

Report Interface Designer 1.0

Copyright © 1997 TRI-I Engineering, Inc.

Overview

Report Interface Designer is a programming utility for use with [Seagate Crystal Reports™](#) and a large number of application development tools.

Crystal Reports is a very powerful and full-featured reporting tool permitting rapid report development for many data sources. While Crystal Reports integrates with the most popular development tools, it still requires a fair amount of coding to connect to your application.

RID is designed for professional or corporate development teams who have a number of people specializing in each aspect of application development. Typically, the person who creates the Crystal Reports is different from the person who integrates that report into the application, possibly creating complex data entry screens to facilitate the passing of dynamic criteria to the report.

RID eases this type of team development and [Criteria Interface](#) dialog creation by permitting the same person who created the report to visually create the data input screens. When it comes time to integrate with your application, two lines of code are all that it takes.

RID embodies a very powerful report integration concept that is easy to use and will quickly become familiar to anyone who has previously used a visual development tool. This help file will assist in learning and using RID, and should be considered a companion reference to the RIDTutor Tutorial document and user guide.

While specifically designed for Borland Delphi 2.0/3 and Microsoft Visual Basic 5.0, RID can be used by any development tool that supports ActiveX controls. Additional help files specific to these programming tools were installed by the RID Setup program. Please consult these for further information on integrating the design component with your application.

If you are new to Report Interface Designer, start by reviewing the [RID Screen](#) and the design-time environment.

Align Property

Applies to

[TLabel](#), [TRadioGroup](#), [TListbox](#), [TGroupbox](#), [TPanel](#)

Declaration

property Align: (alNone, alTop, alBottom, alLeft, alRight, alClient);

Description

The Align property determines how the controls align within their TGroupbox or TPanel parent container.

Alignment property

Applies to

[TLabel](#), [TPanel](#)

Declaration

property Alignment: (taLeftJustify, taRightJustify, taCenter);

Description

The Alignment property specifies how text is aligned within the control.

AutoSelect property

Applies to

[TEdit](#)

Declaration

```
property AutoSelect: (True, False);
```

Description

The value of the AutoSelect property determines if the text in the TEdit control is automatically selected when the user tabs to the control. If AutoSelect is True, the text is selected. If AutoSelect is False, the text is not selected.

The default value is True.

AutoSize property

Applies to

[TLabel](#), [TEdit](#)

Declaration

```
property AutoSize: (True, False);
```

Description

When the AutoSize property is True, the height of the TLabel or TEdit control changes to accommodate font size changes to the text. When AutoSize is False, the size remains the same, regardless of any font changes. The default value is True.

BevelInner property

[See also](#)

Applies to

[TPanel](#)

Declaration

property BevelInner: (bvNone, bvLowered, bvRaised);

Description

A TPanel control has two bevels, an outer bevel drawn next to the border of the control, and an inner bevel drawn inside the outer bevel. The width of the outer bevel is determined by the value of the [BorderWidth](#) property.

BevelOuter property

[See also](#)

Applies to

[TPanel](#)

Declaration

property BevelOuter: (bvNone, bvLowered, bvRaised);

Description

A TPanel control has two bevels, an outer bevel drawn next to the border of the control, and an inner bevel drawn inside the outer bevel. The width of the outer bevel is determined by the value of the [BorderWidth](#) property.

BevelWidth property

[See also](#)

Applies to

[TPanel](#)

Declaration

```
property BevelWidth: Integer;
```

Description

The BevelWidth property determines the width in pixels between the inner and the outer bevels of a [TPanel](#) control.

BorderStyle property

Applies to

[TEdit](#), [TListbox](#), [TPanel](#)

Declaration

```
property BorderStyle: (bsNone, bsSingle);
```

Description

This property determines if the affected controls have a single-line or no visible border.

Caption property

Applies to

[TLabel](#), [TCheckbox](#), [TRadioGroup](#), [TGroupbox](#), [TPanel](#), [TButton](#)

Declaration

```
property Caption: String;
```

Description

This property contains the text string that labels the control.

CharCase property

Applies to

[TEdit](#)

Declaration

property CharCase: (ecNormal, ecUpperCase, ecLowerCase);

Description

The CharCase property determines the case of the Text property of the [TEdit](#) control.

Checked property

Applies to

[TCheckbox](#)

Declaration

property Checked: (True, False);

Description

Determines if the TCheckbox control is "set".

Color property

Applies to

[TLabel](#), [TEdit](#), [TCheckbox](#), [TRadioGroup](#), [TListbox](#), [TCombobox](#), [TGroupbox](#), [TPanel](#), [TButton](#)

Declaration

property Color: {Any valid screen color from dropdown list.}

Description

Determines the color or background (where applicable) color of a RID control.

Columns property

Applies to

[TRadioGroup](#), [TListbox](#)

Declaration

property Columns: Integer;

Description

The Columns property denotes the number of columns in the [TListbox](#) or [TRadioGroup](#) controls.

Ctl3D property

Applies to

[TEdit](#), [TCheckbox](#), [TRadioGroup](#), [TListbox](#), [TGroupbox](#), [TPanel](#), [TButton](#)

Declaration

```
property Ctl3D: (True, False);
```

Description

The Ctl3D property determines whether a control has a three-dimensional (3-D) or two-dimensional look. If Ctl3D is True, the control has a 3-D appearance. If Ctl3D is False, the control appears normal or flat. The default value of Ctl3D is True.

DropDownCount property

Applies to

[TCombobox](#)

Declaration

```
property DropDownCount: Integer;
```

Description

The DropDownCount property determines the number of lines in the drop-down list of a [TCombobox](#).

EditMask property

Applies to

[TEdit](#)

Declaration property

Description

The EditMask property is the mask that is used to limit the valid characters and data that can be put into a [TEdit](#) control.

Enabled property

Applies to

[TLabel](#), [TEdit](#), [TCheckbox](#), [TRadioGroup](#), [TListbox](#), [TCombobox](#), [TGroupbox](#), [TPanel](#), [TButton](#)

Declaration

```
property Enabled: (True, False);
```

Description

The Enabled property determines if the control can be modified when the Dialog Box is displayed.

ExtendedSelect property

Applies to

[TListBox](#)

Declaration

```
property ExtendedSelect: (True, False);
```

Description

The ExtendedSelect property determines if the user can select an range of items in the list box. ExtendedSelect works in conjunction with the [MultiSelect](#) property. If MultiSelect is False, the setting of ExtendedSelect has no effect as the user will not be able to select more than one item at a time in the list box.

If MultiSelect is True and ExtendedSelect is True, the user can select an item then hold down the Shift key and select another and all the items in between the two selected items also become selected. If the user doesn't hold down the Shift or Ctrl key while selecting a second item, the first selected item becomes unselected--in other words, the user must use the Ctrl key to select multiple noncontiguous items, or the Shift key to select a range of items. If ExtendedSelect is False, the user can select multiple items without using the Shift or Ctrl key, but they can't select a range of items in one operation.

