an audible beep is played when the last key pressed creates a non-matching search string.

Properties

- About
- Align
- BeepOnInvalidKey
- BorderStyle
- Color
- Columns
- Ctl3D
- Cursor
- DragCursor
- DragMode
- Enabled
- ExtendedSelect
- Font
- Height
- HelpContext
- Hint
- IntegralHeight
- ItemHeight
- ItemIndex
- Items
- Left
- MultiSelect
- Name
- ParentColor
- ParentCtl3D
- ParentFont
- ParentShowHint
- PopupMenu
- SearchString
- ShowHint
- Sorted
- Style
- TabOrder
- TabStop
- Tag
- Top
- Visible
- Width

Methods

- Clear
- Create
- Destroy

Events

- OnClick
- OnDblClick
- OnDragDrop
- OnDragOver
- OnDrawItem
- OnEndDrag
- OnEnter
- OnExit
- OnKeyDown
- OnKeyPress
- OnKeyUp
- OnMeasureItem
- OnMouseDown
- OnMouseMove
- OnMouseUp
- OnStartDrag

TRzDBListBox Component

Unit RzDBList **Inherits from** TDBListBox

Description
Data-aware version of TRzListBox. Provides standard DataField and DataSource properties for connecting to a column in a database table.

Properties

- About
- Align
- BeepOnInvalidKey
- BorderStyle
- Color
- Ctl3D
- Cursor
- DataField
- DataSource
- DragCursor
- DragMode
- Enabled
- Field
- Font
- Height
- HelpContext
- Hint
- IntegralHeight
- ItemHeight
- ItemIndex
- Items
- Left
- Name
- ParentColor
- ParentCtl3D
- ParentFont
- ParentShowHint
- PopupMenu
- ReadOnly
- SearchString
- ShowHint
- TabOrder
- TabStop
- Tag
- Top
- Visible
- Width

Methods

- Clear
- Create
- Destroy

Events

- OnClick
- OnDblClick
- OnDragDrop
- OnDragOver
- OnDrawItem
- OnEndDrag
- OnEnter
- OnExit
- OnKeyDown
- OnKeyPress
- OnKeyUp
- OnMeasureItem
- OnMouseDown
- OnMouseMove
- OnMouseUp
- OnStartDrag

TRzTabbedListBox Component

Unit RzLstBox **Inherits from** TRzListBox

Description

This component is an enhanced list box component which understands how to process embedded tab characters to display lines in columnar format. In addition, a horizontal scroll bar can be added to the list box. Tab stop values are specified using the TabStops integer list property. In addition, it inherits the keyboard searching features of TRzListBox.

Properties

- About
- Align
- BeepOnInvalidKey
- BorderStyle
- Color
- Columns
- Ctl3D
- Cursor
- DragCursor
- DragMode
- Enabled
- ExtendedSelect
- Font
- Height
- HelpContext
- Hint
- HorzExtent
- HorzScrollBar
- IntegralHeight
- ItemIndex
- Items
- Left
- MultiSelect
- Name
- ParentColor
- ParentCtl3D
- ParentFont
- ParentShowHint
- PopupMenu
- ShowHint
- SearchString
- Sorted
- Style
- TabOrder
- TabStop
- TabStops
- Tag
- Top
- Visible
- Width

Methods

- Clear
- Create
- Destroy
- UpdateTabStops

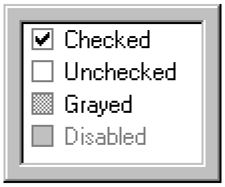
Events

- OnClick
- OnDblClick
- OnDragDrop
- OnDragOver
- OnDrawItem
- OnEndDrag
- OnEnter
- OnExit
- OnKeyDown
- OnKeyPress
- OnKeyUp
- OnMeasureItem
- OnMouseDown
- OnMouseMove
- OnMouseUp
- OnStartDrag

TRzCheckList Component

Unit RzChkLst **Inherits from** TRzCustomListBox

Description
The TRzCheckList component provides all of the features of the standard TListBox, plus it also associates a check box with each item in the list. By default, each check box either has a *checked* or *unchecked* state. To allow the *grayed* state, set the AllowGrayed property to True. To determine the state of a particular item, use the ItemState property, and to enable or disable an individual item, use the ItemEnabled state.



☑ Checked
□ Unchecked
▨ Grayed
□ Disabled

The custom component editor can be used to add, change, and delete items from the list at design-time. In addition, the ItemState and ItemEnabled properties can be modified within the editor.

Properties

About
Align
AllowGrayed
BeepOnInvalidKey
SearchString
BorderStyle
Color
Ctl3D
Cursor
DragCursor
DragMode
Enabled
Font

Height
HelpContext
Hint
ItemEnabled
ItemHeight
ItemIndex
ItemState
Items
Left
Name
ParentColor
ParentCtl3D

ParentFont
ParentShowHint
PopupMenu
ShowHint
Sorted
TabOrder
TabStop
Tag
Top
Visible
Width

Methods

Clear
Create
Destroy

Events

OnChanging
OnClick
OnDblClick
OnDragDrop
OnDragOver

OnEndDrag
OnEnter
OnExit
OnKeyDown
OnKeyPress

OnKeyUp
OnMouseDown
OnMouseMove
OnMouseUp
OnStartDrag

TRzComboBox Component

Unit RzCmboBx **Inherits from** TRzCustomComboBox

Description

TRzComboBox provides all of the features of a regular TComboBox plus adds intuitive keyboard searching. That is, as the user types a series of characters the selection bar is moved to the item in the list that most closely matches the characters entered. The read-only SearchString property records the current search string. Keyboard searching is slightly different depending on the Style of combo box. That is, when the Style property is set to csDropDown, as the user types, the closest matching item is displayed in the edit portion with the non-matching characters highlighted. If the Style property is set to csDropDownList and the BeepOnInvalidKey property is True, an audible beep is played when the last key pressed creates a non-matching search string.

Properties

About
BeepOnInvalidKey
Color
Ctl3D
Cursor
DragCursor
DragMode
DropDownCount
DroppedDown
Enabled
Font
Height
HelpContext
Hint
ItemHeight
ItemIndex
Items
Left
MaxLength
Name
ParentColor
ParentCtl3D
ParentFont
ParentShowHint
PopupMenu
SearchString
SelLength
SelStart
ShowHint
Sorted
Style
TabOrder
TabStop
Tag
Text
Top
Visible
Width

Methods

Clear
Create
Destroy

Events

OnChange
OnClick
OnCloseUp
OnDblClick
OnDragDrop
OnDragOver
OnDrawItem
OnDropDown
OnEndDrag
OnEnter
OnExit
OnKeyDown
OnKeyPress
OnKeyUp
OnMeasureItem
OnStartDrag

TRzDBComboBox Component

Unit RzDBCmbo **Inherits from** TDBComboBox

Description
Data-aware version of TRzComboBox. Provides standard DataField and DataSource properties for connecting to a column in a database table.

Properties

About
BeepOnInvalidKey
Color
Ctl3D
Cursor
DataField
DataSource
DragCursor
DragMode
DropDownCount
Enabled
Field
Font
Height
HelpContext
Hint
ItemHeight
ItemIndex
Items
Left
Name
ParentColor
ParentCtl3D
ParentFont
ParentShowHint
PopupMenu
ReadOnly
SearchString
SelLength
SelStart
ShowHint
Sorted
Style
TabOrder
TabStop
Tag
Top
Visible
Width

Methods

Clear
Create
Destroy

Events

OnChange
OnClick
OnDblClick
OnDragDrop
OnDragOver
OnDrawItem
OnDropDown
OnEndDrag
OnEnter
OnExit
OnKeyDown
OnKeyPress
OnKeyUp
OnMeasureItem
OnStartDrag

TRzColorComboBox Component

Unit RzCmboBx **Inherits from** TRzCustomComboBox

Description

This component provides a convient way for selecting a color value. The list portion of the combo box contains all of the standard 16 colors, unless the ShowSysColors property is set to True. In this case, the standard 16 plus all of the Windows System colors (e.g. clWindow) are listed. Next to each name is a sample of the color. The SelectedColor property allows a developer to determine the color selected by the user without having to translate an index value. The SelectedColor property can also be used to set the combo box to a particular color value.

Properties

About
BeepOnInvalidKey
Color
Ctl3D
Cursor
DefaultColor
DragCursor
DragMode
DropDownCount
DroppedDown
Enabled
Font
Height
HelpContext
Hint
ItemIndex
Items
Left
Name
ParentColor
ParentCtl3D
ParentFont
ParentShowHint
PopupMenu
SearchString
SelectedColor
SelLength
SelStart
ShowHint
ShowSysColors
Sorted
TabOrder
TabStop
Tag
Top
Visible
Width

Methods

Create
Destroy

Events

OnChange
OnClick
OnCloseUp
OnDblClick
OnDragDrop
OnDragOver
OnDropDown
OnEndDrag
OnEnter
OnExit
OnKeyDown
OnKeyPress
OnKeyUp
OnStartDrag

TRzFontComboBox Component

Unit RzCmboBx **Inherits from** TRzCustomComboBox

Description

This component provides a convient way for selecting a font name. The combo box lists the names of fonts along with a glyph that indicates the type of font (i.e. True Type, fixed pitched, or printer). The FontType property is used to restrict the list to only those fonts of a certain type, while the FontDevice property is used to select either the screen or the printer as the source of the font list. The FontName, FontSize, and FontStyle properties are used to determine the characteristics of the font selected. And the SelectedFont property is used to obtain a TFont value corresponding to the font name selected in the list.

Properties

About
BeepOnInvalidKey
Color
Ctl3D
Cursor
DragCursor
DragMode
DropDownCount
Enabled
Font
FontDevice
FontName
FontSize
FontStyle
FontType
Height
HelpContext
Hint
ItemIndex
Items
Left
Name
ParentColor
ParentCtl3D
ParentFont
ParentShowHint
PopupMenu
SearchString
SelectedFont
ShowHint
Sorted
TabOrder
TabStop
Tag
Top
Visible
Width

Methods

Create
Destroy

Events

OnChange
OnClick
OnCloseUp
OnDblClick
OnDragDrop
OnDragOver
OnDropDown
OnEndDrag
OnEnter
OnExit
OnKeyDown
OnKeyPress
OnKeyUp
OnStartDrag

TRzDriveComboBox Component

Unit RzFilSys **Inherits from** TDriveComboBox

Description
This is a simple replacement for the TDriveComboBox component. The glyphs used for the drives have been updated to reflect a more three dimensional appearance. This component completes the set of new file oriented components which includes TRzFileListBox, TRzDirectoryListBox, and TRzSelDirDialog.

Properties

- About
- Color
- Ctl3D
- Cursor
- DirList
- DragCursor
- DragMode
- Drive
- Enabled
- Font
- Height
- HelpContext
- Hint
- Left
- Name
- ParentColor
- ParentCtl3D
- ParentFont
- ParentShowHint
- PopupMenu
- ShowHint
- TabOrder
- TabStop
- Tag
- TextCase
- Top
- Visible
- Width

Methods

- Create
- Destroy

Events

- OnChange
- OnClick
- OnDblClick
- OnDragDrop
- OnDragOver
- OnDropDown
- OnEndDrag
- OnEnter
- OnExit
- OnKeyDown
- OnKeyPress
- OnKeyUp
- OnStartDrag

TRzDirectoryListBox Component

Unit RzFilSys **Inherits from** TDirectoryListBox

Description
This component provides all of the functionality of the TDirectoryListBox component, plus it uses updated three dimensional glyphs that follow the style used in Windows 95 and it supports long directory names even in 16-bit applications. The ShowLongNames property can be used to turn off this feature, and the LongDirName property returns the long version of the currently selected directory name.

Properties

About
Align
Color
Columns
Ctl3D
Cursor
Directory
DirLabel
DragCursor
DragMode
Drive
Enabled
FileList
Font
Height
HelpContext
Hint
IntegralHeight
ItemHeight
Left
LongDirName
Name
ParentColor
ParentCtl3D
ParentFont
ParentShowHint
PopupMenu
ShowHint
ShowLongNames
TabOrder
TabStop
Tag
Top
Visible
Width

Methods

Create
Destroy
GetItemPath
OpenCurrent
Update

Events

OnChange
OnClick
OnDblClick
OnDragDrop
OnDragOver
OnEndDrag
OnEnter
OnExit
OnKeyDown
OnKeyPress
OnKeyUp
OnMouseDown
OnMouseMove
OnMouseUp
OnStartDrag

TRzFileListBox Component

Unit RzFilSys **Inherits from** TFileListBox

Description
This component provides all of the functionality of the TFileListBox component, but it also uses updated three dimensional glyphs that follow the style used in Windows 95 and it supports long file names even in 16-bit applications. The ShowLongNames property can be used to turn off this feature, and the LongFileName property returns the long version of the currently selected file. Plus, this file list box supports multiple columns through the Columns property.

