

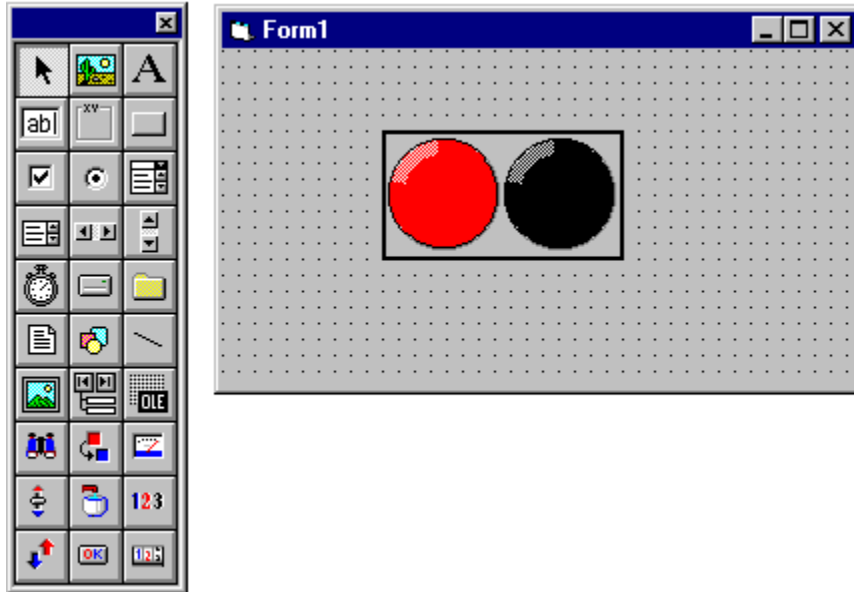
## Introduction

[Properties](#)

[Events](#)

[Methods](#)

The RSButton control is a Visual Basic OLE Custom Control developed by Rockwell Software. Button types available are momentary, toggle, rocker, checkbox, option, dual button, diode and picture buttons. Also features up/down and read/write values



**File Name** RSBTN32.OCX

**Object Type** RSButton

**Remarks** The RSButton control has several custom properties that allow you to monitor and control data. The RSButton control also has custom events that allow you to be notified when data has changed, or if data has finished its request or poke.

**Note** When you create and distribute applications that use the RSButton control, certain files need to be installed in the WINDOWS\SYSTEM directory. (See also [Redistributing Files](#)). The Setup Wizard included with Visual Basic provides tools to help create setup programs for installing your applications.

## Description for the RSButton Control

RSButton is a Visual Basic OLE Custom Control (OCX). It can be accessed directly by Visual Basic, Microsoft Access, or any Microsoft compatible OLE control container.

- The RSButton control is designed to be used with any Dynamic Data Exchange Server and provides enhanced performance when used with those that conform to AdvanceDDEprotocol. In order for RSButton to communicate via AdvanceDDE or XL\_Table protocols, RSJunctionBox, the module that provides RSButton with the ability to use those protocols, must be purchased separately from Rockwell Software Inc. If you choose to write your own DDE server, the AdvanceDDE protocol can be obtained from Rockwell Software Inc., by contacting Technical Support.
- The RSButton control can be easily configured by setting its properties, without writing a single line of code.
- The RSButton control can be used as a bound control. It automatically handles updating and displaying data from the Data control, which supports ODBC, Microsoft Access, dBASE, Foxpro, Paradox, Btrieve and several other database types.
- The RSButton control also supports the Paste Link method from **RS**LinX or WINtelligent LINX, Excel or other servers for transferring DDE Link data via the clipboard.
- The RSButton control has built in error notification and handling. If an error has occurred in the data stream, the error can be displayed in the control's caption property, or handled via code in one of the events.

## Property List

(AboutBox)

(Custom)

[AllowChangeEvent](#)

[ArrayItemExclusive](#)

[AutoColumns](#)

[BackColor](#)

[BackStyle](#)

[BevelBorderEnabled](#)

[BevelHeight](#)

[BevelHighlight](#)

[BevelShadow](#)

[BevelStyle](#)

[BevelWidth](#)

[BorderColor](#)

[BorderInner](#)

[BorderInnerColor](#)

[BorderStyle](#)

[BorderWidth](#)

[BottomBorder](#)

[ButtonBorderColor](#)

[ButtonBorderHeight](#)

[ButtonBorderStyle](#)

[ButtonBorderWidth](#)

[ButtonFaceColor](#)

[ButtonHighlight](#)

[ButtonShadow](#)

[ButtonStyle](#)

[Caption](#)

[CaptionBackColor](#)

[CaptionColor](#)

[CaptionShadow](#)

[CaptionShadowColor](#)

[CaptionTransparency](#)

[CaptionX](#)

[CaptionY](#)

[Clip](#)

[DataChanged](#)

[DataField](#)

[DataSource](#)

[DataUpdate](#)

[DataValue](#)  
[DisplayCaption](#)  
[DisplayCaptionVertically](#)  
[DisplayDownCaption](#)  
[DisplayPicture](#)  
[DownCaption](#)  
[DownCaptionColor](#)  
[DownValue](#)  
DragIcon  
DragMode  
[DrawDisabledShadow](#)  
[DrawFocusBorder](#)  
Enabled  
[ExpressionForRead](#)  
[ExpressionForWrite](#)  
[FlashEnabled](#)  
[FlashOn](#)  
[FlashSpeed](#)  
[FlashTime](#)  
[Font](#)  
[GroupExclusive](#)  
Height  
HelpContextID  
Index  
Left  
[LeftBorder](#)  
[LightNotEnabledColor](#)  
[LightOffColor](#)  
[LightOnColor](#)  
[LinkErrorDisplay](#)  
[LinkErrorNumber](#)  
[LinkErrorString](#)  
[LinkItem](#)  
[LinkMode](#)  
[LinkServer](#)  
[LinkTip](#)  
[LinkTipBackColor](#)  
[LinkTipForeColor](#)  
[LinkTipText](#)  
[LinkTopic](#)  
Name  
[NumberOfDataVa](#)