FocusControl property

Applies to

[TLabel](#)

Declaration

property FocusControl: {Any valid control from the Controls list.}

Description

The FocusControl links the [TLabel](#) control with another control on the Dialog Box. If the Caption of a TLabel includes an accelerator key, the control specified as the value of the FocusControl property becomes the focused control when the user uses the accelerator key.

Font property

Applies to

[TLabel](#), [TEdit](#), [TCheckbox](#), [TRadioGroup](#), [TListbox](#), [TCombobox](#), [TGroupbox](#), [TPanel](#), [TButton](#)

Declaration

```
property Font: {Font selection Dialog}.
```

Description

The Font property is a font object that controls the attributes of text written on or in the control.

Height property

Applies to

[TLabel](#), [TEdit](#), [TCheckbox](#), [TRadioGroup](#), [TListbox](#), [TCombobox](#), [TGroupbox](#), [TPanel](#), [TButton](#)

Declaration

```
property Height: Integer;
```

Description

Height property of a control is the vertical size of the control, in pixels.

Hint property

Applies to

[TLabel](#), [TEdit](#), [TCheckbox](#), [TRadioGroup](#), [TListbox](#), [TCombobox](#), [TGroupbox](#), [TPanel](#), [TButton](#)

Declaration

```
property Hint: String;
```

Description

The Hint property is the text string that can appear when the mouse pointer is moved over a control.

IntegralHeight property

Applies to

[TListBox](#)

Declaration

```
property IntegralHeight: Integer;
```

Description

The IntegralHeight property controls the way the list box represents itself on the form. If IntegralHeight is True, the list box shows only entries that fit completely in the vertical space, and the bottom of the list box moves up to the bottom of the last completely drawn item in the list. If IntegralHeight is False, the bottom of the list box is at the location determined by its ItemHeight property, and the bottom item visible in the list might not be complete.

ItemHeight property

Applies to

[TListbox](#), [TCombobox](#)

Declaration

```
property ItemHeight: Integer;
```

Description

For list boxes, the ItemHeight property is the height (in pixels) of an item in the [TListbox](#) or [TCombobox](#).

ItemIndex property

Applies to

[TRadioGroup](#), [TListbox](#)

Declaration

```
property ItemIndex: Integer;
```

Description

Determines which element will be selected when the Dialog Box is displayed.

Items property

Applies to

[TLabel](#), [TEdit](#), [TCheckbox](#), [TRadioGroup](#), [TListbox](#), [TCombobox](#), [TGroupbox](#), [TPanel](#), [TButton](#), [TReport](#), [TWhere](#), [TParam](#)

Declaration property

Description

Kind property

Applies to

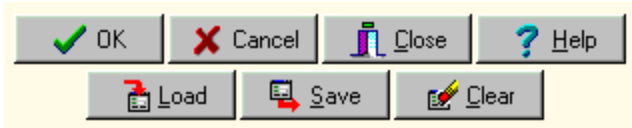
[TButton](#)

Declaration

property Kind: (bkOk, bkCancel, bkClose, bkHelp, bkLoad, bkSave, bkClear);

Description

This property determines which image is displayed as well as which action the Dialog will process.



Layout property

Applies to

[TButton](#)

Declaration

property Layout: (blGlyphLeft, blGlyphRight, blGlyphTop, blGlyphBottom);

Description

This property determines the Caption position in relation to the Glyph image for a [TButton](#) control.

Left property

Applies to

[TLabel](#), [TEdit](#), [TCheckbox](#), [TRadioGroup](#), [TListbox](#), [TCombobox](#), [TGroupbox](#), [TPanel](#), [TButton](#), [TReport](#), [TWhere](#), [TParam](#)

Declaration

```
property Left: Integer;
```

Description

The Left property determines the horizontal distance (in pixels) for a control from the left edge of the Dialog Box.

MaxLength property

Applies to

[TEdit](#), [TCombobox](#)

Declaration

```
property MaxLength: Integer;
```

Description

This property determines the maximum number of characters the user can enter in the Text attribute of the [TEdit](#) or [TCombobox](#) controls.

MultiSelect property

Applies to

[TListBox](#)

Declaration

```
property MultiSelect: (True, False);
```

Description

The MultiSelect property determines whether the user can select more than one element at a time from the list.

Name property

Applies to

[TLabel](#), [TEdit](#), [TCheckbox](#), [TRadioGroup](#), [TListbox](#), [TCombobox](#), [TGroupbox](#), [TPanel](#), [TButton](#)

Declaration

```
property Name: String;
```

Description

This is the name of the control. The name must be unique within the Dialog Box.

Password property

Applies to

[TEdit](#)

Declaration

```
property Password; (True, False);
```

Description

Determines whether an asterick (*) will be displayed instead of Text in a [TEdit](#) control.

ReadOnly property

Applies to

[TEdit](#)

Declaration

```
property ReadOnly: (True, False);
```

Description

The ReadOnly property determines if the user can change the contents of a [TEdit](#) control.

Sorted property

Applies to

[TListBox](#), [TComboBox](#)

Declaration

```
property Sorted: (True, False);
```

Description

The Sorted property determines whether a list will be sorted in alphabetical order.

Style property

Applies to

[TComboBox](#)

Declaration

property Style: (csDropDown, csDropDownList);

Description

This property determines if the user can enter text into a control, or instead must pick a selection from the dropdown list.

TabOrder property

Applies to

[TEdit](#), [TCheckbox](#), [TRadioGroup](#), [TListbox](#), [TCombobox](#), [TGroupbox](#), [TPanel](#), [TButton](#)

Declaration

```
property TabOrder: Integer;
```

Description

The TabOrder property determines the order in which controls receive the focus when the user presses the Tab key.

TabStop property

Applies to

[TEdit](#), [TCheckbox](#), [TRadioGroup](#), [TListbox](#), [TCombobox](#), [TButton](#)

Declaration

```
property TabStop: (True, False);
```

Description

The TabStop property determines if the user can tab to a control.

Text property

Applies to

[TEdit](#), [TCombobox](#)

Declaration

```
property Text: String;
```

Description

The Text property determines the text that appears within an [TEdit](#) or [TCombobox](#) control.

Top property

Applies to

[TLabel](#), [TEdit](#), [TCheckbox](#), [TRadioGroup](#), [TListbox](#), [TCombobox](#), [TGroupbox](#), [TPanel](#), [TButton](#), [TReport](#), [TWhere](#), [TParam](#)

Declaration

```
property Top: Integer;
```

Description

The Left property determines the horizontal distance (in pixels) for a control from the top edge of the Dialog Box.