Under Windows 95 or Windows NT, the ShortFileName property can be used to obtain the short version of the selected file name. In addition, the glyphs displayed next to each file name are obtained from the shell and correspond to the icon associated with the file type.

Properties

About
Align
Color
Columns
Ctl3D
Cursor
Directory
DragCursor
DragMode
Drive
Enabled
FileEdit
FileName
FileType
Font
Height
HelpContext
Hint
IntegralHeight
ItemHeight
Left
LongFileName
Mask
MultiSelect
Name
ParentColor
ParentCtl3D
ParentFont
ParentShowHint
PopupMenu
ShortFileName
ShowGlyphs
ShowHint
ShowLongNames
TabOrder
TabStop
Tag
Top
Visible
Width

Methods

ApplyFilePath
Create
Destroy
Update

Events

OnChange
OnClick
OnDblClick
OnDragDrop
OnDragOver
OnEndDrag
OnEnter
OnExit
OnKeyDown
OnKeyPress
OnKeyUp
OnMouseDown
OnMouseMove
OnMouseUp
32 OnStartDrag

TRzSelDirDialog Component

Unit RzSelDir **Inherits from** TComponent

Description

This dialog component provides a clean and efficient way to prompt a user to select a directory. The control uses the TRzDirectoryListBox and TRzDriveComboBox components to utilize the new glyphs and provide support for long file names. Setting the AllowCreate property to True instructs the dialog component to provide an edit field in which the user may enter a new directory name.

When the user presses the OK button, the selected directory is checked for existence. If it exists, the dialog is closed. If not, the user is prompted to verify the creation of the new directory. Once accepted, the dialog component creates the directory and closes the dialog box. The selected directory is available in the Directory property.

Properties

About
AllowCreate
ButtonGlyphs
Directory
Font
HelpContext
Name
Tag

Methods

Create
Destroy
Execute

TRzLookupDialog Component

Unit RzLookup **Inherits from** TComponent

Description

This dialog component provides an alternative to using a combo box to perform a lookup. The dialog box that is displayed consists of an edit field for keyboard searching and a list box which holds the items being searched. The edit field can be initialized with the contents of a TEdit or TRzButtonEdit component by setting the SearchEdit and SearchBtnEdit properties. To initialize the edit field with a string value, use the SearchString property. Use the List string list property to populate the list box. This can be done at design-time using the string list editor, or at run-time by assigning another string list to the List property. Customize the appearance of the dialog box by using the Caption, Prompt, and ButtonGlyphs properties. The SelectedIndex property can be used to preselect an item in the list as well as indicating the item in the list selected by the user when the dialog is closed.

Properties

About
BorderStyle
ButtonGlyphs
Caption
Font
Height
HelpContext
List
Name
Prompt
SearchBtnEdit
SearchEdit
SearchString
SelectedIndex
Tag
Width

Methods

Create
Destroy
Execute

TRzDBLookupDialog Component

Unit RzDBLook **Inherits from** TComponent

Description
This component is used to display the contents of a dataset (i.e. the Dataset property) from which the user can select a record. This component is more flexible than a lookup combo because a DBGrid is used to display the data within the dialog box. In addition, the data displayed is controlled by the dataset (i.e. a TTable or TQuery). Use the BorderStyle, ButtonGlyphs, Caption, and Prompt properties to adjust the appearance of the dialog box. Use the SearchField property to specify which field will be used for keyboard searching.

Use the Execute method to display the dialog box. When the Execute method returns, the current record of the Dataset will be positioned at the record selected by the user. The edit field used for keyboard searching can be initialized with the contents of a TEdit component or a TRzButtonEdit component using the SearchEdit and SearchBtnEdit properties, respectively.

Properties

About
BorderStyle
ButtonGlyphs
Caption
Dataset
Font
Height
HelpContext
MoveSearchField
Name
NumbersOnly
Prompt
SearchBtnEdit
SearchEdit
SearchField
SearchString
Tag
Width

Methods

Create
Destroy
Execute

TRzLauncher Component

Unit RzLaunch **Inherits from** TComponent

Description

Use a TRzLauncher component to execute (or launch) another application, or process, from within a Delphi program. Set the FileName property to the file to be executed, and specify any command line parameters using the Parameters property. The selected program will be started when the Launch method is called. The launched application will start in the same directory as the executable file or in the directory referenced in the StartDir property, if specified. The initial state of the launched application's main window can be specified using the ShowMode property. Valid values include smNormal, smMaximized, and smMinimized. When the launched application terminates, the OnFinished event is generated.

Properties

About
Action
ExitCode
FileName
HInstance
HProcess
Name
Parameters
ShowMode
StartDir
Tag
Timeout

Methods

Create
Destroy
Launch

Events

OnFinished
OnTimeout

TRzSendMessage Component

Unit RzSndMsg **Inherits from** TComponent

Description
The TRzSendMessage component uses the Simple Messaging API (MAPI) functions (MAPI.DLL and MAPI32.DLL) to send mail messages. The Subject and MessageText properties are used to specify the message subject and message body, respectively. For added simplicity, the SubjectEdit and MessageMemo properties can be used to automatically populate the subject and message from an edit field and a memo component, respectively.

This component supports multiple TO and CC recipient lists through the ToRecipients and CcRecipients string list properties. Any number of attached files can be sent along with the message by adding each file name to the Attachments string list property. Use the Send method to send the message.

Properties

- About
- Attachments
- CcRecipients
- MessageMemo
- MessageText
- Name
- Password
- ProfileName
- Review
- Subject
- SubjectEdit
- Tag
- ToRecipients

Methods

- Create
- Destroy
- Logoff
- Logon
- Send

TRzRapidFireButton Component

Unit RzSpnEdt **Inherits from** TSpeedButton

Description
This component provides all of the features of the standard TSpeedButton component. However, instead of generating a single OnClick event when the button is pressed, TRzRapidFireButton continues to generate OnClick events as long as the button is in the down state. When the button is depressed, the first OnClick event is generated after an InitialDelay. After which, OnClick events are continually generated at intervals defined by the Delay property.

Properties

- About
- AllowAllUp
- Caption
- Cursor
- Delay
- Down
- Enabled
- Font
- Glyph
- GroupIndex
- Height
- Hint
- InitialDelay
- Layout
- Left
- Margin
- Name
- NumGlyphs
- ParentFont
- ParentShowHint
- ShowHint
- Spacing
- Tag
- Top
- Visible
- Width

Methods

- Click
- Create
- Destroy

Events

- OnClick
- OnDblClick
- OnMouseDown
- OnMouseMove
- OnMouseUp

StartYPos Property

Applies to
TRzRadioGroup, TRzDBRadioGroup

Declaration

```
property StartYPos : Integer;
```

Description
Use this property to specify the starting Y position for the first radio button in a radio group. All other radio buttons will be positioned with respect to this value.

TRzRadioGroup Component

Unit RzRadGrp **Inherits from** TRzCustomRadioGroup

Description

This component provides all of the features of the standard TRadioGroup, except that there are many additional ways to customize the appearance of the group box. For example, by descending from TRzPanel, this component inherits the border properties. However, to utilize the border properties, the GroupStyle property must be set to gsCustom. To change the font of the radio buttons and not the Caption, use the ItemFont property. To change the position of the radio buttons, use the Columns, StartXPos, StartYPos, and VerticalSpacing properties.

Properties

About
Align
BevelWidth
BorderColor
BorderInner
BorderOuter
BorderSides
BorderWidth
Caption
Color
Columns
Ctl3D
Cursor
DragCursor
DragMode
Enabled
Font
FrameSides
GroupStyle
Height
HelpContext
Hint
ItemFont
ItemHeight
ItemIndex
Items
Left
Locked
Name
ParentColor
ParentCtl3D
ParentFont
ParentShowHint
PopupMenu
ShowHint
StartXPos
StartYPos
TabOrder
TabStop
Tag
Top
VerticalSpacing
Visible
Width

Methods

Create
Destroy

Events

OnClick
OnDblClick
OnDragDrop
OnDragOver
OnEndDrag
OnEnter
OnExit
OnMouseDown
OnMouseMove
OnMouseUp
OnResize
32 OnStartDrag

TRzDBRadioGroup Component

Unit RzDBRGrp **Inherits from** TRzCustomRadioGroup

Description
Data-aware version of TRzRadioGroup. Provides standard DataField and DataSource properties for connecting to a column in a database table.

Properties

- About
- Align
- BevelWidth
- BorderColor
- BorderInner
- BorderOuter
- BorderSides
- BorderWidth
- Caption
- Color
- Columns
- Ctl3D
- Cursor
- DataField
- DataSource
- DragCursor
- DragMode
- Enabled
- Field
- Font
- FrameSides
- GroupStyle
- Height
- HelpContext
- Hint
- ItemFont
- ItemHeight
- ItemIndex
- Items
- Left
- Locked
- Name
- ParentColor
- ParentCtl3D
- ParentFont
- ParentShowHint
- PopupMenu
- ReadOnly
- ShowHint
- StartXPos
- StartYPos
- TabOrder
- TabStop
- Tag
- Top
- Value
- Values
- VerticalSpacing
- Visible
- Width

Methods

- Create
- Destroy

Events

- OnChange
- OnClick
- OnDblClick
- OnDragDrop
- OnDragOver
- OnEndDrag
- OnEnter
- OnExit
- OnMouseDown
- OnMouseMove
- OnMouseUp
- OnResize
- OnStartDrag

TRzSpinEdit Component

Unit RzSpnEdt **Inherits from** TWinControl

Description
The TRzSpinEdit component combines a edit field with two buttons forming a spinner. The edit portion can only accept numeric data and the buttons allow the user to increment and decrement the value. This component can accept integer values as well as floating point values. To only allow integers, set the IntegersOnly property to True. The Increment property is used to specify the amount the Value is incremented or decremented when one of the buttons is pressed. Increment can be a floating point value. PageSize is similar to the Increment except that PageSize is used when the user presses the PgUp or PgDn keys. The IntValue property is useful when the IntegersOnly property is set to True. IntValue automatically casts the Value to an Integer, thus eliminating this task for the developer.

To override the default glyphs used for the buttons, use the ButtonDownGlyph and ButtonUpGlyph properties. To change the width of the buttons, use the ButtonWidth property.

Properties

- About
- AllowKeyEdit
- ButtonDownGlyph
- ButtonDownNumGlyphs
- ButtonUpGlyph
- ButtonUpNumGlyphs
- ButtonWidth
- Color
- Ctl3D
- Cursor
- Decimals
- DragCursor
- DragMode
- Enabled
- Font
- Height
- HelpContext
- Hint
- Increment
- IntegersOnly
- IntValue
- Left
- Max
- Min
- Name
- PageSize
- ParentColor
- ParentFont
- ParentShowHint
- PopupMenu
- ReadOnly
- ShowHint
- TabOrder
- TabStop
- Tag
- Top
- Value
- Visible
- Width

Methods

- Create
- Destroy

Events

- OnChange
- OnChanging
- OnClick
- OnDragDrop
- OnDragOver
- OnEndDrag
- OnEnter
- OnExit
- OnKeyDown
- OnKeyPress
- OnKeyUp
- OnMouseDown
- OnMouseMove
- OnMouseUp
- OnStartDrag

TRzDBSpinEdit Component

Unit RzDBSpin **Inherits from** TRzSpinEdit

Description
Data-aware version of TRzSpinEdit. Provides standard DataField and DataSource properties for connecting to a column in a database table.