[lues](#)  
[Picture](#)  
[PictureStretch](#)  
[PictureUp](#)  
[PokeLength](#)  
[PokeStartIndex](#)  
[RequestLength](#)  
[RequestStartIndex](#)  
[X](#)  
[RightBorder](#)  
[RoundButton](#)  
[ScreenPriority](#)  
[ServerCanUpdateValue](#)  
[Symbol](#)  
[TabIndex](#)  
[TabStop](#)  
[Tag](#)  
[Top](#)  
[TopBorder](#)  
[UpValue](#)  
[UseInPoke](#)  
[UseInRequest](#)  
[Value](#)  
[Visible](#)  
[WhatsThisHelpID](#)  
[Width](#)  
[WriteStyle](#)  
[WriteValue](#)

## Event List

[Change](#)

[Clicks](#)

[DblClick](#)

[Down](#)

[DragDrop](#)

[DragOver](#)

[GotFocus](#)

[KeyDown](#)

[KeyPress](#)

[KeyUp](#)

[LinkError](#)

[LinkItemNotSupported](#)

[d](#)

[LinkItemSupported](#)

[LinkNotify](#)

[LinkOutOfMemory](#)

[LinkServerDisconnect](#)

[d](#)

[LostFocus](#)

[MouseDown](#)

[MouseMove](#)

[MouseUp](#)

[PokeCompleted](#)

[RequestCompleted](#)

[Up Event](#)

## Method List

[Container](#)

[DoPoke](#)

[DoRequest](#)

[Drag](#)

[LinkPoke](#)

[LinkRequest](#)

[Move](#)

[Object](#)

[Parent](#)

[SetFocus](#)

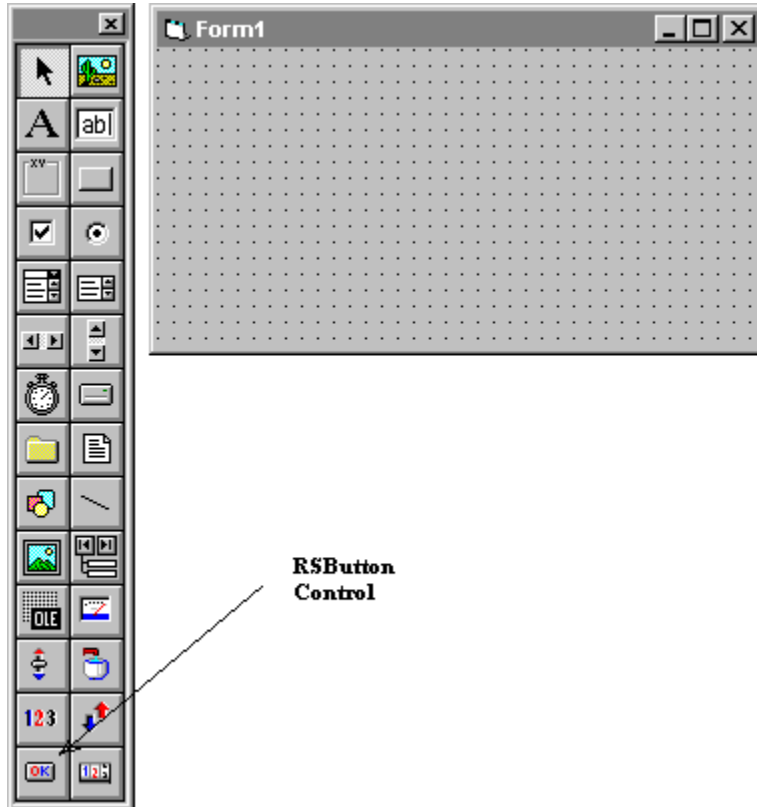
[ShowWhatsThis](#)

[ZOrder](#)

## Loading RSBUTTON (Adding RSBUTTON to your VB Project)

To use RSBUTTON in Visual Basic, you must add the RSBUTTON control to the Visual Basic toolbox.

1. Start Visual Basic.
2. From Visual Basic, select Custom Controls from the Tools menu.
3. Scroll through the available controls listed. Choose "Rockwell Software RSBUTTON" and choose OK.



The RSBUTTON icon is added to the Visual Basic Toolbox.



## Autoloading RSButton

You can configure Visual Basic to automatically load the RSButton control when you start a new project in Visual Basic.

To configure Visual Basic to automatically load RSButton:

1. Start Visual Basic and open AUTO32LD.VBP
2. Choose Custom Controls from the Tools menu. (The “Custom Controls” Dialog appears.)
3. Scroll down the list box until “Rockwell Software RSButton” is visible, select it and choose OK. The control is added to the project list.
4. Choose Save Project from the File menu.

**Note:**        *If “Rockwell Software RSButton” does not appear in the list of available custom controls, you may have to click the Browse button and manually select RSBTN32.OCX and RSBTNX32.DLL, which are located in the C:\WINDOWS\SYSTEM directory. When attempting to load RSBTNX32.DLL the message “Unable to load control from RSBTNX32.DLL” may appear. This occurs because this DLL contains supporting code for the RSButton OCX and does not contain the actual control. Choose OK to continue.*

## Placing the RSButton Control on a Form

Creating a new RSButton control and placing it on a form is as simple as point, click and drag.

1. Select the RSButton tool in the Visual Basic toolbox.
2. Position the mouse on the form at the location where you want to draw the control.
3. Click and drag to draw the outline of the RSButton control on the form.
4. When you release the mouse, the new RSButton control is placed in the location you specified.
5. Or you may double click on the RSButton tool and a RSButton control will be placed in the center of your form.

## **Distributing RSButton Applications**

Please read the license agreement that was shipped with this package. You are bound by the licensing restrictions contained in that document.

## Visual Basic Floating Menu

Click the right mouse button anywhere within an RSBUTTON control to display the floating menu. Once the menu appears, use the left mouse button to select a menu item. Clicking on menu items followed by three periods bring up additional dialog boxes for performing control functions. The Properties menu item will bring up the [Custom Property Tabs](#).