Values property

Applies to

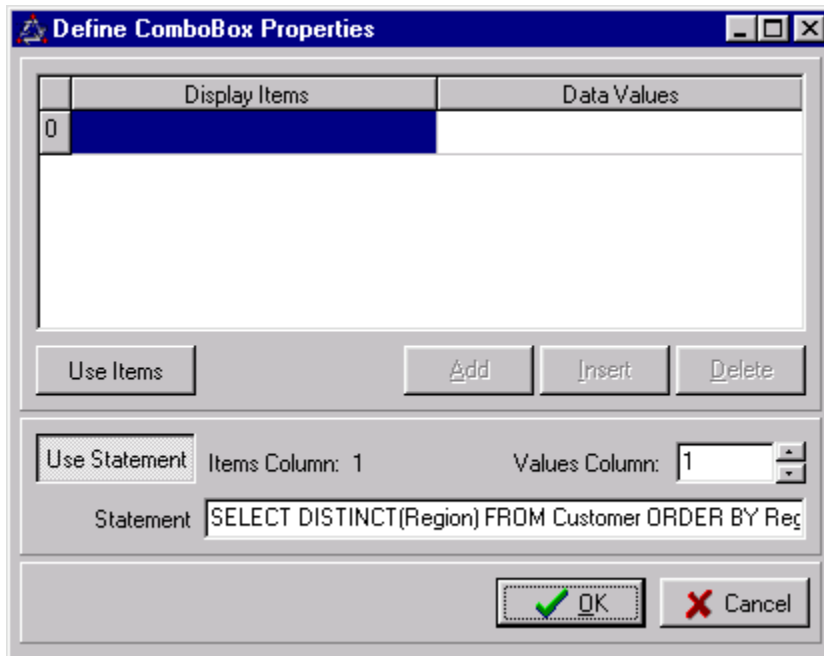
[TRadioGroup](#), [TListbox](#), [TCombobox](#)

Declaration

`property Values; {Dialog Property}`

Description

This property has the ability to present different Display Items and Data Values. For the [TListbox](#) and [TCombobox](#), these values can be [sourced](#) at run-time. Click on the image below for more information.



NOTE: The Values edit dialog for the TRadioGroup control only displays the "Use Items" portion of the above screen.

Visible property

Applies to

[TLabel](#), [TEdit](#), [TCheckbox](#), [TRadioGroup](#), [TListbox](#), [TCombobox](#), [TGroupbox](#), [TPanel](#), [TButton](#)

Declaration

property

Description

The Visible property determines whether the control appears onscreen when the Dialog Box is displayed.

Width property

Applies to

[TLabel](#), [TEdit](#), [TCheckbox](#), [TRadioGroup](#), [TListbox](#), [TCombobox](#), [TGroupbox](#), [TPanel](#), [TButton](#)

Declaration

```
property Width: Integer;
```

Description

The Width property determines the horizontal size of a control.

WordWrap property

Applies to

[TLabel](#)

Declaration

```
property WordWrap: (True, False);
```

Description

The WordWrap property determines the text in the Caption property of a [TLabel](#) control wraps at the right margin.

Default property

Applies to

[TButton](#)

Declaration

```
property Default: (True, False);
```

Description

The Default property indicates whether a [TButton](#) control is the default button, which will be activated when the user presses Enter.

Destination property

Applies to

[TReport](#)

Declaration

property Destination: (File, Printer, Window);

Description

Determines the default output destination when a report is executed.

NOT YET IMPLEMENTED!

ReportName property

Applies to

[TReport](#)

Declaration

```
property ReportName: String;
```

Description

This is the name of the Crystal Reports RPT file associated with the [TReport](#) control.

ReportType property

Applies to

[TReport](#)

Declaration

property ReportType: (Where Clause, Stored Procedure);

Description

This read-only value indicates on what type of data retrieval mechanism the Crystal Report is based.

SQL property

Applies to

[TReport](#)

Declaration

```
property SQL: {SQL Dialog};
```

Description

This is the [SQL Statement](#) used by Crystal Reports to retrieve the data used in generating the report. When [TWhere](#) controls are added to the Dialog Box, they will modify the *Where Clause* component of this statement. For particularly complex reporting or dialog needs, you may manually modify this value. See Tutorial #4 for an example of this technique.

This property is Read-Only for when the [ReportType](#) property is *Stored Procedure*.

LoginID property

Applies to

[TReport](#)

Declaration

```
property LoginID: String;
```

Description

This property value is the LoginID for the DataSource associated with the Crystal Report. It can be changed at both design and run-time.

Password property

Applies to

[TReport](#)

Declaration

property Password: String;

Description

This property value is the Password for the LoginID associated with the Crystal Report. It can be changed at both design and run-time.

DataSource property

Applies to

[TReport](#)

Declaration

property DataSource: {Any valid DataSource from the List.}

Description

This property represents the [ODBC](#) datasource the will be used to populate any [SQL Controls](#) are run-time.

The DataSource must be configured on each PC that will execute the RID Dialog. If it is not available, an ODBC error will occur.

Database property

Applies to

[TReport](#)

Declaration

```
property Database: String;
```

Description

This property value is the Database for the Datasource associated with the Crystal Report. It can be changed at both design and run-time. No validation of this value is performed until the Dialog Box is tested.

If you changed the Database at run-time and it does not exist for the specified Datasource, the an error will be displayed.

AutoLaunch property

Applies to

[TReport](#)

Declaration

```
property AutoLaunch: (True, False);
```

Description

This property determines if the Dialog Box will be displayed when the Execute() method is called. It is most often useful when you wish to execute a Crystal Report that doesn't have any user-determined criteria.

FieldName property

Applies to

[TWhere](#)

Declaration

```
property FieldName: String;
```

Description

This property corresponds to the name of the table attribute to be used in the modification of the "Where Clause" of the [SQL Statement](#). It must be set to reflect an actual table attribute name, or an error will be generated when Crystal Reports attempts to generate the report.

DataType property

Applies to

[TWhere](#), [TParam](#)

Declaration

```
property DataType: (Boolean, Character, DateTime, Numeric);
```

Description

This property value determines how RID will format the user-entered criteria for use in the modified [SQL Statement](#). This property must be accurate or an error will be generated when Crystal Reports attempts to generate the report.

For [TParam](#) controls, this is a read-only property.

DialogControl property

Applies to

[TWhere](#), [TParam](#)

Declaration

property DialogControl: {Any valid control from the Controls list.}

Description

This is the key property of these controls, and provides the link between the other data entry controls on the Dialog Box and the modified [SQL Statement](#) or [Stored Procedure](#). This property must be set accordingly or an error will be generated when Crystal Reports attempts to generate the report.

Value property

Applies to

[TWhere](#), [TParam](#)

Declaration

```
property Value: String;
```

Description

By setting this property, a [TWhere](#) or a [TParam](#) control does not need to be associated with a user-entered data control. Use this when you wish to pass default values to a [SQL Statement](#) or [Stored Procedure](#).

Comparator property

Applies to

[TWhere](#)

Declaration

property Comparator: (Between, Equals, GT, GTE, In, LT, LTE, Like, NE);

Description

This property determines the comparison methodology to be used in the modified [SQL Statement](#). The following describes in detail how each Comparator is used:

Between	The database value must fall between the two components of the user-entered value. Data Controls associated with this Comparator must format data in a "Value AND Value" manner for correct functionality. For this reason, it is recommended to use a TRadioGroup control when using this comparator, since you can format the Data Value as you wish.
Equals	Database values must equal the user-entered value.
Greater Than	Database values must be greater than the user-entered value.
Greater Than/Equals	Database values must be greater than or equal to the user-entered value.
In	The database value must be in the list of values selected by the user. This Comparator will mostly be used with a TListbox control that has its MultiSelect property set to true. NOTE: If the DataType for this attribute is Character and you are using a TListbox control, you will need to set the WrapValue property for the TListbox control to TRUE and set the WrapValue property for this TWhere control to FALSE. This indicates to RID to format the selected TListbox values in the following manner: ("Value1", "Value2", "Valuen").
Less Than	Database values must be less than the user-entered value.
Less Than/Equals	Database values must be less than or equal to the user-entered value.
Like	The database values must be equal to the user-entered value, but the user may enter database-specific wildcard characters to initiate a wildcard search.
Not Equals	Database values must be not equal to the user-entered value.