Properties

- About
- AllowKeyEdit
- ButtonDownGlyph
- ButtonDownNumGlyphs
- ButtonUpGlyph
- ButtonUpNumGlyphs
- ButtonWidth
- Color
- Ctl3D
- Cursor
- DataField
- DataSource
- Decimals
- DragCursor
- DragMode
- Enabled
- Font
- Height
- HelpContext
- Hint
- Increment
- IntegersOnly
- IntValue
- Left
- Max
- Min
- Name
- PageSize
- ParentColor
- ParentFont
- ParentShowHint
- PopupMenu
- ReadOnly
- ShowHint
- TabOrder
- TabStop
- Tag
- Top
- Value
- Visible
- Width

Methods

- Create
- Destroy

Events

- OnChange
- OnChanging
- OnClick
- OnDragDrop
- OnDragOver
- OnEndDrag
- OnEnter
- OnExit
- OnKeyDown
- OnKeyPress
- OnKeyUp
- OnMouseDown
- OnMouseMove
- OnMouseUp
- OnStartDrag

TRzButtonEdit Component

Unit RzBtnEdt **Inherits from** TWinControl

Description

The TRzButtonEdit component combines a edit field with two buttons, a primary and an alternate. The two buttons are automatically aligned to the right of the edit field, and each one generates its own event when clicked (i.e. OnButtonClick and OnAltBtnClick). The TRzButtonEdit component is often used as a replacement for a combo box. That is, rather than dropping down a list of items, one of the buttons can invoke a dialog box that is used to lookup values. For example, the TRzLookupDialog and TRzDBLookupDialog components.

By default, only the primary button is visible, and each button has an ellipsis for a glyph. To display the alternate button, set the AltBtnVisible property to True. To change the button glyphs and appearance, use the ButtonXXX and AltBtnXXX properties.

Properties

- About
- AltBtnGlyph
- AltBtnKind
- AltBtnNumGlyphs
- AltBtnVisible
- AltBtnWidth
- ButtonGlyph
- ButtonKind
- ButtonNumGlyphs
- ButtonVisible
- ButtonWidth
- CharCase
- Color
- Ctl3D
- Cursor
- DragCursor
- DragMode
- Enabled
- Font
- Height
- HelpContext
- Hint
- Left
- MaxLength
- Name
- ParentColor
- ParentFont
- ParentShowHint
- PopupMenu
- ReadOnly
- ShowHint
- TabOrder
- TabStop
- Tag
- Text
- Top
- Visible
- Width

Methods

- Create
- Destroy

Events

- OnAltBtnClick
- OnButtonClick
- OnChange
- OnDragDrop
- OnDragOver
- OnEndDrag
- OnEnter
- OnExit
- OnKeyDown
- OnKeyPress
- OnKeyUp
- OnMouseDown
- OnMouseMove
- OnMouseUp
- 32 OnStartDrag

TRzDBButtonEdit Component

Unit RzDBBnEd **Inherits from** TRzButtonEdit

Description
Data-aware version of TRzButtonEdit. Provides standard DataField and DataSource properties for connecting to a column in a database table.

Properties

- About
- AltBtnGlyph
- AltBtnKind
- AltBtnNumGlyphs
- AltBtnVisible
- AltBtnWidth
- ButtonGlyph
- ButtonKind
- ButtonNumGlyphs
- ButtonVisible
- ButtonWidth
- CharCase
- Color
- Ctl3D
- Cursor
- DataField
- DataSource
- DragCursor
- DragMode
- Enabled
- Font
- Height
- HelpContext
- Hint
- Left
- MaxLength
- Name
- ParentColor
- ParentFont
- ParentShowHint
- PopupMenu
- ReadOnly
- ShowHint
- TabOrder
- TabStop
- Tag
- Text
- Top
- Visible
- Width

Methods

- Create
- Destroy

Events

- OnAltBtnClick
- OnButtonClick
- OnChange
- OnDragDrop
- OnDragOver
- OnEndDrag
- OnEnter
- OnExit
- OnKeyDown
- OnKeyPress
- OnKeyUp
- OnMouseDown
- OnMouseMove
- OnMouseUp
- OnStartDrag

TRzLineEdit Component

Unit RzLnEdit **Inherits from** TCustomEdit

Description

This component provides all of the features of a standard TEdit component. However, it does not appear as a standard box. Instead, the TRzLineEdit component is displayed as a single line. The LineStyle property controls the appearance of the line. This component is very useful in creating electronic forms that mimic a paper version.

Properties

About
AutoSelect
AutoSize
CharCase
Color
Cursor
DragCursor
DragMode
Enabled
Font
Height
HelpContext
HideSelection
Hint
Left
LineStyle
MaxLength
Name
OEMConvert
ParentColor
ParentCtl3D
ParentFont
ParentShowHint
PasswordChar
PopupMenu
ReadOnly
ShowHint
TabOrder
TabStop
Tag
Text
Top
Visible
Width

Methods

Create
Destroy

Events

OnChange
OnClick
OnDblClick
OnDragDrop
OnDragOver
OnEndDrag
OnEnter
OnExit
OnKeyDown
OnKeyPress
OnKeyUp
OnMouseDown
OnMouseMove
OnMouseUp
32 OnStartDrag

TRzDBLineEdit Component

Unit RzDBLnEd **Inherits from** TDBEdit

Description
Data-aware version of TRzLineEdit. Provides standard DataField and DataSource properties for connecting to a column in a database table.

Properties

- About
- AutoSelect
- AutoSize
- BorderStyle
- CharCase
- Color
- Ctl3D
- Cursor
- DataField
- DataSource
- DragCursor
- DragMode
- Enabled
- Field
- Font
- Height
- HelpContext
- Hint
- Left
- LineStyle
- MaxLength
- Name
- ParentColor
- ParentCtl3D
- ParentFont
- ParentShowHint
- PasswordChar
- PopupMenu
- ReadOnly
- ShowHint
- TabOrder
- TabStop
- Tag
- Top
- Visible
- Width

Methods

- Create
- Destroy

Events

- OnChange
- OnClick
- OnDblClick
- OnDragDrop
- OnDragOver
- OnEndDrag
- OnEnter
- OnExit
- OnKeyDown
- OnKeyPress
- OnKeyUp
- OnMouseDown
- OnMouseMove
- OnMouseUp
- OnStartDrag

TRzPanel Component Editor

This component editor provides direct access to many of the properties of a TRzPanel component. As each property is modified, its effect is immediately displayed in the preview area. Use the Border Width track bar to change the number of pixels between the inner and outer borders. The BorderSides and FrameSides properties can be altered using the two sets of check boxes. The frame style used for the inner and outer borders, as well as the color used to fill the area between the borders can be changed using the combo boxes in the Border Style group box.

TRzLabel Component Editor

This component editor provides direct access to many of the properties of a TRzLabel component. As each property is modified, its effect is immediately displayed in the preview area. The first page provides controls for modifying the standard font properties such as the font name, point size, color, and style. The second page provides controls for modifying the three-dimensional style of the label. For example, the type of 3D style can be selected (i.e. Normal, Raised, Recessed, and Shadow). If the Shadow style is selected, the Shadow Depth track bar is enabled allowing the user to change the depth by dragging the track thumb. The angle of the label can be modified using the track bar at the bottom of the page. Check the check box to restrict angles to 15 degree increments.

TRzToolbar Component Editor

The TRzToolbar component editor provides a fast and easy way to create speed buttons for a TRzToolbar component. Once the editor is displayed, speed buttons are created by simply pressing the desired button in the component editor. When a button is pressed, a new button is created on the toolbar. The new button is positioned to the right of the last control in the toolbar. If the Insert Spacer check box is checked, then a spacer of 8 pixels will be inserted before the new button is created. The Insert Spacer check box is turned off after a button is selected. Once created, the glyph and hint of the new button are set to the values used in the editor. The editor stays open until the Done button is pressed. This way, multiple buttons can be created without having to reopen the editor.

TRzSplitter Component Editor

This component editor provides direct access to many of the properties of a TRzSplitter component. As each property is modified, its effect is immediately displayed in the preview area. Change the orientation of the splitter by selecting the desired setting in the Orientation group box. The Real Time Drag check box controls how the splitter updates its display when the splitter bar is moved. When the Real Time Drag option is cleared and the user moves the splitter bar, a mask is displayed showing the new position of the bar, and when the user releases the mouse, the bar is repositioned. When the Real Time Drag option is checked, the bar is repositioned as the user moves the mouse.

The top portion of the component editor is broken up into four pages. Each page provides access to modifying the border styles used for the splitter, the splitter bar, and the two panes. In addition, each of the panes can be made invisible by clearing the Visible check box on the appropriate page in the notebook.

TRzCheckList Component Editor

This component editor is used to add, change, and delete items from a TRzCheckList component at design-time. Simply press the button corresponding to the desired operation and follow the prompts. To change the state of a list item, simply click in the check box associated with that item. To disable an item, first selected the item, and then clear the Enabled check box. Items in the list can be rearranged by using the Move Up and Move Down buttons.

Raize String List Editor

The Raize String List Editor is a replacement for the standard string list editor. Unlike the standard editor, this version can be resized and customized. That is, the font type, style, and size can be customized. Even the tab stop size can be modified. To make your customizations permanent, check the Default check box before closing the editor. This ensures that your settings will be used the next time the editor is displayed. All settings are stored in the Delphi.ini file for Delphi 1.02 or the Windows Registry for Delphi 2.0.

The toolbar provides quick access to file operations, printing, and clipboard operations. This version also supports Indent and Unindent features. Simply select the desired text, or place the cursor on a particular line, and press either the Indent or Unindent buttons. The selected text will be repositioned to the next (or prior) tab stop.

EMapiUserAbort Object

Declaration

```
EMapiUserAbort = class( EAbort );
```

Description

This silent exception is raised when a user aborts sending a message from the MAPI compose dialog box after it was originally sent using a TRzSendMessage component.

EMapiError Object

Declaration

```
EMapiError = class( Exception )
  ErrorCode : Integer;
end;
```

Description

This exception is raised by a TRzSendMessage component when a problem occurs in executing a function in the *Messaging API*.   The error code returned from the function is stored in the ErrorCode field.

ELaunchError Object

Declaration

```
ELaunchError = class( Exception )
  ErrorCode : Integer;
end;
```

Description

This exception is raised when a TRzLauncher component's Launch method is called, and a problem is experienced when trying to launch the desired application.   The ErrorCode field gets populated with the error code returned from the operating system.

EInvalidSearchField Object

Declaration

```
EInvalidSearchField = class( Exception );
```

Description

This exception is raised when a TRzDBLookupDialog component's Execute method is called, and the SearchField property is **nil**.

EInvalidDataset Object

Declaration

```
EInvalidDataset = class( Exception );
```

Description

This exception is raised when a TRzDBLookupDialog component's Execute method is called, and the Dataset property is **nil**.

About Property

Applies to
All Raize Components

Declaration

```
property About : TRzAboutInfo;
```

Description
This property serves as a place holder for a custom property editor that displays a brief description of the component as well as copyright information. The dialog box that is displayed by the property editor is only available at design-time and **is not** linked into an application.

BorderColor Property

Applies to
TRzBorder, TRzPanel, TRzStatusBar, TRzToolbar, TRzProgressBar, TRzDBProgressBar, TRzRadioGroup, TRzDBRadioGroup

Declaration

```
property BorderColor : TColor;
```

Description
Use this property to specify the color used to fill the region between the inner and outer borders.

BorderInner Property

Applies to

TRzBorder, TRzPanel, TRzStatusBar, TRzToolbar, TRzProgressBar, TRzDBProgressBar, TRzRadioGroup, TRzDBRadioGroup, TRzSplitter

Declaration

```
property BorderInner : TFrameStyle;
```

Description

Use this property to change the style of the inner border used by a component.   There are 10 possible styles to choose from, including the new Windows 95 user interface styles.

Other Types

TAlignmentVertical
TBarStyle
TButtonKind
TDisplayStyle
TDrawTickEvent
TFontDevice
TFontType
TFrameStyle
TGlyphAlignment
TGroupStyle
TLineStyle
TOrientation
TPointerType
TPositionChangingEvent
TProgressChangeEvent
TResourceType
TSegmentRange
TSelectedPane
TShowMode
TSides
TSpinButtonEvent
TSpinChangingEvent
TStateChangingEvent
TTextStyle
TThumbStyle
TTickStyle
TToggleKey

BorderOuter Property

Applies to
TRzBorder, TRzPanel, TRzStatusBar, TRzToolbar, TRzProgressBar, TRzDBProgressBar, TRzRadioGroup, TRzDBRadioGroup, TRzSplitter

Declaration

```
property BorderOuter : TFrameStyle;
```

Description
Use this property to change the style of the outer border used by a component.   There are 10 possible styles to choose from, including the new Windows 95 user interface styles.