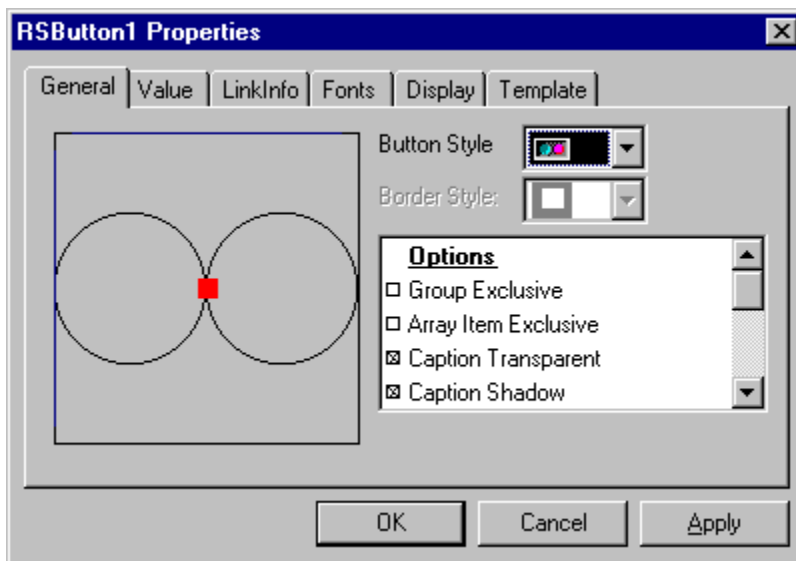


The following table describes the purpose of each item on the floating menu.

Menu item	Description
Cut	Deletes the selected item and copies it to the clipboard.
Copy	Copies the selected item to the clipboard.
Paste	Pastes the contents of the clipboard onto the form.
Delete	Deletes the selected item.
Bring To Front	Brings the selected item to the top of the z-order.
Send To Back	Sends the selected item to the bottom of the z-order.
View Code	Displays the selected item's code window.
Align to Grid	Aligns the control's Top and Left property to the form's grid.
Properties	Displays Custom Property Dialog.
Paste Link	Supports pasting a link to a valid DDE source.

## Using the RSBUTTON Custom Property Tabs

The RSBUTTON OCX dialog box provides sets of options grouped on separate tabs. As you click each tab, the controls in the dialog box change to allow you to edit a different set of options. Any options that are not appropriate for the current chart type or situation are grayed. The following illustration shows an example of an RSBUTTON OCX dialog box.



*Note: Switching between tabs will force an automatic "apply" to the control.*

See also

[General Tab](#)

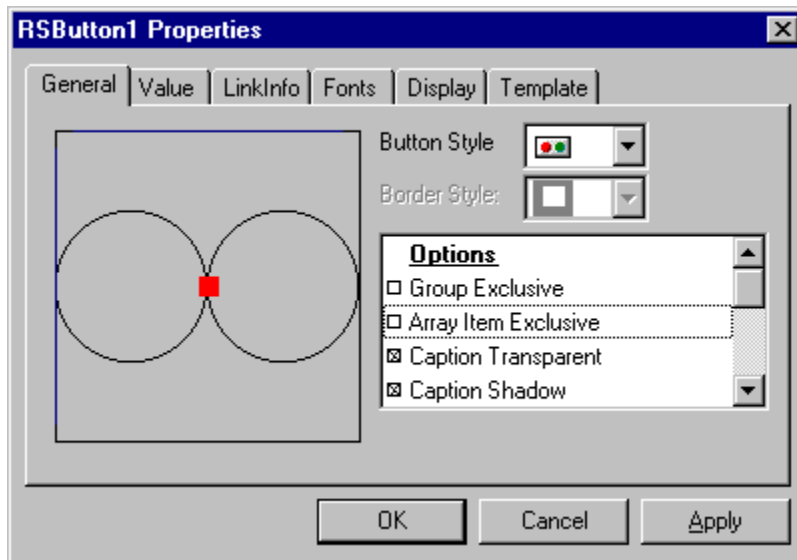
[Value Tab](#)

[LinkInfo Tab](#)

[Display Tab](#)

[Template Tab](#)

## General Tab



See also

[Setup Area](#)

[General Options List Box](#)

[General Tab Combo Boxes](#)

## Setup Area

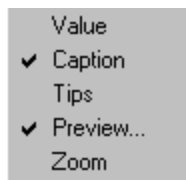
Use this display to change the attributes of the control. The graphic image of the button is determined by the currently selected Button Type. To change the size of the button inside its container grab and drag one of the four corners of the border.

If the Value and Caption are being displayed on the RSTButton control, their positions can be adjusted by moving their placeholders within the setup area. The Value display is represented by a blue square, and the Caption display is represented by a red square.

RSTButton's shadow position can also be changed here by clicking and dragging the shadow within the setup area. The Shadow can be turned off and on in the Options list box or with the right-click menu described below.

The size of the RSTButton control within the control's borders can be adjusted using the sizing handles. Each of the four corners of the Setup Area has a sizing handle, and by clicking and dragging those sizing handles you can adjust the TopBorder, BottomBorder, LeftBorder, and RightBorder properties of the RSTButton.

While in the Setup Area, click the right mouse button to bring up the floating menu.



From this floating menu you can enable/disable the **Value**, **Caption**, or **LinkTips** properties. You can also display the **Preview** window, which allows you to view a floating display of your control. The **Zoom** option enlarges the OCX Display to allow you to make more detailed adjustments to the button image.