FieldName property

Applies to

[TWhere](#)

Declaration

```
property FieldName: String;
```

Description

UNUSED.

ParamType property

Applies to

[TParam](#)

Declaration

```
property ParamType;
```

Description

INTERNAL USE ONLY.

ParamNum property

TParam

Declaration

property ParamNum: Integer;

Description

INTERNAL USE ONLY - Indicates the position in the Stored Procedure parameter list for the TParam control.

WrapValue property

Applies to

[TListBox](#), [TWhere](#), [TParam](#)

Declaration

```
property WrapValue: (True, False);
```

Description

Set this property to True to cause RID to wrap the user-entered value with double-quotation mark characters. This property is *linked* to the [DataType](#) property, and by default, the only time it will be False is when the DataType is numeric.

You may need to set this property to False when the following conditions apply:

- The TWhere control [DataType](#) is Character.
- The TWhere control [DialogControl](#) links to a [TListBox](#).
- The TWhere control [Comparator](#) is **In**.
- The TListBox [MultiSelect](#) property is True.

NOTE: If this is the case, then don't forget to set the WrapValue property for the TListbox control to True.

This property is Read-Only for [TParam](#) controls.

BlankIsNull property

Applies to

[TWhere](#), [TParam](#)

Declaration

```
property BlankIsNull: (True, False);
```

Description

Set this property to True to cause RID to pass a NULL value where the data control value is blank at run-time.



DialogBox Control

Properties

Description

Consider this your dialog design canvas. Select and drop other controls onto this form to design the look & feel that you desire.

The value in the Caption property appears as the Title of the displayed Dialog Box.



Selector Tool

Description

The Selector Tool can be pressed at any time to de-select the active control and reset the Dialog Box to edit mode.



TLabel Control

[See Also](#) [Properties](#)

Description

The TLabel control displays text on the Dialog Box. Usually this text labels some other control.

The text of a label is the value of its Caption property. Within the caption, you can include an accelerator key. When the user presses the accelerator key, the control that is the value of the label's [FocusControl](#) will be selected as the active control.

How the text of the caption aligns within the label is determined by the value of the [Alignment](#) property. You can have the label resize automatically to fit a changing caption if you set the [AutoSize](#) property to True. If you prefer to have the text wrap, set [WordWrap](#) to True.



TEdit Control

[See Also](#)

[Properties](#)

Description

The TEdit control is a standard Windows edit control that can be associated with an [EditMask](#) so that the user is forced to input according to a defined format.

Set the [Password](#) property to True to display a series of astericks instead of the characters entered by the user.



TCheckbox Control

[See Also](#) [Properties](#)

Description

Use the TCheckbox control when you need to present a Yes/No, True/False type input to the user. The [Checked](#) property determines the initial state of the control when the Dialog Box is displayed.

The [Caption](#) property sets the text associated with the TCheckbox control to identify its purpose.



TRadioGroup Control

[See Also](#) [Properties](#)

Description

The TRadioGroup control is used when you wish to present a collection of mutually-exclusive choices to the user.

This control can Display a different value to the user than the value which is sent to Crystal Reports for report generation, e.g. the user sees "New York" but "NY" is sent to Crystal Reports.

The [Values](#) property permits modification of both Display Items and Data Values. The [Columns](#) property determines how many columns of choices the control will display. You can set the [ItemIndex](#) property to present the user with a pre-selected option when the DialogBox is displayed.



TListbox Control

[See Also](#) [Properties](#)

Description

The TListbox control is a standard Windows Listbox control offering the user the ability to select one or more elements from a scrolling list of choices. The [MultiSelect](#) property determines if the user can select more than one element from the list. The [ExtendedSelect](#) property determines how the user can select multiple elements.

This control can Display a different value to the user than the value which is sent to Crystal Reports for report generation, e.g. the user sees "New York" but "NY" is sent to Crystal Reports.

The TListbox is one of two [SQL Controls](#) offered by RID. The [Values](#) property determines if the list is based on a finite set of choices or is [sourced from the database](#) at run-time.



TComboBox Control

[See Also](#) [Properties](#)

Description

The TComboBox control is a combination of an edit control for entering text and a drop down list for selecting from predefined values. The [Style](#) property determines if the user can enter data or is forced to select an element from the drop down list.

This control can Display a different value to the user than the value which is sent to Crystal Reports for report generation, e.g. the user sees "New York" but "NY" is sent to Crystal Reports.

The TComboBox is one of two [SQL Controls](#) offered by RID. The [Values](#) property determines if the list is based on a finite set of choices or is [sourced from the database](#) at run-time.



TGroupbox Control

[See Also](#) [Properties](#)

Description

The TGroupbox control is a standard Windows group box. Use a TGroupbox control to group related controls on the Dialog Box.



TPanel Control

[See Also](#) [Properties](#)

Description

The TPanel control is placed on the Dialog Box as a container for other controls. When the TPanel is moved, all of the contained controls move with it.

TPanels can be aligned with the form so that they maintain the same relative position to the Dialog Box even when the Dialog Box is resized. Align a panel with the [Align](#) property.

To customize the appearance of a TPanel, modify the [BevelInner](#), [BevelOuter](#), and [BevelWidth](#) properties.

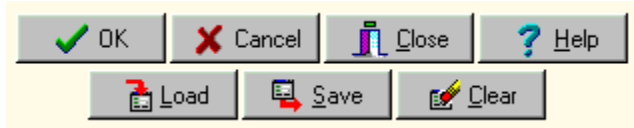


TButton Control

[See Also](#) [Properties](#)

Description

This control is used to initiate Actions on the Dialog Box. The [Kind](#) property determines which image is displayed as well as which action the Dialog Box will process.



In your application, you can provide an [event handler](#) to process non-default behavior.



TReport Control

See Also [Properties](#)

Description

This control provides the critical link between Report Interface Designer and Crystal Reports. At a minimum, you need to set the [ReportName](#) property. This will automatically set various other property values.

When the AutoLaunch property is True, then the Dialog Box is not displayed and the Crystal Report is immediately called. If there are any [TWhere](#) or [TParam](#) controls, then their default [Value](#) will be used.



TWhere Control

[See Also](#) [Properties](#)

Description

This control represents part of the [SQL Statement](#) associated with the [TReport](#) control.

You will need to assign a [DialogControl](#) to link to this TWhere control, or a default [Value](#) if one is known.



TParam Control

See Also [Properties](#)

Description

This control represents a [Stored Procedure](#) parameter, and will be automatically created when a Stored Procedure-based report is associated to the [TReport](#) control.

You will need to assign a [DialogControl](#) to link to this TParam control, or a default [Value](#) if one is known.