BorderSides Property

Applies to
TRzBorder, TRzPanel, TRzStatusBar, TRzToolbar, TRzRadioGroup, TRzDBRadioGroup

Declaration
```
property BorderSides : TSides;
```

Description
Use this property to select which sides of the component will display a border. For example, create a Windows 3.1 style toolbar by setting BorderSides to [sdTop, sdBottom] while the BorderOuter property is set to fsPopup and BorderInner to fsNone. For added effect, set the FrameSides property to [sdBottom].

OnMouseEnter Event

Applies to
TRzBorder, TRzPanel, TRzStatusBar, TRzStatusPane, TRzDBStatusPane, TRzGlyphStatus, TRzDBStateStatus, TRzClockStatus, TRzKeyStatus, TRzResourceStatus, TRzLabel, TRzDBLabel

Declaration

```
property OnMouseEnter : TNotifyEvent;
```

Description
The event occurs whenever the mouse pointer is moved into the region defined by the component.

OnMouseLeave Event

Applies to
TRzBorder, TRzPanel, TRzStatusBar, TRzStatusPane, TRzDBStatusPane, TRzGlyphStatus, TRzDBStateStatus, TRzClockStatus, TRzKeyStatus, TRzResourceStatus, TRzLabel, TRzDBLabel

Declaration

```
property OnMouseLeave : TNotifyEvent;
```

Description
The event occurs whenever the mouse pointer is moved out of the region defined by the component.

Action Property

Applies to
TRzLauncher

Declaration

```
property Action : string;
```

Description
Use this property to specify the operation that is to be performed. By default, the action performed is *open*. The other common operation is *print*.

FileName Property

Applies to
TRzLauncher

Declaration

```
property FileName : string;
```

Description
Use this property to designate which file to execute. If a full path is not specified, the current search path is used.

NOTE: The TRzLauncher uses the Windows ShellExecute function to launch the application in Delphi 1.0, and ShellExecuteEx under Delphi 2.0. This means that the FileName can be set to any file that has a corresponding association registered with Windows. For example, setting FileName to ‘README.TXT’ will invoke the NotePad when launched.

Parameters Property

Applies to
TRzLauncher

Declaration

```
property Parameters : string;
```

Description
This property is used to specify the parameters that are passed to the selected application. For example, if the FileName property is set to "NOTEPAD.EXE," the Parameters property could be set to "WIN.INI." In this case, when the application is launched, the WIN.INI file will be loaded into NotePad.

ShowMode Property

Applies to
TRzLauncher

Declaration

```
property ShowMode : TShowMode;
```

Description
The initial state of the launched applications main window can be specified using this property. The application can be started in its normal state, maximized, minimized, or hidden.

StartDir Property

Applies to
TRzLauncher

Declaration

```
property StartDir : string;
```

Description
This property can be used to change the default starting directory for the application. If empty, the application starts in the same directory that contains the executable file.

Timeout Property

Applies to
TRzLauncher

Declaration

```
property Timeout : Integer;
```

Description
This property is only available under Win32. Use this property to specify a time limit (in milliseconds) to wait for the launched process to terminate. If set to INFINITE (-1), the launcher will wait indefinitely for the process to terminate. When the timeout limit is reached, the OnTimeout event is generated.

OnFinished Event

Applies to
TRzLauncher

Declaration
```
property OnFinished : TNotifyEvent;
```

Description
This event is generated when the launched application is terminated. This gives the application that launched the program a way of determining when the other application ends.

OnTimeout Event

Applies to
TRzLauncher

Declaration

```
property OnTimeout : TNotifyEvent;
```

Description
This event occurs whenever a process launched by TRzLauncher takes longer than the Timeout value to terminate.

Launch Method

Applies to
TRzLauncher

Declaration

```
procedure Launch;
```

Description
Under Windows 3.1, this method uses the ShellExecute API function to execute the application specified by the FileName property. If the launch was successful, the HInstance property is set to the new application's instance handle.

Under Windows 95 or Windows NT, this method uses the ShellExecuteEx API function to execute the application specified by the FileName property. If the launch was successful, the HProcess property is set to the handle of the new process.

HInstance Property

Applies to
TRzLauncher

Declaration

```
property HInstance : THandle;
```

Description
Run-time and read only. When the selected application is launched, this property provides access to the application instance handle of the currently launched program. When the application terminates (and before one is launched), this property is set to zero.

FrameSides Property

Applies to
TRzPanel, TRzStatusBar, TRzToolbar, TRzRadioGroup, TRzDBRadioGroup

Declaration

```
property FrameSides : TSides;
```

Description
Use this property to specify which sides of the component will be framed with a black line. For example, by default, under Windows 3.1, the TRzStatusBar component sets the FrameSides property to [sdTop] to draw a black line between the status bar and the client area.

HProcess Property

Applies to
TRzLauncher

Declaration

```
property HProcess : THandle;
```

Description
This read-only property is only available under Win32, and is populated with the handle of the process once it is launched, otherwise it is zero.

ExitCode Property

Applies to
TRzLauncher

Declaration
```
property ExitCode : DWord;
```

Description
This read-only property is only available under Win32, and is populated with the return value of the launched process when it terminates.

AlignmentVertical Property

Applies to
TRzPanel

Declaration

```
property AlignmentVertical : TAlignmentVertical;
```

Description
Use this property to change the vertical alignment of the caption.   The caption can be positioned at the top, center, or bottom of the component.

RealTimeDrag Property

Applies to
TRzSplitter

Declaration

```
property RealTimeDrag : Boolean;
```

Description
This property controls the appearance of the splitter component when the user moves the splitter bar. By default, RealTimeDrag is False and thus a mask is displayed showing the new position of the bar, and when the user releases the mouse, the bar is repositioned. When RealTimeDrag is set to True, the bar is repositioned as the user moves the mouse.

Position Property

Applies to
TRzSplitter, TRzTrackBar, TRzDBTrackBar

Declaration

```
property Position : Integer;
```

Description
For splitters, this property indicates the current position of the splitter bar.   That is, the distance from either the left or top of the component depending on the orientation of the component.

For track bars, this property indicates the current position of the thumb along the track.

SelectedPane Property

Applies to
TRzSplitter

Declaration

```
property SelectedPane : TSelectedPane;
```

Description
This property indicates the currently selected pane in a splitter component. It is rarely used at runtime, but is extremely valuable at design-time. Since there are two panes in a splitter that can serve as containers for other components, pasting controls previously copied to the clipboard requires selecting the destination pane before pasting the controls. The currently selected pane has a dashed line around its perimeter. To change the selected pane, use the SelectedPane property or select the desired pane from the splitter's popup menu.

TPositionChangingEvent Type

Declaration

```
type
  TPositionChangingEvent = procedure ( Sender : TObject;
                                       NewPos : Integer;
                                       var AllowChange : Boolean ) of object;
```

OnChanging Event

Applies to
TRzSplitter, TRzTrackBar, TRzDBTrackBar, TRzCheckList, TRzSpinEdit, TRzDBSpinEdit

Declaration
For Splitters and Track Bars

```
property OnChanging : TPositionChangingEvent;
```

For Check Lists

```
property OnChanging : TStateChangingEvent;
```

For Spin Edits

```
property OnChanging : TSpinChangingEvent;
```

Description
For splitters and track bars, this event occurs whenever the user repositions the splitter bar of a splitter component or the thumb of a track bar. The new position is passed to an event handler. To prevent the component from moving to the new position, set the AllowChange parameter to False.

For check lists, this event occurs whenever the user changes the state of the check box associated with an item in a TRzCheckList component. The index of the selected list item and the new state are passed to an event handler. To prevent the state from changing, set the AllowChange parameter to False.

For spin edits, this event occurs whenever the user presses either the up or down button to change the value of a spin edit component. To prevent the value from being changed, set the AllowChange parameter to False.

SplitterWidth Property

Applies to
TRzSplitter

Declaration

```
property SplitterWidth : Word;
```

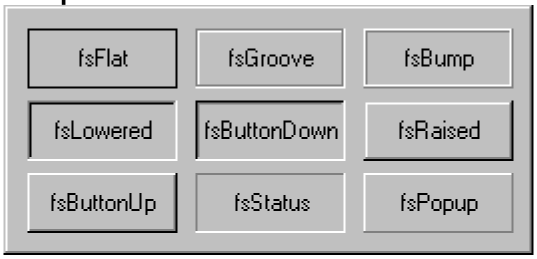
Description
Use this property to specify the width (in pixels) of the splitter bar in a TRzSplitter component.

TFrameStyle Type

Declaration

```
type
  TFrameStyle = ( fsNone, fsFlat, fsGroove, fsBump,
                  fsLowered, fsButtonDown,
                  fsRaised, fsButtonUp,
                  fsStatus, fsPopup );
```

Samples



SplitterStyle Property

Applies to
TRzSplitter

Declaration

```
property SplitterStyle : TFrameStyle;
```

Description
Use this property to change the style of the border used for the splitter bar.   There are 10 possible styles to choose from, including the new Windows 95 user interface styles.

TextStyle Property

Applies to
TRzLabel, TRzDBLabel

Declaration

```
property TextStyle : TTextStyle;
```

Description
Use this property to specify the visual style of the label. The four possible values are: tsNone, tsRaised, tsRecessed, and tsShadow.

ShowDivider Property

Applies to
TRzToolbar

Declaration
```
property ShowDivider : Boolean;
```

Description
Use this property to show or hide the divider line of a TRzToolbar component. Under Windows 95 or Windows NT 4.0, the divider line is on by default. The divider line is a groove that separates the toolbar from the main menu or from other toolbars.

TOrientation Type

Declaration

```
type
  TOrientation = ( orHorizontal, orVertical );
```

Orientation Property

Applies to
TRzSplitter, TRzTrackBar, TRzDBTrackBar, TRzProgressBar, TRzDBProgressBar

Declaration
```
property Orientation : TOrientation;
```

Description
Use this property to specify whether a component has horizontal or vertical orientation.

Percent Property

Applies to
TRzProgressBar, TRzDBProgressBar

Declaration
```
property Percent : Word;
```

Description
The Percent property determines how much of the progress area gets filled. The Percent property can be set directly, or indirectly using the TotalParts and PartsComplete properties.

For TRzDBProgressBar, the Percent property is only available at runtime and is read-only.

OnChange Event

Applies to
TRzProgressBar, TRzDBProgressBar

Declaration

```
property OnChange : TProgressChangeEvent;
```

Description
This event occurs whenever the percentage value of the progress bar changes. The new percentage value is passed as a parameter to the event handler.

TotalParts Property

Applies to
TRzProgressBar, TRzDBProgressBar

Declaration

```
property TotalParts : Word;
```

Description
This property, when used with the PartsComplete property, allows a user to quickly adjust the progress without having to manually calculate the percentage value. First, set the TotalParts to the number of parts that define a task. When the progress needs to be adjusted, change the PartsComplete value. The IncParts and IncPartsByOne methods provide fast ways of changing the number of parts completed.

PartsComplete Property

Applies to
TRzProgressBar, TRzDBProgressBar

Declaration
```
property PartsComplete : Word;
```

Description
This property, when used with the TotalParts property, allows a user to quickly adjust the progress without having to manually calculate the percentage value. First, set the TotalParts to the desired value. When the progress needs to be adjusted, change the PartsComplete value. The IncParts and IncPartsByOne methods provide fast ways of changing these values.

InteriorOffset Property

Applies to
TRzProgressBar, TRzDBProgressBar

Declaration

```
property InteriorOffset : Byte;
```

Description
This property is used to make the progress area smaller than the size defined by the inner border. The InteriorOffset specifies the number of pixels inside the inner border.

Value Property

Applies to
TRzDBTrackBar, TRzSpinEdit, TRzDBSpinEdit

Declaration
For Data-Aware Track Bars
```
property Value : string;
```

For Spin Edits
```
property Value : Single;
```

Description
For track bars, this property represents the current contents of the field for the current record in the dataset. When the user changes the track bar thumb, the Value property changes to the item in the Values string list for that position. The Value property becomes the value of the field for the current record in the dataset.

For spin edits, this property is equal to the contents of the edit portion converted to a floating point value. If the IntegersOnly property is True, then using the IntValue property eliminates the need to manually perform the conversion to an Integer value.

UpperLeft Property

Applies to
TRzSplitter

Declaration

```
property UpperLeft : TRzPaneData;
```

Description
This property provides access to the upper pane for a vertically oriented splitter or the left pane for a horizontally oriented splitter.   This property provides design-time access to the inner and outer border styles, the border color, and the border width of the upper left pane.   This property also allows the border color and interior color of the pane to be changed.   The final design-time subproperty controls the visible state of the pane.