See also

[General Tab](#)

## General Options List Box

This list box lists all of the general options for the control. You can enable or disable these properties from this list box by clicking in the appropriate check box. Following is a list of the options available:

[Group Exclusive](#)

[Array Item Exclusive](#)

[Caption Transparent](#)

[Caption Shadow](#)

[Round Button](#)

[Bevel Border Enabled](#)

[Draw Focus Border](#)

[Server Update Button State](#)

[Display Picture](#)

[Picture Stretch](#)

[Support Button Down Caption](#)

[Draw Disabled Shadow](#)

See also

[General Tab](#)



## General Tab Combo Boxes

**Button Style Combo Box.** Set the button's [ButtonStyle](#) property here. These are the graphical selections for Normal, Toggle, Horizontal Led Pair, Horizontal Reverse Led Pair, Vertical Led Pair, Vertical Reverse Led Pair, Diode, Check Box, Square, Radio Button, Horizontal Switch, Horizontal Reverse Switch, Vertical Switch, Vertical Reverse Switch and Push Pop.

**Border Style Combo Box.** Sets the button's [ButtonBorderStyle](#) property. This is active for the Button Styles of Normal, Toggle, Square, Horizontal Switch, Horizontal Reverse Switch, Vertical Switch and Vertical Reverse Switch.

When finished, press the **OK** button to save any changes or the **Cancel** button to exit without saving changes.

See also

[General Tab](#)

## Value Tab

The screenshot shows a Windows-style dialog box titled 'RSButton1 Properties'. It has five tabs: 'General', 'Value', 'LinkInfo', 'Fonts', and 'Template'. The 'Value' tab is selected. Inside the dialog, there are several input fields and a dropdown menu. The 'Up:' label is above a text box containing '0'. The 'Down:' label is above a text box containing '1'. To the right of these are 'Up Caption:' and 'Down Caption:' labels, each followed by an empty text box. Below these is a 'Write Style:' label followed by a dropdown menu showing 'All Values'. At the bottom are 'OK', 'Cancel', and 'Apply' buttons.

Property	Value
Up Value	0
Down Value	1
Up Caption	
Down Caption	
Write Style	All Values

- **Up Value.** Sets the value for the button when it is in the up position.
- **Down Value.** Sets the value for the button when it is in the down position.
- **Up Caption.** Sets the default caption when the button is in the up position.
- **Down Caption.** Sets the default caption when the button is in the down position.
- **Write Style.** Sets the write style to the DDE server.

See also

[UpValue Property](#)

[DownValue Property](#)

[Caption Property](#)

[DownCaption Property](#)

[WriteStyle Property](#)

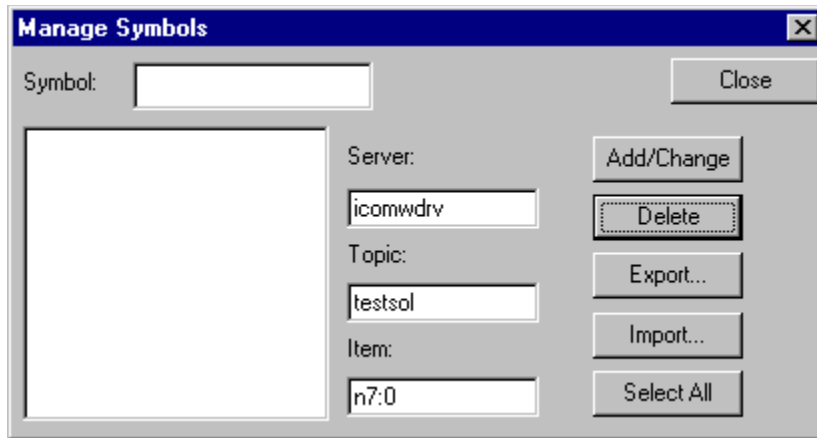
## LinkInfo Tab

The screenshot shows the 'RSButton1 Properties' dialog box with the 'LinkInfo' tab selected. The 'Symbol' dropdown is set to 'RSI', 'Server' to 'icomwdrv', 'Topic' to 'testsol', and 'Item' to 'n7:0,110'. The 'LinkMode' is set to 'Automatic'. The 'AutoColumns' checkbox is checked. A 'Manage Symbols' button is located below the 'LinkMode' dropdown. The dialog has standard 'OK', 'Cancel', and 'Apply' buttons at the bottom.

- Symbol.** This combo box is used to choose the name of the Symbol that has been previously defined for a specific Server, Topic and Item in a DDE link. Select a Symbol from the combo box or type the Symbol's name directly in. When a defined Symbol is chosen, then the Server, Topic and Item combo boxes on the LinkInfo page will automatically be filled in with the appropriate data. The *Manage Symbols* section below describes how to define a symbol. (See also [Symbol](#))
- Server.** This combo box determines the application or Server name that the RSButton control is linked to. If the Server name has been previously used it can be chosen from the combo box list; if not, enter a Server name in the combo box.
- Topic.** This combo box determines the LinkTopic portion of the data link string, which the RSButton control uses for addressing in a DDE link. If the Topic has been previously used it can be chosen from the combo box list; if not, enter a Topic in the combo box.
- Item.** This combo box determines the LinkItem portion of the data link string, which the RSButton control uses for addressing in a DDE link. If the Item has been previously used it can be chosen from the combo box list; if not, enter an Item in the combo box.
- LinkMode.** Sets the type of link to be used for a DDE conversation and activates the connection. Available options for the LinkMode property are:
  - 0 = None** - No DDE connection is established between the control and **Server**.
  - 1= Automatic** - A "HOT link". The **Server** automatically updates the control when data changes.
  - 2 = Manual** - A "COLD link". The client has to issue a DoRequest method to get data from the **Server**.
  - 3 = Notify** - Causes Visual Basic to Fire the LinkNotify event which the user can respond to.



**Manage Symbols.** This button is used to display the Manage Symbols dialog (see below), which is used to define Symbols for DDE links to the control.



To define a new Symbol for a specific Server, Topic, and Item in a DDE Link, enter the name for the new Symbol in the **Symbol** text box. Place the DDE Link Server name in the **Server** text box, the DDE Link Topic name in the **Topic** text box, and the DDE Link Item name in the **Item** text box.