Report Interface Designer Controls

Designer Controls

Report Interface Designer supports the most common application controls. Most controls are visual components, in that you can see and interact with them when the RID Dialog is visible. Most controls have common properties that specify the visual attributes of controls, such as [Left](#), [Top](#), [Height](#), [Width](#), and [Hint](#).



[TLabel](#) Use this control to display text on a form, such as text labels for other controls.



[TEdit](#) Like a common Windows edit control except that you can restrict what characters are entered.



[TCheckbox](#) Use this control to give the user a Yes/No type selection option.



[TRadioGroup](#) This is a groupbox that presents multiple Radio Buttons.



[TListbox](#) This controls is a Windows list box from which users can select one or more items.



[TCombobox](#) This control combines an edit box with a listbox so users can either type text in the edit box or select an item from the list.



[TGroupbox](#) Use a group box control to group other controls on the RID Dialog.



[TPanel](#) Use a panel as a container for other controls, with various cosmetic attributes.



[TButton](#) Use this control to permit the user to initiate a RID Dialog action.



[TReport](#) This control provides the programmatic link between RID and Crystal Reports.



[TWhere](#) This control represents a part of the "Where" clause of a [SQL selection statement](#).



[TParam](#) This control represents an input parameter for reports based on [Stored Procedures](#).

Alignment property

[See also](#) _____

Applies to

[TCheckbox](#)

Declaration

property Alignment: (taLeftJustify, taRightJustify);

Description

This property determines which side of the Checkbox the Caption will be situated.

See also

[Alignment](#) for other controls

ParamName property

Applies to

[TParam](#)

Declaration

```
property ParamName: String;
```

Description

This property is the name of the [Stored Procedure](#) parameter associated with this [TParam](#) control.

This property is Read-Only.

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

[TWhere](#) control

[TParam](#) control

[Working with Crystal Reports](#)

[Sourcing Control data](#)

See also


[TReport](#) control
[TParam](#) control

See also

[TReport](#) control

[TWhere](#) control

Properties

 Key properties

Align

 Alignment

 AutoSize

Caption

Color

Enabled

 FocusControl

Font

Height

Hint

Left


Name

Top

Width

 WordWrap

Properties

 Key properties



[AutoSelect](#)



[AutoSize](#)



[BorderStyle](#)



[CharCase](#)

[Color](#)

[Ctl3D](#)



[EditMask](#)

[Enabled](#)

[Font](#)

[Height](#)

[Hint](#)

[Left](#)



[MaxLength](#)

[Name](#)



[Password](#)



[ReadOnly](#)

[TabOrder](#)

[TabStop](#)



[Text](#)

[Top](#)

[Visible](#)

[Width](#)

Properties



Key properties

[Alignment](#)

[Caption](#)



[Checked](#)

[Color](#)

[Ctl3D](#)

[Enabled](#)

[Font](#)

[Height](#)

[Hint](#)

[Left](#)

[Name](#)

[TabOrder](#)

[TabStop](#)

[Top](#)

[Visible](#)

[Width](#)

Properties



Key properties

[Align](#)
[Caption](#)
[Color](#)



Columns

[Ctl3D](#)
[Enabled](#)
[Font](#)
[Height](#)
[Hint](#)



[ItemIndex](#)
[Left](#)
[Name](#)
[TabOrder](#)
[TabStop](#)
[Top](#)



[Values](#)

[Visible](#)
[Width](#)

Properties



Key properties

Align



BorderStyle

Color



Columns

Ctl3D

Enabled



ExtendedSelect

Font

Height

Hint



IntegralHeight



ItemHeight



ItemIndex

Left



MultiSelect

Name



Sorted

TabOrder

TabStop

Top



Values

Visible

Width



WrapValue

Properties



Key properties

Color



DropDownCount

Enabled

Font

Height

Hint



ItemHeight

Left



MaxLength

Name



Sorted



Style

TabOrder

TabStop



Text

Top



Values

Visible

Width

Properties



Key properties

[Align](#)

[Caption](#)

[Color](#)

[Ctl3D](#)

[Enabled](#)

[Font](#)

[Height](#)

[Hint](#)

[Left](#)

[Name](#)

[TabOrder](#)

[Top](#)

[Visible](#)

[Width](#)

Properties



Key properties

Align



Alignment



BevelInner



BevelOuter



BevelWidth



BorderStyle

Caption

Color

Ctl3D

Enabled

Font

Height

Hint

Left

Name

TabOrder

Top

Visible

Width

Properties



Key properties

Caption



Default

Enabled

Font

Height

Hint



Kind



Layout

Left

Name

TabOrder

TabStop

Top

Visible

Width

Properties



Key properties



AutoLaunch



DataSource



Database



Destination
Left
Name



ReportName



ReportType



SQL



LoginID



Password
Top

Properties



Key properties



BlankIsNull



Comparator



Datatype



DialogControl



FieldName

Left

Top



Value



WrapValue

Properties



Key properties



BlankIsNull



Comparator



Datatype



DialogControl
Left



ParamName

Top



Value



WrapValue

Properties



Key properties

[Caption](#)

[Color](#)

[Ct13D](#)

[Height](#)

[Left](#)

[Name](#)

[Top](#)

[Width](#)

Ok Button

Setting the Kind property to bkOk will cause this image to be displayed. The RID Dialog default response for this action is to launch the associated Crystal Report.

Cancel Button

Setting the Kind property to bkCancel will cause this image to be displayed. The RID Dialog default response for this action is to close the dialog.

Close Button

Setting the Kind property to bkClose will cause this image to be displayed. The RID Dialog default response for this action is to close the dialog.

Help Button

Setting the Kind property to bkHelp will cause this image to be displayed. There is no default RID Dialog response for this action. You may add functionality for this action in the assigned OnAction event handler.

Load Button

Setting the Kind property to bkLoad will cause this image to be displayed. There is no default RID Dialog response for this action. You may add functionality for this action in the assigned OnAction event handler. Typically, use this button to initiate a process of loading pre-determined criteria into the Dialog Box.

Save Button

Setting the Kind property to bkSave will cause this image to be displayed. There is no default RID Dialog response for this action. You may add functionality for this action in the assigned OnAction event handler. Typically, use this button to initiate a process of saving the user-entered criteria from the Dialog Box.