At runtime, the Pane subproperty can be used to access the underlying TRzSplitterPane component. This is useful, when you wish to create components dynamically on one of the panes.   Set the new component's Parent property to the reference returned by the UpperLeft.Pane property.

LowerRight Property

Applies to
TRzSplitter

Declaration
```
property LowerRight : TRzPaneData;
```

Description
This property provides access to the lower pane for a vertically oriented splitter or the right pane for a horizontally oriented splitter. This property provides design-time access to the inner and outer border styles, the border color, and the border width of the lower right pane. This property also allows the border color and interior color of the pane to be changed. The final design-time subproperty controls the visible state of the pane.

At runtime, the Pane subproperty can be used to access the underlying TRzSplitterPane component. This is useful, when you wish to create components dynamically on one of the panes. Set the new component's Parent property to the reference returned by the LowerRight.Pane property.

FillColor Property

Applies to
TRzStatusPane, TRzDBStatusPane, TRzGlyphStatus, TRzDBStateStatus, TRzClockStatus, TRzKeyStatus, TRzResourceStatus

Declaration

```
property FillColor : TColor;
```

Description
Use this property to specify the color used to fill in the interior of a status component.

Angle Property

Applies to
TRzLabel, TRzDBLabel

Declaration

```
property Angle : Integer;
```

Description
Use this property to specify the angle used to rotate the label.

ShadowDepth Property

Applies to
TRzLabel, TRzDBLabel

Declaration

```
property ShadowDepth : Integer;
```

Description
Use this property to specify how far down and to the right the shadow text will be displayed from the main text.   This property is only applicable when TextStyle is set to tsShadow.

TTextStyle Type

Declaration

```
type
  TTextStyle = ( tsNormal, tsRaised, tsRecessed, tsShadow );
```

CaptionOffset Property

Applies to
TRzStatusPane, TRzDBStatusPane, TRzGlyphStatus, TRzDBStateStatus, TRzClockStatus, TRzKeyStatus, TRzResourceStatus

Declaration

```
property CaptionOffset : Integer;
```

Description
This property determines how far away from the border the caption is positioned.   The property value indicates the number of pixels between the border and the caption.

FrameStyle Property

Applies to
TRzStatusPane, TRzDBStatusPane, TRzGlyphStatus, TRzDBStateStatus, TRzClockStatus, TRzKeyStatus, TRzResourceStatus

Declaration

```
property FrameStyle : TFrameStyle;
```

Description
Use this property to change the style of the border used by the status component. There are 10 possible styles to choose from, including the new Windows 95 user interface styles.

FieldLabel Property

Applies to
TRzDBStatusPane

Declaration
```
property FieldLabel : string;
```

Description
Use this property to specify a static text string that is positioned to the left of the value stored in the dataset.

GlyphOffset Property

Applies to
TRzGlyphStatus, TRzDBStateStatus

Declaration

```
property GlyphOffset : Integer;
```

Description
This property determines how far away from the border the glyph is positioned.   The property value indicates the number of pixels between the border and the glyph.

GlyphAlignment Property

Applies to
TRzGlyphStatus, TRzDBStateStatus

Declaration

```
property GlyphAlignment : TGlyphAlignment;
```

Description
This property determines the location of a glyph displayed within a status control. The glyph can be placed either to the left (the default) or to the right of the caption.

ShowCaption Property

Applies to
TRzDBStateStatus

Declaration
```
property ShowCaption : Boolean;
```

Description
Use this property to hide and show the caption of a TRzDBStateStatus component.

Active Property

Applies to
TRzClockStatus, TRzKeyStatus, TRzResourceStatus

Declaration

```
property Active : Boolean;
```

Description
This property indicates that the shared polling timer used by all Raize polling status components is generating timer events. Set this property to False to disable the shared timer. All status components relying on a timer (i.e. TRzClockStatus, TRzKeyStatus, TRzResourceStatus), will not received timer events to update their displays.

Format Property

Applies to
TRzClockStatus

Declaration
```
property Format : string;
```

Description
The Format property determines how the current date and time is displayed within the TRzClockStatus component. The string specified in this property is passed to the standard Delphi Format procedure. The following are some sample Format values and the corresponding status display:

Format String	Output
c	9/15/90 5:18:23 PM
dddddd	Saturday, September 15, 1990
m/d/yy	9/15/90
mm/dd/yyy	09/15/1990
t	5:18 PM
hh:nn a/p	05:18 p
h:nn:ss am/pm	5:18:35 pm

Interval Property

Applies to
TRzClockStatus, TRzKeyStatus, TRzResourceStatus

Declaration

```
property Interval : Word;
```

Description
The Interval property specifies the number of milliseconds between OnTimerExpired events in Raize Status Controls that utilize a shared polling mechanism by descending from TRzPollingStatus.

Key Property

Applies to
TRzKeyStatus

Declaration
`property Key : TToggleKey;`

Description
Use this property to specify the key to monitor.   The CapsLock, NumLock, and ScrLock keys can be monitored.

ShowPercent Property

Applies to
TRzResourceStatus, TRzProgressBar, TRzDBProgressBar

Declaration

```
property ShowPercent : Boolean;
```

Description
Controls whether or not the percentage value is displayed. This property has no effect when the BarStyle is set to bsLED.

BackColor Property

Applies to
TRzResourceStatus, TRzProgressBar, TRzDBProgressBar

Declaration

```
property BackColor : TColor;
```

Description
Use this property to specify the color used to fill the background of the progress area.   When the BarStyle is bsTraditional, the BackColor is also used to display the percent value text in the filled area.

BarColor Property

Applies to
TRzResourceStatus, TRzProgressBar, TRzDBProgressBar

Declaration

```
property BarColor : TColor;
```

Description
Use this property to specify the color used to fill the progress area. When the BarStyle is bsTraditional, the BarColor is also used to display the percent value text in the unfilled area.

BarStyle Property

Applies to
TRzResourceStatus, TRzProgressBar, TRzDBProgressBar

Declaration

```
property BarStyle : TBarStyle;
```

Description
The property controls the style used to fill the progress area. When set to bsTraditional, the progress area is filled using a solid bar progressing from left to right or bottom to top depending on the orientation. When set to bsLED, the progress area is filled with segments instead of a solid color.

TBarStyle Type

Declaration

```
type
  TBarStyle = ( bsTraditional, bsLED );
```

DisplayStyle Property

Applies to
TRzResourceStatus

Declaration

```
property DisplayStyle : TDisplayStyle;
```

Description
This property controls whether the amount of free resources is displayed using a progress bar or as a simple text value.

TDisplayStyle Type

Declaration

```
type
  TDisplayStyle = ( dsBar, dsText );
```

NumSegments Property

Applies to
TRzResourceStatus, TRzProgressBar, TRzDBProgressBar

Declaration
```
property NumSegments : TSegmentRange;
```

Description
The property is used to specify the number of segments it takes to fill the progress bar at 100%.   By default, NumSegments is set to 20.

ResourceType Property

Applies to
TRzResourceStatus

Declaration
```
property ResourceType : TResourceType;
```

Description
Use this property to specify which resource value to monitor.   Valid options are System, User, GDI, and Memory.

CustomThumb Property

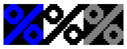
Applies to
TRzTrackBar, TRzDBTrackBar

Declaration

```
property CustomThumb : TBitmap;
```

Description
Use this property to specify a bitmap that defines a custom thumb for a track bar. The bitmap should contain three glyphs of the same width arranged in a row. The first glyph specifies the normal appearance of the thumb. Use black to fill transparent areas. The second glyph specifies the masking bitmap. This glyph only contains black and white pixels. Use black for areas defining the thumb. Use white for the transparent areas. The last glyph specifies the disable appearance of the thumb. It follows the same rules as the first glyph. The following figure gives an example of a custom thumb bitmap. The thumb is a percent sign, the areas around the percent sign and inside each circle are transparent.

Custom Thumb Sample



Max Property

Applies to
TRzTrackBar, TRzDBTrackBar, TRzSpinEdit, TRzDBSpinEdit

Declaration
For Track Bars
```
property Max : Integer;
```

For Spin Edits
```
property Max : Single;
```

Description
For track bars, use this property to specify the maximum position along the track.

For spin edits, use this property to specify the maximum value that can be entered.

Min Property

Applies to
TRzTrackBar, TRzDBTrackBar, TRzSpinEdit, TRzDBSpinEdit

Declaration
For Track Bars

```
property Min : Integer;
```

For Spin Edits

```
property Min : Single;
```

Description
For track bars, use this property to specify the minimum position along the track.

For spin edits, use this property to specify the minimum value that can be entered.

PageSize Property

Applies to
TRzTrackBar, TRzDBTrackBar, TRzSpinEdit, TRzDBSpinEdit

Declaration
For Track Bars

```
property PageSize : Word;
```

For Spin Edits

```
property PageSize : Single;
```

Description
For track bars, use this property to specify the amount the Position changes when either the PgUp or PgDn key is pressed.

For spin edits, use this property to specify the amount the Value property is incremented or decremented by when either the PgUp or PgDn key is pressed. Note that the PageSize property can be set to a floating point value, thus allowing non-integer increments.

PointerType Property

Applies to
TRzTrackBar, TRzDBTrackBar

Declaration

```
property PointerType : TPointerType;
```

Description
By default, the pointer style thumb is pointing up. To use the Windows 95 style of a down pointer, set this property to ptWin95.

TPointerType Type

Declaration

```
type
  TPointerType = ( ptStandard, ptWin95 );
```

ShowTicks Property

Applies to
TRzTrackBar, TRzDBTrackBar

Declaration

```
property ShowTicks : Boolean;
```

Description
This property determines whether tick marks are displayed along the track.   Setting this property to False is useful when the tick marks cannot be distinguished.

ThumbStyle Property

Applies to
TRzTrackBar, TRzDBTrackBar

Declaration

```
property ThumbStyle : TThumbStyle;
```

Description
Use the ThumbStyle property to change the appearance of the track thumb.   You can choose between three standard thumb styles (i.e. Pointer, Mixer, Box), or you can use a custom thumb by setting the CustomThumb property.

TThumbStyle Type

Declaration

```
type
  TThumbStyle = ( tsBox, tsCustom, tsMixer, tsPointer );
```

TickStyle Property

Applies to
TRzTrackBar, TRzDBTrackBar

Declaration

```
property TickStyle : TTickStyle;
```

Description
By default, TickStyle is set to tsStandard, which instructs the track bar to display small lines (or tick marks) at each position. For more control over how the tick marks are displayed, set TickStyle to tsOwnerDraw and write an event handler for the OnDrawTick event.

TTickStyle Type

Declaration

```
type
  TTickStyle = ( tkStandard, tkOwnerDraw );
```

TrackColor Property

Applies to
TRzTrackBar, TRzDBTrackBar

Declaration

```
property TrackColor : TColor;
```

Description
Use this property to specify the color used to fill the interior of the track groove.

TrackOffset Property

Applies to
TRzTrackBar, TRzDBTrackBar

Declaration

```
property TrackOffset : Word;
```

Description
Use this property to specify the number of pixels from the left or top the track is positioned in the track bar component.

TrackWidth Property

Applies to
TRzTrackBar, TRzDBTrackBar

Declaration

```
property TrackWidth : Word;
```

Description
Use this property to adjust the width of the track groove.   Defaults to 8.

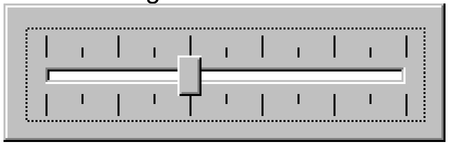
OnDrawTick Event

Applies to
TRzTrackBar, TRzDBTrackBar

Declaration
```
property OnDrawTick : TDrawTickEvent;
```

Description
This event occurs whenever a tick mark needs to be displayed. An OnDrawTick event handler is passed four parameters. The first parameter is a reference to the track bar. The next parameter is the canvas on which to draw the tick mark. A TPoint record is passed third. The X value holds the center location for the tick mark for a horizontal track bar. The Y value is used for a vertical track bar. The last parameter is the position number of the tick mark.