Click on the **Add/Change** button to add a new Symbol name to the list box. To edit an existing Symbol, select the appropriate name from the list box and click on the **Add/Change** button to change the Server, Topic, or Item text. To delete an existing Symbol, select the appropriate Symbol name from the list box and click on the **Delete** button to delete that Symbol.

Press **Export** to write the symbol information to a .RSS file (Rockwell Software Symbol). This file is a text file with the following format:



```
[SYM]
RSI=icomwdrv|testsol|n7:0
icom=icomwdrv|testsol|n7:22
excel=excel|[book1]sheet1|R1C1
```

This is essentially a .INI file format with a separate *entry*= line for each symbol. This file can be read by a programmer to utilize symbol information in an application.

Click **Import** to import symbol information from an existing .RSS file.

Click **Select All** to select all symbols displayed.

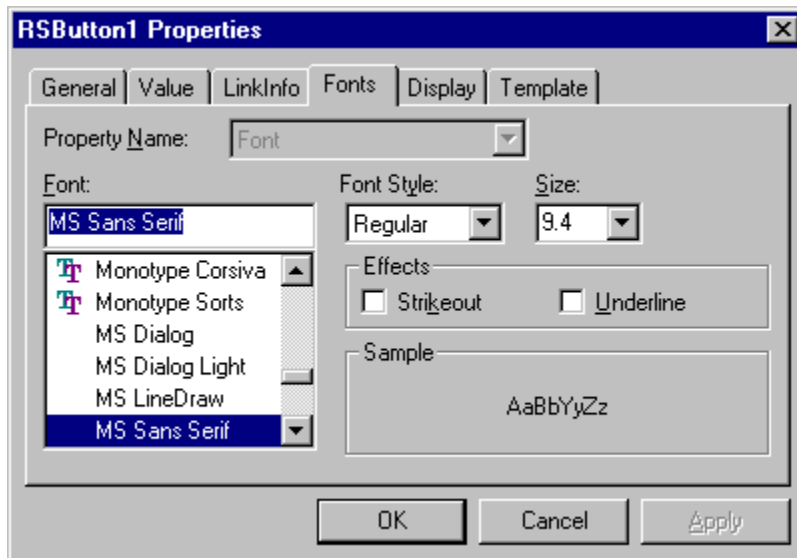
When finished, press the **Close** button to close the dialog box and save any changes.

-  **Paste Link.** This button is enabled when the user has copied specific DDE Link information from another application (for example, Microsoft Excel) to the clipboard. Pressing this button will then fill in the appropriate Server, Topic, and Item for the DDE link.
-  **Expression.** This feature is available only when the RSJunctionBox module is installed on the computer.
- **AutoColumns.** This check box is used to enable or disable the AutoColumns property for the control. AdvancedDDE servers can provide data in block format.

When finished, press the **OK** button to save any changes or the **Cancel** button to exit without saving changes.

## Font Tab

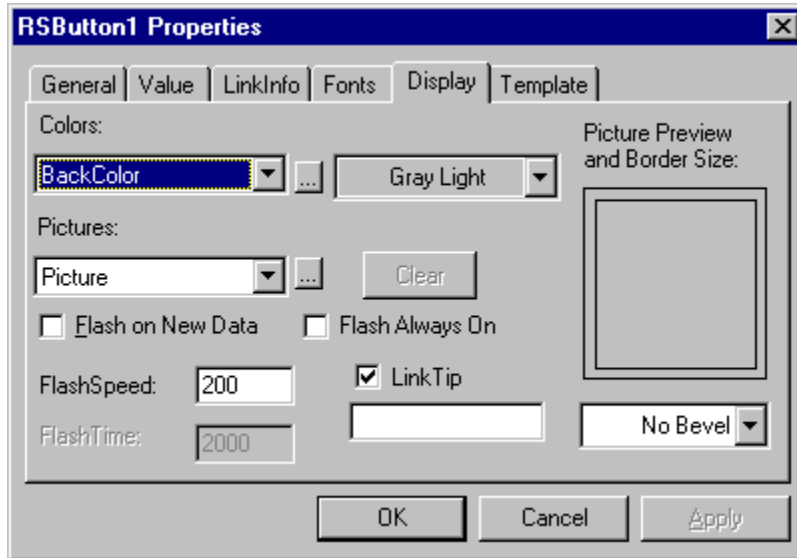
The Font tab is where you can change the name, style and size of the RSBUTTON's fonts.



- **Property Name.** Selects the property of the control that you want to format.
- **Font.** Select a font from the list of fonts installed on your windows system.
- **Font Style.** Select a font style from the list of supported styles for the font you selected.
- **Size.** Select a size from the list of valid sizes for the font you selected. You can also type a valid size in the Size field.
- **Strikeout.** Select the Strikeout checkbox if you want strikeout applied to the text.
- **Underline.** Select the Underline checkbox if you want the text to be underlined.

## Display Tab

All of the display properties for the RSButton can be set using the Display tab. Display properties include things like colors, pictures, and bevel styles. This section describes the setting of these properties using this tab.



- **Colors.** Setting of colors involves selecting the color property (Example : BackColor) and then choosing a color. (See also [RSButton Color Properties](#))  
Select a color property by highlighting it in the drop down combo box and then set the color you want to apply to the property from either the basic color palette (activated by clicking on the ellipsis button) or the drop down colors combo box. The drop down colors includes windows default scheme colors in addition to standard colors. If you want your control to inherit the colors of the Windows desktop color schemes, use this drop down combo box. The properties set to Windows desktop colors will automatically take up the new desktop's colors' setting (if the project is compiled and made into an executable, then the exe control colors would also inherit Windows desktop colors.)
- **Picture.** You can assign a picture to the RSButton control's Picture property with the Picture combo box. Click on the three dot button to bring up a File Open dialog box. With this dialog box you can choose any \*.bmp, \*.ico or \*.wmf file to be used as the picture. After selecting a file, that file will be displayed in the *Picture Preview and Border Size* window. To clear this picture, simply click on the **Clear** button.
- **Flash on New Data.** The FlashEnabled property is turned off and on with this check box. When turned on, the RSButton control will flash when the RSButton's value changes for the amount of time designated by the FlashTime property, at the rate designated by the FlashSpeed property.
- **Flash Always On.** The FlashOn property is turned off and on with this check box. When turned on, the RSButton control will continually flash at the rate designated by the FlashSpeed property.
- **FlashSpeed.** Sets the FlashSpeed property by entering a number here. FlashSpeed is measured in milliseconds. FlashSpeed is the rate at which the RSButton control will flash on and off.
- **FlashTime.** Set the FlashTime property value by entering a number here. FlashTime is measured in milliseconds. FlashTime is the amount of time that the RSButton control will continue to flash.
- **LinkTip.** The RSButton provides the ability to add tool tips for the control. The tool tip can be turned on by checking this check box. Text for the tool tip is entered in the text box below the check box. If no text is entered, then the tool tip defaults to the DDE Link details or *Symbol* name for the control, if any.
- **Bevel Style.** This combo box allows you to select a bevel style for the RSButton. The available bevel styles are:

<i>No Bevel</i>	<i>Marble</i>	<i>Thick</i>
<i>Indented</i>	<i>Bevel</i>	<i>Stripe</i>

The amount of 3D effect on the bevel is determined by the border size as set above.

## Color Properties for RSButton

[BackColor](#)

[BevelHighlight](#)

[BevelShadow](#)

[BorderColor](#)

[BorderInnerColor](#)

[ButtonBorderColor](#)

[ButtonFaceColor](#)

[ButtonHighlight](#)

[ButtonShadow](#)

[CaptionBackColor](#)

[CaptionColor](#)

[CaptionShadowColor](#)

[DownCaptionColor](#)

[LightNotEnabledColor](#)

[LightOffColor](#)

[LightOnColor](#)

[LinkTipBackColor](#)

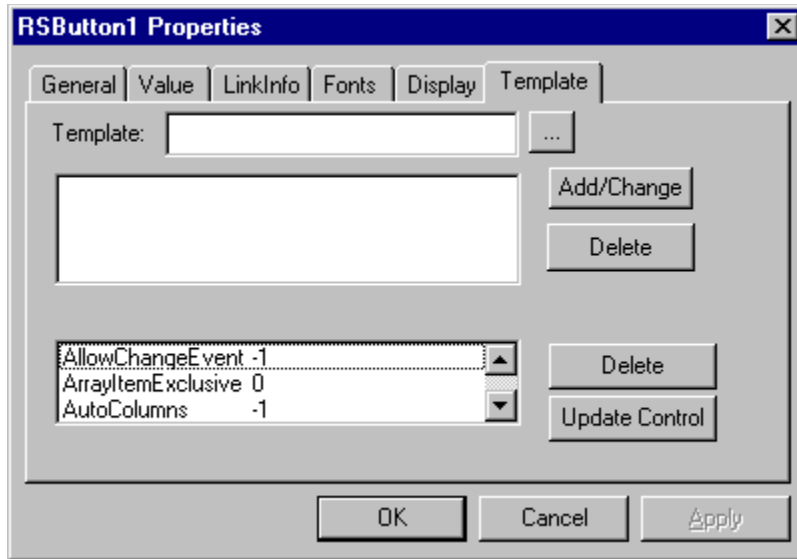
[LinkTipForeColor](#)


See also

[Display Tab](#)

## Template Tab

You can save the property settings of a control and apply them again to different controls using templates. Once a template of properties has been saved, it can be applied to another RSButton control and the properties settings applicable to that control are changed as per the template. Using this Custom Property Page Tab, templates can be Added, Changed and Managed. These property templates are saved in a file with an “.rwc” extension. Each property template is saved with a distinct name. Duplicate names would replace any other template with the same name in that file.



- **Template.** This field displays the name of the template to be applied, added or changed. The template file can be selected by clicking on the ellipsis button  next to the field. Available templates names are displayed in the list below this field and can be selected by clicking on the template name (selected template is highlighted).
- **Add/Change Button.** New templates or changes in an existing template can be saved by clicking on this button. The template is saved in the selected \*.rwc file. The template file can be selected as described above.
  - **Delete Button.** Use this button to delete a template from the file. Only the template is removed from the file with all other templates in that file remaining.
  - **Properties.** This field, below the template list, describes the properties and their settings for the selected template. Properties can be deleted if not required to be in the template.
  - **Delete Property Button.** If you want to take out a property from the template, select the property from the list (selected property is highlighted) and click on this button.
  - **Update Control Button.** To apply an existing template to the control, select the template and click on this button. This is the only button that actually applies the property settings to the control. You can still click the **Cancel** button to exit without saving the template updates to the control.

**Note:** *Properties have to be set using the other property pages and then saved in templates. Property values cannot be set on this property page*



## Down Event

The **Down** event occurs when the button is put into its down state.

1. Place a RSButton on your form and set ButtonStyle = 3(Horizontal Led Pair), LightNotEnabled = Black, LightOnColor = Green, LightOffColor = Red, DownValue = 1, UpValue = 0 and Value =0.
2. Type in the following code:

```
Private Sub RSButton2_Down(ByVal DownValue As Double, ByVal ButtonIndex  
As Integer)
```

```
    Dim Msg As String
```

```
    Msg = "Button is down"
```

```
    MsgBox Msg, 48, "Down Event"
```

```
End Sub
```

3. Run your program. The RSButton control will have a pair of horizontal led buttons with the left one being red and the right black. When you click on the right (black) button it will turn green, the Down event will be fired and your message box will appear.

## LinkError Event

The **LinkError** event occurs when there is an error during a DDE conversation. This event is recognized only as the result of a DDE-related error that occurs when no Visual Basic code is being executed. The error number and error string are passed as arguments.