Clear Button

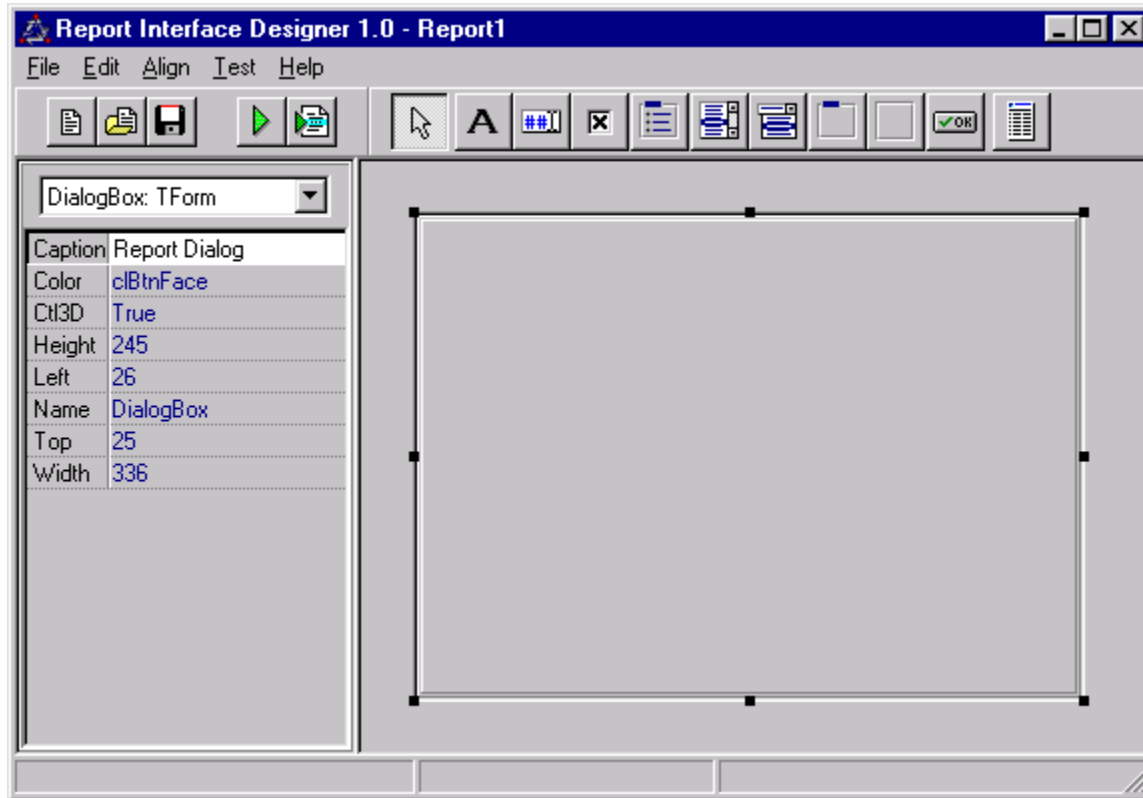
Setting the Kind property to `bkClear` will cause this image to be displayed. There is no default RID Dialog response for this action. You may add functionality for this action in the assigned `OnAction` event handler. Typically, use this button to reset all controls in the Dialog Box to empty or default values.



The RID Screen

The RID Development Environment

The Report Interface Designer development environment is very similar to other popular programming tools. Click on the image below for further details.



Dialog Box

Consider this to be your *Design Canvas*. Add controls by selecting them from the Controls Palette and then dropping into this area. Certain visual aspects of the DialogBox can be controlled, such as size, color, caption, etc.

Property Inspector

The Property Inspector permits you to set properties values for the controls in the Dialog Box.

Controls Palette

Most of the controls listed on the palette are the standard Windows control elements which you can add to your dialog.

Object Selector

Use this combobox to quickly locate and select controls already present in the Dialog Box.

New Button

Corresponds to File|New. Press this button to clear the Dialog Box and create a new Dialog Definition File.

Open Button

Corresponds to File|Open. Press this button to open an existing Dialog Definition File.

Save Button

Corresponds to File|Save. Press this button to save your work to a Dialog Definition File.

Test Dialog Button

Corresponds to Test|Dialog Only. Pressing this button will display the current Dialog Box.

Test Dialog With Report Button

Corresponds to Test|Dialog with Report. Pressing this button will display the current Dialog Box, and if an "Ok" button is pressed then the associated Crystal Reports will be launched.



The Property Inspector

Whenever a control is selected in the Dialog Box, the control properties are available for editing using the Property Inspector. Control Properties are generally of three types:

- Edit** where you enter text representing the property value.
- Combobox** where you select a property value from a drop-down list of options.
- Dialog** where pressing the "..." button displays a dialog box for modifying property values.

Use the mouse to select the property you wish to change and enter the desired value. Most property value changes will be immediately reflected by the control in the Dialog Box.




The Controls Palette

[See also](#)

Designer Controls



The Selector Tool  is the default control. When the application starts or after another control has been placed, the Selector Tool will be highlighted. When a control is highlighted, you can not select any current controls from the Dialog Box, though you could still choose an existing control by activating the [Object Selector](#) and picking the desired control from the list.

Using the mouse, select the image of the control you wish to work with and then move the mouse pointer to a location on the Dialog Box where you wish the control to be placed. Press the mouse button and the control will appear on the Dialog Box.

The Selector Tool can be pressed at any time to de-select the active control and reset the Dialog Box to edit mode.

Controls Palette - See also

[RID Controls](#)



File Menu

The File menu permits creating, loading, and saving of the Dialog Definition Files (DDF).

New	Clear the Dialog Box and create a new DDF
Open	Open an existing DDF
Close	Close the current DDF and clear all controls from the Dialog Box
Save	Save the current DDF
Save As	Save the current DDF under a new name.



Edit Menu

The Edit menu implements the standard Windows Cut, Copy, & Paste functionality.



Align Menu

The Align menu allows you to select multiple controls and align them in relation to each other.

- Align Right** Align the right sides of selected controls with the Right-most control.
- Align Left** Align the left sides of selected controls with the Left-most control.
- Align Center** Align the selected controls in the horizontal center of the Dialog Box.

- Align Top** Align the top sides of selected controls with the Top-most control.
- Align Bottom** Align the bottom sides of selected controls with the Bottom-most control.
- Align Center** Align the selected controls in the vertical center of the Dialog Box.



Test Menu

The Test menu permits testing of the current set of controls in the Dialog Box.

- Dialog Only** This will display the Dialog Box. If there are any [SQL Controls](#) (TListbox, TCombobox) on the dialog, then they will be sourced with live data.
- Dialog with Report** This will display the Dialog Box, and also execute Crystal Reports if an "Ok" button is pressed.



Help Menu

The Help menu provides access to Help as well as other information.

Contents	Displays the Help Contents.
Search	Displays the Help Index.
Contacting TRI-I	Displays the TRI-I Engineering Contact Info dialog.
Register	If you are using the Evaluation version, this will permit entry of Registration details.
About	Displays the RID About Box.



Creating Interface Dialogs

[See also](#)

Creating RID Dialogs is similar to the dialog creation methodology employed by many visual development tools. You begin with a form [canvas](#), and add controls to this form which correspond to the functionality you wish to produce.

[RID controls](#) can be divided into 4 distinct group: cosmetic or informational controls; data entry controls; action controls; and reports controls.

Informational Controls -



These controls are essentially non-interactive, but either describe, group, or contain other controls in a manner they conveys information to the user.

Data Entry Controls -

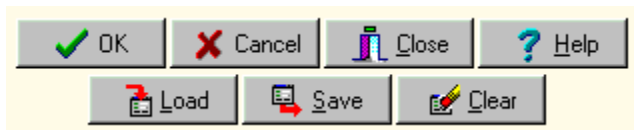


Use a combination of these controls to accept criteria from the user in either a free-form or controlled manner.

Action Controls -



Though only a single control, the TButton [Kind](#) property determines which image is displayed as well as which action the Dialog will process.