Example
The following handler creates a track bar with alternating long and short lines at each position:



```
procedure TForm1.RzTrackBar1DrawTick( TrackBar : TRzTrackBar;
  Canvas : TCanvas; Location : TPoint; Index : Integer );
begin
  with Canvas, Location do
  begin
    if Index mod 2 = 0 then
      MoveTo( X, 4 )
    else
      MoveTo( X, 10 );
    LineTo( X, 15 );
    if Index mod 2 = 0 then
      MoveTo( X, TrackBar.Height - 4 )
    else
      MoveTo( X, TrackBar.Height - 10 );
    LineTo( X, TrackBar.Height - 15 );
  end;
end;
```

TDrawTickEvent Type

Declaration

```
type
  TDrawTickEvent = procedure ( TrackBar : TRzTrackBar;
                               Canvas : TCanvas;
                               Location : TPoint;
                               Index : Integer ) of object;
```

IncParts Method

Applies to
TRzProgressBar, TRzDBProgressBar

Declaration

```
procedure IncParts( N : Integer );
```

Description
Increments the PartsComplete property by N.

TProgressChangeEvent Type

Declaration

```
type
  TProgressChangeEvent = procedure ( Sender : TObject;
                                     Percent : Integer ) of object;
```

Values Property

Applies to
TRzDBTrackBar

Declaration

```
property Values : TStrings;
```

Description
The Values string list property is populated with the values to be written to the database table.   After a set of values have been entered, the Min and Max positions of the track bar are updated according to the number of items in the list box.   At run-time, as the user moves the thumb of the track bar, the value from the Values list corresponding to the selected position is stored in the database field.

For example, suppose the Values list contains the items: *Small*, *Medium*, *Large*, *X-Large*.   And the component is attached to a column called *ShirtSize*.   As the user navigates through the table, the track bar thumb moves to the position corresponding to the value item   That is, if the record has a ShirtSize of Large, the thumb moves to the 3rd position.   Simply drag the thumb to change the shirt size.

BaseValue Property

Applies to
TRzDBProgressBar

Declaration

```
property BaseValue : Double;
```

Description
Use this property to specify a value to be used in calculating the percentage to be displayed in a TRzDBProgressBar. That is, set the DataField to a column in the table and set the BaseValue property to a static value. The percentage is calculated using the BaseValue using the following formula:

    Percent := DataField / BaseValue * 100

BaseField Property

Applies to
TRzDBProgressBar

Declaration

```
property BaseField : string;
```

Description
Use this property to specify a second field to be used in calculating the percentage to be displayed in a TRzDBProgressBar. That is, set the DataField to one column and the BaseField to another column. The percentage is calculated between the two columns using the following formula:

```
    Percent := DataField / BaseField * 100
```

BeepOnInvalidKey Property

Applies to
TRzListBox, TRzDBListBox, TRzTabbedListBox, TRzCheckList, TRzComboBox, TRzDBComboBox, TRzColorComboBox, TRzFontComboBox

Declaration

```
property BeepOnInvalidKey : Boolean;
```

Description
This property is used by all Raize Components that provide intuitive keyboard speed searching.   If the user presses an invalid key, the BeepOnInvalidKey property determines whether or not a beep is indicated.

SearchString Property

Applies to
TRzListBox, TRzDBListBox, TRzTabbedListBox, TRzCheckList, TRzComboBox, TRzDBComboBox, TRzColorComboBox, TRzFontComboBox, TRzLookupDialog, TRzDBLookupDialog

Declaration
```
property SearchString : string;
```

Description
For list boxes and combo boxes, this read-only property records the series of characters typed by the user while navigating through the list.   As the user types the SearchString property is built up, or taken down if the BackSpace key is pressed.   The list component then moves the selection bar to the item which most closely matches the SearchString.

For lookup dialog components, the SearchString property is used to specify an initial search string. When the dialog is executed, the SearchString, if specified, is copied to the edit field used for keyboard searching.   This causes the lookup dialog to try to find the item that most closely matches the characters specified in the search string.

OnStartDrag Event

This event is only available in Delphi 2.0.   To see the Delphi 2.0 help topic on this event, click here.

HorzScrollBar Property

Applies to
TRzTabbedListBox

Declaration
```
property HorzScrollBar : Boolean;
```

Description
When this property is set to True, a horizontal scroll bar will appear in the list box when the HorzExtent is greater than the width of the list box.

TabStops Property Editor

This property editor is used to add, change, and delete tab stops in a TRzTabbedListBox component at design-time. As tab stops are modified, the preview area is updated to reflect the current settings. Press the Add button to add a new tab stop. By default, the new tab stop is positioned 8 characters after the last tab stop. Select a tab stop from the list of tab stop values and press the Delete button to remove the tab stop, or use the track bar to change the position of the tab stop. If a tab stop is between two other tab stops, you will only be able to move the tab stop within that range.

UpdateTabStops Method

Applies to
TRzTabbedListBox

Declaration

```
procedure UpdateTabStops;
```

Description
This method is called whenever the TabStops list is modified.   This ensures that the underlying list box control uses the same tab stops specified in the TabStops property.

TabStops Property

Applies to
TRzTabbedListBox

Declaration
```
property TabStops : TRzTabStopList;
```

Description
Use this integer list to maintain the list of tab stops used by the TRzTabbedListBox.   The TabStops property is equivalent to the Items property, but instead of dealing with strings, TabStops utilizes integers. The custom property editor gives developers the ability to visually set tab stops at design-time

HorzExtent Property

Applies to
TRzTabbedListBox

Declaration
```
property HorzExtent : Word;
```

Description
Use this property to specify the virtual width of the list box. The actual width of a list box is defined by its Width property and border size. The horizontal extent must be larger than the actual width in order to display a horizontal scroll bar.

ItemState Property

Applies to
TRzCheckList

Declaration
`**property** ItemState[ Index : Integer ] : TCheckBoxState;`

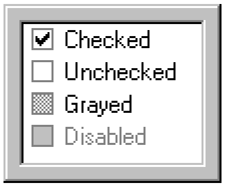
Description
Use this run-time only property to set or get the state of the check box associated with an item in a TRzCheckList component. A check box can be checked, unchecked, or grayed. However, in order to set a check box to the grayed state, the AllowGrayed property must be set to True. The Index value is zero based and corresponds to the index used in the Items property.

AllowGrayed Property

Applies to
TRzCheckList

Declaration
```
property AllowGrayed : Boolean;
```

Description
Setting this property to True allows the user to change a check box to the *grayed* state. By default, AllowGrayed is False, meaning that for each check box only the *checked* and *unchecked* states are allowed.



☑ Checked
□ Unchecked
▨ Grayed
■ Disabled

ItemEnabled Property

Applies to
TRzCheckList

Declaration

```
property ItemEnabled[ Index : Integer ] : Boolean;
```

Description
Use this run-time only property to enable or disable individual list item in a TRzCheckList component. Disabled list items appear in the disabled font style and the associated check box cannot be changed. The Index value is zero based and corresponds to the index used in the Items property.

SelectedColor Property

Applies to
TRzColorComboBox

Declaration
```
property SelectedColor : TColor;
```

Description
Use this property to obtain the TColor value corresponding to the color item selected in the component. Setting SelectedColor to a TColor value causes the component to change its selection to the corresponding color item.   Using this property is much easier than trying to manipulate the ItemIndex property.

OnCloseUp Event

Applies to
TRzComboBox, TRzColorComboBox, TRzFontComboBox

Declaration

```
property OnCloseUp : TNotifyEvent;
```

Description
This event occurs whenever the list portion of a combo box closes. This event is the opposite of OnDropDown.

ShowSysColors Property

Applies to
TRzColorComboBox

Declaration

```
property ShowSysColors : Boolean;
```

Description
Use this property to instruct a TRzColorComboBox component to display color values for all of the Windows defined System colors.   For example, clWindow, clBtnFace, and clHighlight.

DefaultColor Property

Applies to
TRzColorComboBox

Declaration

```
property DefaultColor : TColor;
```

Description
Use this property to specify the default color value used by a TRzColorComboBox is the user does not make a color selection.

FontDevice Property

Applies to
TRzFontComboBox

Declaration
```
property FontDevice : TFontDevice;
```

Description
The screen and printer each have their own list of available fonts.   This property allows a user to select which list is used to populate the combo box.

TFontDevice Type

Declaration

```
type
  TFontDevice = ( fdScreen, fdPrinter );
```

FontName Property

Applies to
TRzFontComboBox

Declaration

```
property FontName : string;
```

Description
Use this property to set the selected font name using the name of the font rather than a TFont value. This property is automatically updated if a new font is selected in the list.

FontSize Property

Applies to
TRzFontComboBox

Declaration

```
property FontSize : Integer;
```

Description
This property is used to specify the point size used in the TFont value returned by the SelectedFont property. When a new font is selected, only the name is chosen. However, a TFont object needs a font size, and this property allows a developer to override the default size of 8.

FontStyle Property

Applies to
TRzFontComboBox

Declaration

```
property FontStyle : TFontStyles;
```

Description
This property is used to specify the font style (e.g. bold, italic, etc.) used in the TFont value returned by the SelectedFont property. When a new font is selected, only the name is chosen. However, a TFont object needs a font style, and this property allows a developer to override the default style.

FontType Property

Applies to
TRzFontComboBox

Declaration
```
property FontType : TFontType;
```

Description
Use this property to restrict the fonts displayed in a TRzFontComboBox. This list can be restricted to True Type fonts, fixed-pitched fonts, or printer fonts. To remove any restriction, set FontType to ftAll.

TFontType Type

Declaration

```
type
  TFontType = ( ftAll, ftTrueType, ftFixedPitch, ftPrinter );
```

SelectedFont Property

Applies to
TRzFontComboBox

Declaration

```
property SelectedFont : TFont;
```

Description
Use this property to obtain a TFont value corresponding to the font name selected in the list. Setting SelectedFont to a TFont value causes the component to change its selection to the corresponding font name. Using this property is much easier than trying to manipulate the ItemIndex property.

LongDirName Property

Applies to
TRzDirectoryListBox

Declaration

```
property LongDirName : Boolean;
```

Description
Use this property to obtain the long version of the currently selected directory name in a TRzDirectoryListBox component.

ShowLongNames Property

Applies to
TRzDirectoryListBox, TRzFileListBox

Declaration
```
property ShowLongNames : Boolean;
```

Description
Under Delphi 1.0, this property controls whether long names are displayed when the application is run under an operating system that supports long file names. That is, Windows 95 or Windows NT.

LongFileName Property

Applies to
TRzFileListBox

Declaration

```
property LongFileName : string;
```

Description
Use this property to obtain the long version of the currently selected file name in a TRzFileListBox component.

ShortFileName Property

Applies to
TRzFileListBox

Declaration

```
property ShortFileName : string;
```

Description
This run-time only property can be used to obtain the short file name selected in a TRzFileListBox.

AllowCreate Property

Applies to
TRzSelDirDialog

Declaration

```
property AllowCreate : Boolean;
```

Description
Use this property to instruct the TRzSelDirDialog component to provide an edit field in which the user may enter a new directory name.   When the user presses the OK button, the selected directory is checked for existence.   If it exists, the dialog is closed.   If not, the user is prompted to verify the creation of the new directory.   Once accepted, the dialog component creates the directory and closes the dialog box.

ButtonGlyphs Property

Applies to
TRzSelDirDialog, TRzLookupDialog, TRzDBLookupDialog

Declaration

```
property ButtonGlyphs : Boolean;
```

Description
Set this property to True to display glyphs on the buttons of the dialog box.

Directory Property

Applies to
TRzSelDirDialog

Declaration

```
property Directory : string;
```

Description
Use this property to specify the initial directory displayed in a TRzSelDirDialog component.   After the dialog box is closed, this property contains the directory selected by the user.

Execute Method

Applies to
TRzSelDirDialog, TRzLookupDialog, TRzDBLookupDialog

Declaration

```
function Execute : Boolean;
```

Description
Displays the corresponding dialog box associated with the component. If the user presses the OK button in the dialog box, the Execute method returns True. If the user presses the Cancel button or closes the dialog box, the Execute method returns False.

Caption Property

Applies to
TRzLookupDialog, TRzDBLookupDialog

Declaration

```
property Caption : string;
```

Description
Use this property to specify the string used to populate the lookup dialog's title bar.

List Property

Applies to
TRzLookupDialog

Declaration

```
property List : TStrings;
```

Description
Use this property to populate the list box of a TRzLookupDialog component. The list can be populated at design-time using the string list editor, or at run-time by assigning another string list to the List property.