1. Place a RSButton control on your form.
2. Set the RSButton1 **AllowChangeEvent** = True, **LinkServer** = "Excel", **LinkTopic** = "[Book1]Sheet1", **LinkItem** = "R1C1" and **LinkMode** = 1-Automatic.
3. Type in the following code.

```
Private Sub RSButton1_LinkError(ByVal iRet As Integer, ByVal ErrorString  
As String)  
    Dim Msg as String  
    Msg = "Link Error Occurred"  
    MsgBox Msg, 48, "Error Report"  
End Sub
```

4. Start Excel and run your Visual Basic program.
5. While your program is running, close Book1 in Excel. Click on the button control and the error message will occur since the link topic has been closed.

## LinkItemNotSupported Event

The **LinkItemNotSupported** event occurs when the button's LinkItem has an incorrect format.

1. Place a RSButton control on your form.
2. Set the RSButton1 **AllowChangeEvent** = True, **LinkServer** = "Excel", **LinkTopic** = "[Book1]Sheet1", **LinkItem** = "R0C1" and **LinkMode** = 1-Automatic
3. Type in the following code.

```
Private Sub RSButton1_LinkItemNotSupported()  
    Dim Msg as String  
    Msg = "Link Item Not Supported"  
    MsgBox Msg, 48, "Error Report"  
End Sub
```

4. Run the program.
5. The error message will occur since R0C1 is not a valid link item.

## LinkItemSupported Event

The **LinkItemSupported** is fired when the DDE communication is initialized and the specified LinkItem is supported by the server.

1. Place a RSButton control on your form.
2. Set the RSButton1 **AllowChangeEvent** = True, **LinkServer** = "Excel", **LinkTopic** = "[Book1]Sheet1", **LinkItem** = "R1C1" and **LinkMode** = 1-Automatic
3. Type in the following code.

```
Private Sub RSButton1_LinkItemSupported()  
    Dim Msg as String  
    Msg = "Link Item " & RSButton1.LinkItem & " is  
    supported by server " & RSButton1.LinkServer  
    MsgBox Msg, 48, "Link Report"  
End Sub
```

4. Run the program.
5. The message will occur since R1C1 is a valid link item.

## LinkNotify Event

The **LinkNotify** event occurs when the source has changed the data defined by the DDE link, if the **LinkMode** property of the RSTButton control is set to 3-Notify.

```
Private Sub RSTButton1_LinkNotify()  
    Dim Msg as String  
    Msg = "Data Value has Changed"  
    MsgBox Msg, 48, "Link Report"  
End Sub
```

## LinkOutOfMemory Event

The **LinkOutOfMemory** event occurs when the client (RSButton) has exhausted its memory resources.

```
Private Sub RSButton1_LinkOutOfMemory()  
    Dim Msg as String  
    Msg = "Out of memory to carry on Link activities"  
    MsgBox Msg, 48, "Error Report"  
End Sub
```

## LinkServerDisconnected Event

This event is another tool for efficient DDE link error management. The **LinkServerDisconnect** event occurs when the RSTButton control is connected to a server and that server becomes unavailable.

1. Place a RSTButton control on a form. Set the following properties: **AllowChangeEvent** = True, **LinkServer** = "Excel", **LinkTopic** = "[Book1]Sheet1", **LinkItem** = "R1C1" and **LinkMode** = 1 - Automatic
2. Type in the following code.

```
Private Sub RSTButton1_LinkServerDisconnected()  
    Dim Msg as String  
    Msg = "Link Server Disconnected"  
    MsgBox Msg, 48, "Error Report"  
End Sub
```

3. Start Microsoft EXCEL.
4. Run the program.
5. Type in a value in Row 1 Cell 1 of your Excel spreadsheet.
6. The button control value should change accordingly.
7. Now shutdown EXCEL. *LinkServerDisconnected* event is now fired and the above error message should be displayed.

## LinkUnableToConnectToServer

The LinkUnableToConnectToServer event occurs when the RSButton attempts to connect to a server that is unavailable.

1. Place a RSButton control on a form. Set the following properties: **AllowChangeEvent** = True, **LinkServer** = "Excel", **LinkTopic** = "[Book1]Sheet1", **LinkItem** = "R1C1" and **LinkMode** = 1 - Automatic
2. Type in the following code.

```
Private Sub RSButton1_LinkUnableToConnectToServer()  
    Dim Msg as String  
    Msg = "Unable to connect to link server"  
    MsgBox Msg, 48, "Error Report"  
End Sub
```

3. Run the program without having started Excel. This error handling event should be fired.



## PokeCompleted Event

The **PokeCompleted** event occurs when the RSButton control completes a DDE Poke.

```
Private Sub RSButton1_PokeCompleted(ByVal iRet As Integer)
    'Once the Poke is completed, tell the user.
    RSButton1.Caption = "The Download is processed."
End Sub
```

```
Private Sub RSButton1_Clicks(ByVal Value As Double,      ByVal
    ButtonIndex As Integer)
    'Once the Poke is completed, the Excel cell is updated.
    Const manual = 2, none = 0
    RSButton1.UseInPoke = True
    RSButton1.LinkMode = none
    RSButton1.LinkServer = "Excel"
    RSButton1.LinkTopic = "[Book1]Sheet1"
    RSButton1.LinkItem = "R1C1"
    RSButton1.LinkMode = manual
    RSButton1.DoPoke
    RSButton1.LinkMode = none
End Sub
```

**Note:** This event is very useful for preventing “outrunning the server”, or sending additional data to a server while it is still processing the previous data.

## RequestCompleted Event

```
Private Sub RSButton1_RequestCompleted(ByVal iRet As Integer)
    'Once the Request is completed, tell the user.
    RSButton1.Caption = "The Request has completed successfully!"
End Sub

Private Sub RSButton1_Clicks(ByVal Value As Double, ByVal ButtonIndex As Integer)
    'Once the Request is completed, cell R1C1 is updated.
    Const manual = 2, none = 0
    RSButton1.UseInRequest = True
    RSButton1.LinkMode = none
    RSButton1.LinkServer = "Excel"
    RSButton1.LinkTopic = "[Book1]Sheet1"
    RSButton1.LinkItem = "R1C1"
    RSButton1.LinkMode = manual
    RSButton1.DoRequest
    RSButton1.LinkMode = None
End Sub
```

**Note:** *This event is very useful for preventing "outrunning the server", or sending additional data to a server while it is still processing the previous data.*

## Up Event

The **Up** event occurs when the RSButton is put into its up state.