In your application, you provide an [event handler](#) to process the specific Action associated with the TButton.

Report Controls -





The [TReport](#) control provides the link between Report Interface Designer and Crystal Reports. When you select an RPT file using the [ReportName](#) property, RID determines certain aspects of the associated report: which server, database, and user, as well as the [SQL Statement](#) basis for retrieving data to be processed by the report.

Crystal Reports supported by RID are based on a [SQL Statement](#) or a server [Stored Procedure](#). When the report is first assigned to the TReport control, the [SQL](#) property will be set with this attribute value, and the [ReportType](#) property will be updated with the report type.

If the ReportType is *Stored Procedure* and the Stored Procedure has input parameters, then a [TParam](#) control will be automatically created for each of these parameters. The TParam control provides the link between a Data Entry control on the Dialog Box and the actual parameter value passed to the Stored Procedure at run-time.

If the ReportType is *Where Clause*, then the [TWhere](#) control will become visible on the [Controls Palette](#) and can be added to the [Dialog Box](#). The TWhere controls are used to modify the SQL Statement associated with the TReport control, and permit establishing additional selection criteria. The TWhere control provides the link between a Data Entry control on the Dialog Box and the report SQL Statement at run-time.

Creating Interface Dialogs - See also

Informational Controls Action Control



[TLabel](#)



[TButton](#)



[TGroupbox](#)



[TPanel](#)

Data Controls Report Controls



[TEdit](#)



[TReport](#)



[TCheckbox](#)



[TWhere](#)



[TRadioGroup](#)



[TParam](#)



[TListbox](#)



[TCombobox](#)

Event Handler

An event handler is simply a method called by the RIDComp component or RIDCompX control in response to the user clicking on a TButton in the Dialog Box. You assign the event handler in your development environment and then add the code specific to how you wish your RID Dialog to respond.

SQL Statement

This is an ANSI/SQL statement following a specific syntax. At a minimum there will be the SELECT keyword followed by a series of table attributes then a FROM keyword and a table name. Additional syntactical structures can include a WHERE clause, an ORDER BY clause, a GROUP BY clause, etc.

Stored Procedure

A Stored Procedure is a series of ANSI/SQL statements which are executed on the Server when requested. Only certain database servers support Stored Procedures, among these are Oracle, Sybase SQL Server, Microsoft SQL Server, Sybase SQL Anywhere, Borland Interbase, etc.



Working with Crystal Reports

In order for the correct operation of Report Interface Designer, please insure that Seagate Crystal Reports™ "Professional" Version 5.0 (or greater) for Windows 95/NT has been installed and is operating correctly. In addition, the RID Tutorial uses some of the sample reports and databases which are installed during program setup.

While Crystal Reports can directly access many of the most common PC database formats, i.e., dBASE, FoxPro, Clipper, Btrieve, Paradox, and Microsoft Access, Report Interface Designer requires the developer to use an SQL/ODBC datasource for creating reports.

Crystal Reports allows you to change the driver associated with a report at any time. For CRW 5.0, select File|Report Options and then modify the values in the Convert Report section. For CRW 6.0, select Database|Convert Database Driver and then modify the dialog values.

When a report is first created, CRW requires the user to choose a database and then *Login*. When RID reads the saved RPT file, the Server, Database, and LoginID values are retrieved and populate the appropriate TReport properties. If the server requires a Login Password, RID will display a dialog box permitting you to enter this value, which is then stored to the TReport Password property.

During the design phase, you will more than likely use the default Server, Database, LoginID, and Password values. At run-time, however, you can programatically change any of these values to modify how Crystal Reports retrieves its data. Be aware, however, that database objects must be comparable across different servers or databases for the functionality to work correctly.

TIP: If you plan on changing the "Data Source" of the Report Object from that which is saved with the report, you should set the "Verify on Every Print" option for the report in Crystal Reports. You may also need to remove the table "Database & Owner qualifiers" in the "Set Location..." dialog. These qualifiers are automatically added when the report is created, and may be different across servers or databases. A general rule-of-thumb about Crystal Reports and SQL databases is that as long as the table name(s) or Stored Procedure can be found in the assigned server and database, then the report will process correctly. Be advised, however, that this type of functionality may require a fair amount of testing, and if it does fail, the errors reported by both Crystal Reports and RID may not be totally deterministic as to the cause of the failure.



A Note about ODBC, CRW, and RID

[See also](#)

Crystal Reports can use standard ODBC connectivity or a Native database driver to retrieve data from the database. Report Interface Designer, however, relies on ODBC for populating the [TListbox](#) and [TCombobox](#) [SQL Controls](#). This may have some impact on your design decisions.

If you are planning on using the Native drivers in Crystal Reports and you do not use any of the [SQL Control](#) capabilities of the [TListbox](#) and [TCombobox](#) controls, then there is nothing more you need to do: everything will work seamlessly.

However, if a [TListbox](#) or [TCombobox](#) control is configured to retrieve data to populate the control, then you must do a couple of additional steps.

When the CRW RPT file is first assigned to a [TReport](#) control, the [Datasource](#) associated with the report is compared to configured ODBC datasources. If a match is found, then you will not notice any functional differences. At run-time, RID will connect to this datasource using the LoginID and Password property values of the [TReport](#) control to retrieve any required data. **However, this same ODBC datasource will then need to be configured on all your users computers.**

If the associated [Datasource](#) is not found in the configured ODBC datasources, the [Datasource](#) property value will be set to *<unknown>*. If you then configure a datasource with the correct name, you can assign the [Datasource](#) property to this datasource. Until a correct datasource is assigned, the [SQL Controls](#) will not be populated and an error will be displayed at run-time.

NOTE: When RID attempts to populate a [SQL Control](#) at run-time, the connection to the server is made at this moment and then RID disconnects from the server immediately after the data is retrieved. While this lessens the number of connections your database server needs to support, if your application is still connected when RID is activated, you will open an additional connection to the server. If this is a problem, the only solution is to either increase the number of simultaneous connections supported by your server, or disconnect your application temporarily while RID is activated.