Prompt Property

Applies to
TRzLookupDialog, TRzDBLookupDialog

Declaration
```
property Prompt : string;
```

Description
Use this property to specify the string placed above the edit field in the lookup dialog components. This strings prompts the user what to enter.

SearchBtnEdit Property

Applies to
TRzLookupDialog, TRzDBLookupDialog

Declaration

```
property SearchBtnEdit : TRzButtonEdit;
```

Description
Use this property to specify a TRzButtonEdit component (or descendant) that will be used to initialize the edit field of the lookup dialog. This property is mutually exclusive with the SearchEdit property. That is, only one of these two properties can be specified at a time. If the SearchBtnEdit property is assigned to a TRzButtonEdit component, then the SearchEdit property is set to **nil**.

SearchEdit Property

Applies to
TRzLookupDialog, TRzDBLookupDialog

Declaration

```
property SearchEdit : TCustomEdit;
```

Description
Use this property to specify a TEdit component (or descendant) that will be used to initialize the edit field of the lookup dialog. This property is mutually exclusive with the SearchBtnEdit property. That is, only one of these two properties can be specified at a time. If the SearchEdit property is assigned to a TEdit component, then the SearchBtnEdit property is set to **nil**.

SelectedIndex Property

Applies to
TRzLookupDialog

Declaration

```
property SelectedIndex : Integer;
```

Description
Use this property to specify the index of the item in the list that is to be initially selected when the dialog box is displayed.   When the dialog box is closed, SelectedIndex indicates the item selected by the user.

Dataset Property

Applies to
TRzDBLookupDialog

Declaration
```
property Dataset : TDataset;
```

Description
Use this property to specify the dataset that will be used to populate the grid portion of a TRzDBLookupDialog component. Any dataset component can be used by the TRzDBLookupDialog component, including a TQuery. When the dialog is closed, the current record in the dataset is the record selected by the user.

Note: When using TQuery result sets, the query must be ordered by the search field in order for fast keyboard searching to be effective.

MoveSearchField Property

Applies to
TRzDBLookupDialog

Declaration
```
property MoveSearchField : Boolean;
```

Description
Set this property to True to move the field specified in the SearchField property to the first column of the grid in a TRzDBLookupDialog component.

NumbersOnly Property

Applies to
TRzDBLookupDialog

Declaration

```
property NumbersOnly : Boolean;
```

Description
Use this property to restrict entry into the edit portion of a TRzDBLookupDialog component to only numeric characters.

SearchField Property

Applies to
TRzDBLookupDialog

Declaration

```
property SearchField : string;
```

Description
Use this property to specify the field in the Dataset that will be used for keyboard searching.   When the lookup dialog is displayed, as the user enters characters into the edit field, the closest matching field is selected in the grid.

Logoff Method

Applies to
TRzSendMessage

Declaration
```
procedure Logoff;
```

Description
Terminates the current mail session.   Any successive MAPI function call will reinitiate the log on process.

Logon Method

Applies to
TRzSendMessage

Declaration

```
procedure Logon;
```

Description
This method uses the ProfileName and Password properties, if specified, to log on to the mail server. The end result is that a session ID is retrieved which is then used for all other MAPI function calls. If Logon is not called, the Logon process will be initiated during the first call to the Send method.

Send Method

Applies to
TRzSendMessage

Declaration

```
procedure Send;
```

Description
Calling this method instructs the TRzSendMessage component to package the message subject, text, recipient lists, and file attachments into the correct format required by the *Messaging API*. Once the message is formatted, it is sent using the MAPI protocol. If the message is not completely specified or the Review property is set to True, then the MAPI Compose dialog box is displayed so that the user can complete the message.

Attachments Property

Applies to
TRzSendMessage

Declaration
```
property Attachments : TStrings;
```

Description
This property is used to specify a list of files to be sent along with the message. Each string in the list holds the full path of the file to be attached.

CcRecipients Property

Applies to
TRzSendMessage

Declaration

```
property CcRecipients : TStrings;
```

Description
This property is used to specify a list of recipients to be *copied* on the message. Equivalent to specifying CC: in a letter.

MessageMemo Property

Applies to
TRzSendMessage

Declaration

```
property MessageMemo : TCustomMemo;
```

Description
If this property references a Memo component, the contents of that memo control are used to populate the MessageText property before the message is sent. By connecting this property to a Memo component at design-time, you do not have to worry about populating the MessageText property before calling Send.

MessageText Property

Applies to
TRzSendMessage

Declaration

```
property MessageText : TStrings;
```

Description
This string list is used to hold the contents of the message. The property can be directly modified using the standard string list functions, or it can be assigned to another string list using the Assign method.

Password Property

Applies to
TRzSendMessage

Declaration

```
property Password : string;
```

Description
Use this property to specify the password to be used during the Logon process.

ProfileName Property

Applies to
TRzSendMessage

Declaration

```
property ProfileName : string;
```

Description
Use this property to specify a Profile to be used by the *Messaging API* when the Send method is called. Typically, this property is left blank which instructs MAPI to use the default profile.

Review Property

Applies to
TRzSendMessage

Declaration
```
property Review : Boolean;
```

Description
If this property is set to **True** the common Compose MAPI dialog is displayed before the message is sent regardless of whether or not the message is completely specified. If this property is set to **False**, the common Compose MAPI dialog is only displayed if the mail message is not completely specified.

Subject Property

Applies to
TRzSendMessage

Declaration

```
property Subject : string;
```

Description
Use the property to specify the subject string for the message.

SubjectEdit Property

Applies to
TRzSendMessage

Declaration

```
property SubjectEdit : TCustomEdit;
```

Description
If this property references an Edit component, the contents of that edit control are used to populate the Subject property before the message is sent.   By connecting this property to an Edit component at design-time, you do not have to worry about populating the Subject property.

ToRecipients Property

Applies to
TRzSendMessage

Declaration

```
property ToRecipients : TStrings;
```

Description
This property is used to specify the list of recipients for the message.   Equivalent to specifying TO: in a letter.

Delay Property

Applies to
TRzRapidFireButton

Declaration

```
property Delay : Word;
```

Description
Use this property to specify the interval (in milliseconds) between OnClick events generated when a TRzRapidFireButton is depressed.

LineStyle Property

Applies to
TRzLineEdit, TRzDBLineEdit

Declaration

```
property LineStyle : TLineStyle;
```

Description
Use this property to specify the type of line used by a line edit component.   Valid options include: groove, bump, or flat.

GroupStyle Property

Applies to
TRzRadioGroup, TRzDBRadioGroup

Declaration

```
property GroupStyle : TGroupStyle;
```

Description
Use this property to specify whether a radio group uses the standard style for its border or a custom style. When the custom style is selected, the border's appearance is defined by the border properties.

InitialDelay Property

Applies to
TRzRapidFireButton

Declaration

```
property InitialDelay : Word;
```

Description
Use this property to specify the number of milliseconds before the first OnClick event is generated after a TRzRapidFireButton is depressed.

StartXPos Property

Applies to
TRzRadioGroup, TRzDBRadioGroup

Declaration

```
property StartXPos : Integer;
```

Description
Use this property to specify the starting X position for the first radio button in a radio group. All other radio buttons will be positioned with respect to this value.

VerticalSpacing Property

Applies to
TRzRadioGroup, TRzDBRadioGroup

Declaration

```
property VerticalSpacing : Integer;
```

Description
Use this property to specify the number of pixels used between rows of radio buttons in a radio group.

TGroupStyle Type

Declaration

```
type
  TGroupStyle = ( gsStandard, gsCustom );
```

ItemFont Property

Applies to
TRzRadioGroup, TRzDBRadioGroup

Declaration

```
property ItemFont : TFont;
```

Description
Use this property to specify the font used to display the radio buttons in a radio group.   This property allows the radio group items to use a different font from the Caption property.

IntValue Property

Applies to
TRzSpinEdit, TRzDBSpinEdit

Declaration

```
property IntValue : Integer;
```

Description
Use this read-only property to retrieve the Value property converted to an Integer value. Since the Value property is of type Single, using it in an Integer calculation will result in type-mismatch errors. The IntValue property eliminates the need to perform the typecast.

OnClick Event

Applies to
TRzSpinEdit, TRzDBSpinEdit

Declaration

```
property OnClick : TSpinButtonEvent;
```

Description
This event occurs whenever one of the buttons in a spin edit component gets pressed. The event handler is passed a parameter indicating which button (up or down) was pressed.

AllowKeyEdit Property

Applies to
TRzSpinEdit, TRzDBSpinEdit

Declaration
```
property AllowKeyEdit : Boolean;
```

Description
Set this property to True to allow a user to type in a numeric value into the edit portion. When this property is False, only the buttons or arrow keys can be used to modify the value.

ButtonDownGlyph Property

Applies to

TRzSpinEdit, TRzDBSpinEdit

Declaration

```
property ButtonDownGlyph : TBitmap;
```

Description

Use this property to override the default down arrow bitmap used for a spin edit component.

ButtonDownNumGlyphs Property

Applies to
TRzSpinEdit, TRzDBSpinEdit

Declaration

```
property ButtonDownNumGlyphs : Integer;
```

Description
Use this property to specify the number of glyphs embedded in the ButtonDownGlyph bitmap. All glyphs must be the same size and next to each other in a row. Valid values are 1 to 4. The default value is 1. The actual glyph displayed depends on the state of the button. The first glyph is used for the normal *Up* state. The second one is used when the button is *Disabled*, and the third is used for the *Down* state. The last glyph is used when the button can be kept in the *Down* state. This last glyph is not used by the spin edit components.

ButtonUpGlyph Property

Applies to
TRzSpinEdit, TRzDBSpinEdit

Declaration

```
property ButtonUpGlyph : TBitmap;
```

Description
Use this property to override the default up arrow bitmap used for a spin edit component.

ButtonUpNumGlyphs Property

Applies to
TRzSpinEdit, TRzDBSpinEdit

Declaration

```
property ButtonUpNumGlyphs : Integer;
```

Description
Use this property to specify the number of glyphs embedded in the ButtonUpGlyph bitmap. All glyphs must be the same size and next to each other in a row. Valid values are 1 to 4. The default value is 1. The actual glyph displayed depends on the state of the button. The first glyph is used for the normal *Up* state. The second one is used when the button is *Disabled*, and the third is used for the *Down* state. The last glyph is used when the button can be kept in the *Down* state. This last glyph is not used by the spin edit components.

ButtonWidth Property

Applies to
TRzSpinEdit, TRzDBSpinEdit, TRzButtonEdit, TRzDBButtonEdit

Declaration
```
property ButtonWidth : Integer;
```

Description
For spin edits, use this property to specify the width of both spin buttons.

For button edits, use this property to specify the width of the primary button.

Decimals Property

Applies to
TRzSpinEdit, TRzDBSpinEdit

Declaration

```
property Decimals : Byte;
```

Description
Use this property to specify the number of decimal places used by a spin edit when displaying floating point values.

Increment Property

Applies to
TRzSpinEdit, TRzDBSpinEdit

Declaration

```
property Increment : Single;
```

Description
Use this property to specify the amount the Value property is incremented or decremented by when one of the spin buttons is pressed or when either the up or down arrow key is pressed. Note that the Increment property can be set to a floating point value, thus allowing non-integer increments.

IntegersOnly Property

Applies to
TRzSpinEdit, TRzDBSpinEdit

Declaration

```
property IntegersOnly : Boolean;
```

Description
Set this property to True to restrict values to Integers. If keyboard entry is allowed via AllowKeyEdit, the user is not allowed to enter a decimal point. Therefore, the Decimals property has no effect when IntegersOnly is True.

TSpinButtonEvent Type

Declaration

```
type
  TSpinButtonType = ( sbUp, sbDown );
  TSpinButtonEvent = procedure( Sender : TObject;
                                Button : TSpinButtonType ) of object;
```

AltBtnGlyph Property

Applies to
TRzButtonEdit, TRzDBButtonEdit

Declaration

```
property AltBtnGlyph : TBitmap;
```

Description
Use this property to specify a new bitmap for the alternate button in a button edit component.

OnButtonClick Event

Applies to
TRzButtonEdit, TRzDBButtonEdit

Declaration
```
property OnButtonClick : TNotifyEvent;
```

Description
This event occurs whenever the primary button in a button edit component gets pressed.

AltBtnKind Property

Applies to
TRzButtonEdit, TRzDBButtonEdit

Declaration
`property AltBtnKind : TButtonKind;`

Description
Use this property to select between two standard button types, or specify a custom type. The two standard types are bkLookup and bkDropDown. When the bkLookup type is specified, the button is displayed with an ellipsis glyph, and when the bkDropDown type is selected, the button is displayed with the standard combo box drop down glyph. If the AltBtnGlyph is set to a new bitmap, the bkCustom type is automatically set.

TButtonKind Type

Declaration

```
type
  TButtonKind = ( bkCustom, bkLookup, bkDropDown );
```

AltBtnNumGlyphs Property

Applies to
TRzButtonEdit, TRzDBButtonEdit

Declaration

```
property AltBtnNumGlyphs : Integer;
```

Description
Use this property to specify the number of glyphs embedded in the AltBtnGlyph bitmap. All glyphs must be the same size and next to each other in a row. Valid values are 1 to 4. The default value is 1. The actual glyph displayed depends on the state of the button. The first glyph is used for the normal *Up* state. The second one is used when the button is *Disabled*, and the third is used for the *Down* state. The last glyph is used when the button can be kept in the *Down* state. This last glyph is not used by the button edit components.

AltBtnVisible Property

Applies to
TRzButtonEdit, TRzDBButtonEdit

Declaration

```
property AltBtnVisible : Boolean;
```

Description
Use this property to show or hide the alternate button in a button edit component.

AltBtnWidth Property

Applies to
TRzButtonEdit, TRzDBButtonEdit

Declaration

```
property AltBtnWidth : Integer;
```

Description
Use this property to specify the width of the alternate button in a button edit component.

ButtonGlyph Property

Applies to
TRzButtonEdit, TRzDBButtonEdit

Declaration

```
property ButtonGlyph : TBitmap;
```

Description
Use this property to specify a new bitmap for the primary button in a button edit component.

ButtonKind Property

Applies to
TRzButtonEdit, TRzDBButtonEdit

Declaration

```
property ButtonKind : TButtonKind;
```

Description
Use this property to select between two standard button types, or specify a custom type. The two standard types are bkLookup and bkDropDown. When the bkLookup type is specified, the button is displayed with an ellipsis glyph, and when the bkDropDown type is selected, the button is displayed with the standard combo box drop down glyph. If the ButtonGlyph is set to a new bitmap, the bkCustom type is automatically set.

ButtonNumGlyphs Property

Applies to
TRzButtonEdit, TRzDBButtonEdit

Declaration
```
property ButtonNumGlyphs : Integer;
```

Description
Use this property to specify the number of glyphs embedded in the ButtonGlyph bitmap. All glyphs must be the same size and next to each other in a row. Valid values are 1 to 4. The default value is 1. The actual glyph displayed depends on the state of the button. The first glyph is used for the normal *Up* state. The second one is used when the button is *Disabled*, and the third is used for the *Down* state. The last glyph is used when the button can be kept in the *Down* state. This last glyph is not used by the button edit components.

ButtonVisible Property

Applies to
TRzButtonEdit, TRzDBButtonEdit

Declaration

```
property ButtonVisible : Boolean;
```

Description
Use this property to show or hide the primary button in a button edit component.

OnAltBtnClick Event

Applies to
TRzButtonEdit, TRzDBButtonEdit

Declaration

```
property OnAltBtnClick : TNotifyEvent;
```

Description
This event occurs whenever the alternate button in a button edit component gets pressed.

👓 This is a *read-only* property or event.

This property or event is only available at *run-time*.

32	This property or event is only available under Win32.

TLineStyle Type

Declaration

```
type
  TLineStyle = ( lsNone, lsFlat, lsGroove, lsBump );
```

TSides Type

Declaration

```
type
  TSide = ( sdLeft, sdTop, sdRight, sdBottom );
  TSides = set of TSide;
```

TGlyphAlignment Type

Declaration

```
type
  TGlyphAlignment = ( gaLeft, gaRight );
```

TStateChangingEvent Type

Declaration

```
type
  TStateChangingEvent = procedure( Sender : TObject;
                                   Index : Integer;
                                   NewState : TCheckBoxState;
                                   var AllowChange : Boolean ) of object;
```

TSpinChangingEvent Type

Declaration

```
type
  TSpinChangingEvent = procedure( Sender : TObject;
                                  var AllowChange : Boolean ) of object;
```

TSelectedPane Type

Declaration

```
type
  TSelectedPane = ( spUpperLeft, spLowerRight );
```

TResourceType Type

Declaration

```
type
  TResourceType = ( rtSystem, rtUser, rtGDI, rtMemory );
```

TRzPaneData Type

Declaration

```
type
  TRzPaneData = class( TPersistent )
  public
    property Pane : TRzSplitterPane;
  published
    property BorderColor : TColor;
    property BorderInner : TFrameStyle;
    property BorderOuter : TFrameStyle;
    property BorderWidth : TBorderWidth;
    property Color : TColor;
    property Visible : Boolean;
  end;
```

TAlignmentVertical Type

Declaration

```
type
  TAlignmentVertical = ( avTop, avCenter, avBottom );
```

TToggleKey Type

Declaration

```
type
  TToggleKey = ( tkCapsLock, tkNumLock, tkScrollLock );
```

TSegmentRange Type

Declaration

```
type
  TSegmentRange = 1..100;
```

TShowMode Type

Declaration

```
type
  TShowMode = ( smNormal, smMaximized, smMinimized, smHide );
```

TRzPollingStatus Component

Unit RzStatus **Inherits from** TRzCustomStatusPane

Description
The TRzPollingStatus class is the base class for all status controls that need to utilize a polling mechanism to determine what information to display. This class introduces the OnTimerExpired event which gets triggered whenever an internal timer event occurs. The state of the internal timer can be controlled through the Active and Interval properties. When Active is True the internal timer continues to generate OnTimerExpired events. The Interval property specifies the number of milliseconds between events.

TRzCustomStatusPane Component

Unit RzStatus **Inherits from** TGraphicControl

Description

This is the base class for Raize status components. This class simply provides the ability to display a status border, which by default is fsStatus. The style of the border is controlled by the FrameStyle property. By default the frame is drawn two pixels away from the edge of the control. This distance can be changed by setting the BorderWidth property to a new value.

This base class also declares three protected properties. The Caption property specifies the text to be displayed within the status frame. The CaptionOffset property determines how far away from the frame the Caption will be drawn, and the Alignment property determines which side of the control the text is displayed.

State Property

Applies to
TRzKeyStatus

Declaration

```
property State : TToggleState;
```

Description
Read-only property. This property indicates the current state of the selected Key.

TToggleState Type

Declaration

```
type
  TToggleState = ( tsOn, tsOff );
```

Licensing Agreement

This software is protected by copyright law and international copyright treaty. Therefore, you must treat this software just like a book, except that you may copy it onto a computer to be used and you may make archive copies of the software for the sole purpose of backing up our software and protecting your investment from loss. The software may be moved from one computer to another, so long as there is no possibility of it being used by more than one person at a time.

Adding Users
You may add users by paying for a separate software package for each user you wish to add. You may also add users by purchasing a site-license, so long as the number of persons who are able to use the software at one time is not more than the number of authorized users specified in our package or license.

Transferring the Software
You may transfer all of your rights to use the software to another person, provided that you transfer to that person all of the software, diskettes, and documentation provided in this package (including this statement), and transfer or destroy all copies in any form. Remember, once you transfer the software, you no longer have any right to use it, and the person to whom it is transferred may use it only in accordance with the copyright law, international treaty, and this statement.

If you have purchased an upgrade version of the software, it constitutes a single product with the Raize Software Solutions software that you upgraded. For example, the upgrade and the software that you upgraded cannot both be available for use by two different people at the same time, and cannot be transferred separately, without written permission from Raize Software Solutions.

Except as provided in this statement, you may not transfer, rent, lease, lend, copy, modify, translate, sublicense, time-share, or electronically transmit or receive the software, media, or documentation.

Limited Warranty
Raize Software Solutions, Inc. warrants the physical media and physical documentation provided by Raize Software Solutions to be free of defects in materials and workmanship for a period of sixty (60) days from the original purchase date. If Raize Software Solutions receives notification within the warranty period of defects in materials or workmanship, and determines that such notification is correct, Raize Software Solutions will replace the defective media or documentation.

The entire and exclusive liability and remedy for breach of this limited warranty shall be limited to replacement of defective media or documentation and shall not include or extend to any claim for or right to recover any other damages, including but not limited to, loss of profit, data, or use of the software or special, incidental or consequential damages, or other similar claims, even if Raize Software Solutions has been specifically advised of the possibility of such damages. In no event will Raize Software Solutions' liability for any damages to you or any other person ever exceed the lower of the list price or the actual price paid for the package or the license to use the software, regardless of the form of the claim.

RAIZE SOFTWARE SOLUTIONS, INC. SPECIFICALLY DISCLAIMS ALL OTHER WARRANTIES, REPRESENTATIONS, OR CONDITIONS, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTY OR CONDITION OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. ALL OTHER IMPLIED TERMS ARE EXCLUDED.

Specifically, Raize Software Solutions makes no representation or warranty that the software or documentation are "error-free," or meet any user's particular standards, requirements, or needs. In all events, any implied warranty, representation, condition, or other term is limited to the physical media and documentation and is limited to the 60-day duration of the limited warranty.

Raize Software Solutions is not responsible for, and does not make any representation, warranty, or condition concerning product, media, software, or documentation not manufactured or supplied by Raize Software Solutions, such as third-parties' programs that are designed using Raize Software Solutions software or which include Raize Software Solutions programs or files.

General Terms that Apply to Compiled Programs
The license granted in this statement for you to create your own compiled programs and distribute your programs using the software in this package is subject to all of the following conditions:

- All copies of the programs you create must include a valid copyright notice, or Raize Software Solutions' copyright notice on your product label.

- You may not remove or alter any Raize Software Solutions' copyright, trademark, or other proprietary rights notice contained in any portion of Raize Software Solutions units, source code, or other files that bear such a notice.

- Raize Software Solutions provides no warranty at all to any person, other than the Limited Warranty provided to the original purchaser of this package.

- You will remain solely responsible to anyone receiving your programs for support, service, upgrades, or technical or other assistance, and such recipients will have no right to contact Raize Software Solutions for such services or assistance.

- You will indemnify, hold harmless, and defend Raize Software Solutions from and against any claims or lawsuits, including attorney's fees, that arise or result from the use, reproduction, or distribution or your programs.

- Your programs must be written using a licensed, registered copy of this Raize Software Solutions' product.

- You may not use Raize Software Solutions' name, logo, or trademarks to market your programs, except to state that your program was written using this Raize Software Solutions' product.

- All Raize Software Solutions' units, source code, and other files remain Raize Software Solutions' exclusive property.

Provisions for Visual Component Library Classes (i.e. Components)
Raize Software Solutions, Inc. grants you a non-exclusive royalty-free right to compile, reproduce, and distribute any new software programs created using the Components included in this package provided that you: (a) distribute the Components only in compiled executable programs; (b) do not use any part of the source code of the Components to build any other components for public distribution or commercial sale; and (c) do not use any of the Components as an object-oriented ancestor to build any other components (through inheritance) for public distribution or commercial sale.

Using Raize Components in Property Editors and Component Editors
Distributing a property editor or component editor that uses one or more of the Components from this package requires that the compiled unit (i.e. the DCU file) for the Components also be distributed. However, the compiled unit for the Component is sufficient for a user to install and use the Component, even though the user may be a non-licensed, non-registered user of Raize Components for Delphi. As a result, it is permissible to use the Raize Components in custom property editors and component editors. However, your users will not be able to receive technical support on the Raize Components included in your editor, unless the user is also a registered user of Raize Components for Delphi.

U.S. Government Restricted Rights

The Software and documentation are provided with Restricted Rights.   Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 or subparagraphs (c)(1) and (2) of the Commercial Computer Software--Restricted Rights at 48 CFR 52.227-19, as applicable. Contractor/manufacturer is Raize Software Solutions, Inc. at 2111 Templar Drive, Naperville, IL 60565.

This statement shall be construed, interpreted, and governed by the laws of the State of Illinois, U.S.A.

Technical Support

In addition to the *User's Guide* and the online help system, you can obtain technical support for *Raize Components for Delphi* by sending electronic mail to: support@raize.com.

Important: Technical support will only be provided to registered users.

On the Internet
Be sure to visit our Web site for product updates, tips and techniques, and upgrade information. Point your browser to http://www.raize.com.

Exceptions

EInvalidDataset
EInvalidSearchField
ELaunchError
EMapiError
EMapiUserAbort