See also

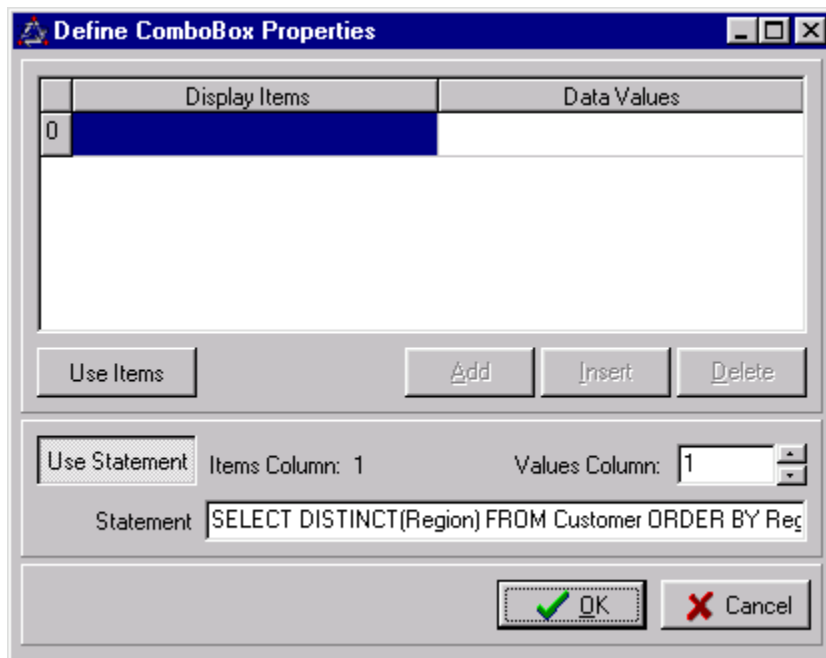
[Sourcing Control data at Run-Time](#)



Sourcing Control data at Run-Time

[See also](#)

Two of the Report Interface Designer controls, [TListbox](#) and [TCombobox](#), have the ability to populate their display values at run-time. To enable this functionality, modify the [Values](#) property of the control and set an [ODBC](#) SQL statement. In addition, the value displayed to the user can be different from the value which is forwarded to Crystal Reports for the actual report generation. Click on the image below for more information.



Sourcing Controls - See also

[TListbox](#) Control

[TCombobox](#) Control

SQL Controls

This refers to the TListbox and TCombobox controls when the values they display are based on a SQL Statement that is executed when the Dialog Box is displayed.

Display Items

Values in this column will be displayed in the control. This is what the user sees.

Data Items

Values in this column will be passed to Crystal Reports when the report is executed.

Use Items

Select this option when you have a finite list of Display and Data values.

Add Item

Press this button to insert a new element in the Display Items/Data Values grid at the end of the current element list.

Insert Item

Press this button to insert a new element in the Display Items/Data Values grid after the current row.

Delete Item

Press this button to delete the current element in the Display Items/Data Values grid.

Use Statement

Select this option when you want the Display and Data values to be based on a SQL Statement which will be populated at run-time.

Value

This values corresponds to the column number in the SQL Statement result set that you wish to use as the Data values for this control. The Display values will always be based on column 1 in the result set.

Statement

This is the SQL Statement which will be used at run-time to populate the Display and Data values for the control.

Bevel - See Also

[BevelInner](#) property

[BevelOuter](#) property

[BevelWidth](#) property



Contacting TRI-I Engineering

For product information or registration, please visit our website for the most up-to-date information. Our normal business hours are Monday - Friday, 9:00am to 6:00pm, PST. You may leave a voice message outside of these hours and we will contact you as soon as possible. You may also contact us by e-mail.

Address:

TRI-I Engineering, Inc.
200 W. 17th Street, Suite 80
Cheyenne, WY 82001

Telephone:

Sales & Information
(888) 551-3500

Support
(310) 967-3966

World Wide Web:

<http://www.iii-eng.com>

sales@iii-eng.com

support@iii-eng.com



Technical Support

We provide free Technical Support to registered users for 60 days from purchase or registration. To receive support, you will be asked to provide your product serial number. Technical Support for the Evaluation version is not available, however, we will attempt to answer via e-mail any support questions you may have during the evaluation period.

An ever growing [FAQ](#) (Frequently Asked Questions) is available on our web site, and we recommend reviewing this document prior to calling or contacting Technical Support.

Our normal business hours are Monday - Friday, 9:00am to 6:00pm, PST. The Technical Support telephone number is (310) 967-3966.

World Wide Web:

<http://www.iii-eng.com>

sales@iii-eng.com

support@iii-eng.com



License Agreement

Report Interface Designer (RID) is Copyright © 1997, by TRI-I Engineering, Inc., All Rights reserved.

This software and accompanying documentation are protected by United States copyright law and also by International Treaty provisions. Any use of this software in violation of copyright law or the terms of this agreement will be prosecuted to the fullest extent permissible by law.

TRI-I Engineering, Inc. authorizes you to make archival copies of this software for the sole purpose of back-up and protecting your investment from loss. Under no circumstances may you copy this software or documentation for the purposes of distribution. You are not to remove any of the copyright notices included in the software or product documentation.

You may distribute, without runtime fees or further licenses, your own compiled programs based on any of the compile units and/or components included with Report Interface Designer (RID). You may also distribute, without runtime fees or further licenses, the RIDEXEC.DLL dynamic link library, as well as any compiled sample programs contained with this package. You may not distribute any of the Report Interface Designer (RID) design-time components or the designer application without written permission from TRI-I Engineering, Inc.

The previous restrictions do not prohibit you from distributing your own source code, units, or components that depend upon Report Interface Designer (RID). However, others who receive your source code, units, or components need to purchase their own copies of Report Interface Designer (RID) in order to compile the source code or to write programs that use your units or components which are dependent on either the Report Interface Designer (RID) components or the RIDEXEC.DLL dynamic link library.

The supplied software may be used by one person on as many computer systems as that person uses. Group programming projects making use of this software must purchase a copy of the software and documentation for each member of the group.

TRI-I Engineering, Inc. warrants that the physical CD or diskettes and documentation provided with Report Interface Designer (RID) shall be free of defects in materials and workmanship for a period of 60 days from the date of receipt. If you notify us of such a defect within the warranty period, TRI-I Engineering, Inc. will replace the defective CD, diskette(s), or documentation at no cost to you.

TRI-I Engineering, Inc. warrants that the software will function as described in this documentation for a period of 60 days from receipt. If you encounter a bug or deficiency, we will require a problem report detailed enough to allow us to find and fix the problem. If you properly notify us of such a software problem within the warranty period, TRI-I Engineering, Inc. will update the defective software at no cost to you.

TRI-I Engineering, Inc. further warrants that the purchaser will remain fully satisfied with the product for a period of 60 days from receipt. If you are dissatisfied for any reason, and TRI-I Engineering, Inc. cannot correct the problem, contact the party from whom the software was purchased for a return authorization. If you purchased the product directly from TRI-I Engineering, Inc., we will refund the full purchase price of the software (not including shipping costs) upon receipt of the original program CD or diskette(s) and documentation in undamaged condition. TRI-I Engineering, Inc. cannot offer refunds directly to anyone who did not purchase a product directly from us.

Disclaimer of Warranty

TRI-I ENGINEERING, INC. DOES NOT ASSUME ANY LIABILITY FOR THE USE OF REPORT INTERFACE DESIGNER (RID) BEYOND THE ORIGINAL PURCHASE PRICE OF THE SOFTWARE. IN NO EVENT WILL TRI-I ENGINEERING, INC. BE LIABLE TO YOU FOR ADDITIONAL DAMAGES,

INCLUDING ANY LOST PROFITS, LOST SAVINGS, OR OTHER INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OF OR INABILITY TO USE THESE PROGRAMS, EVEN IF TRI-I ENGINEERING, INC. HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Governing Law

This agreement shall be governed by the laws of the State of Wyoming, United States of America.

All TRI-I Engineering, Inc. product names are trademarks or registered trademarks of TRI-I Engineering, Inc. Other brand and product names are trademarks or registered trademarks of their respective holders.