1. Place a RSButton on your form and set ButtonStyle = 3(Horizontal Led Pair), LightNotEnabled = Black, LightOnColor = Green, LightOffColor = Red, DownValue = 1, UpValue = 0 and Value =0.
2. Type in the following code:

```
Private Sub RSButton2_Up(ByVal UpValue As Double, ByVal ButtonIndex As Integer)
```

```
    Dim Msg As String
```

```
    Msg = "Button is up"
```

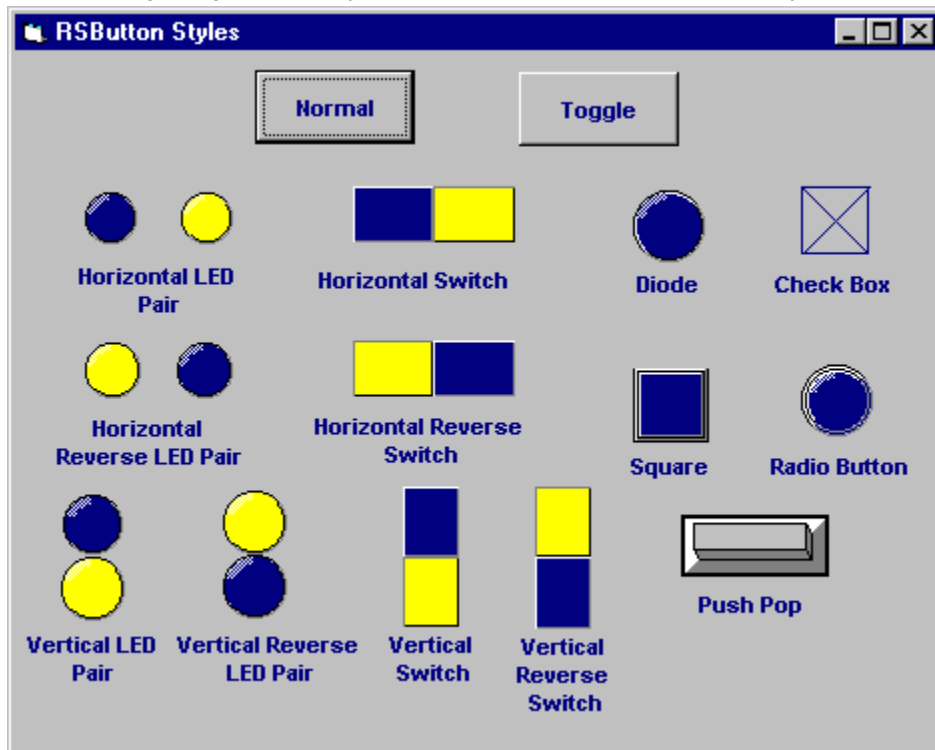
```
    MsgBox Msg, 48, "Up Event"
```

```
End Sub
```

3. Run your program. The RSButton control will have a pair of horizontal led buttons with the left one being red and the right black. When you click on the left (red) button it will turn black, the Up event will be fired and your message box will appear.

## Using Button Styles

The following form gives an example of each of the RSTButton control's Button Styles.

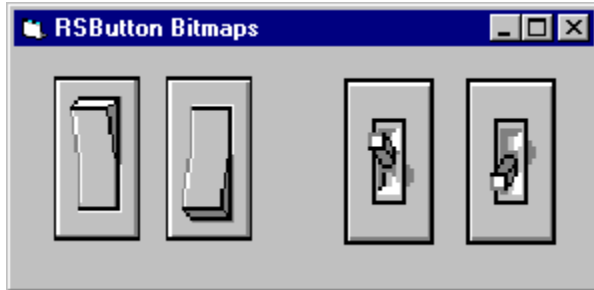


See also

[ButtonStyle Property](#)

## Displaying Bitmaps on the RSButton Control

It is possible to create a RSButton control which displays bitmaps. An example would be an RSButton that will display one bitmap in the up state and a different bitmap in the down state. Below is a form containing two examples of this situation. Included with the RSButton control are four bitmaps that will allow you to recreate these two examples; these four bitmap files can be found in the C:\RSWKSHOP\IRSTOOLBX\DEMO directory.

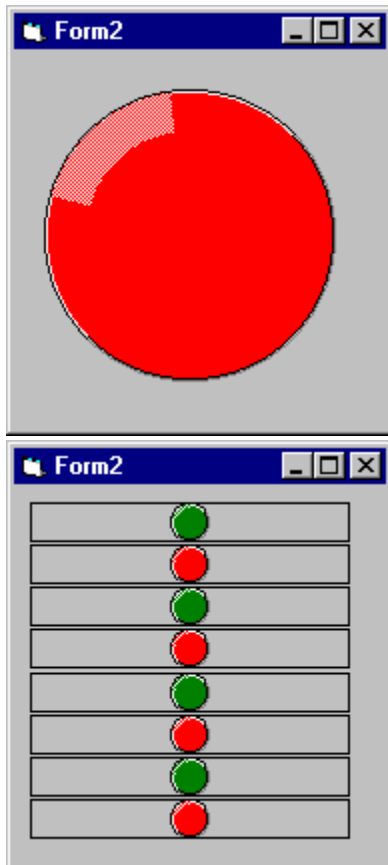


1. Place two RSButton controls on your form. Set **ButtonStyle** = 1 (Toggle), **BevelBorderEnabled** = False, **DrawFocusBorder** = False, and **DisplayPicture** = True on both RSButton controls.
2. Right click on the first RSButton control and bring up the Custom Properties page.
3. Go to the [Display tab](#) and under Pictures select PictureUp. Then click on the ellipsis button. This will allow you to browse through your directories and choose a bitmap for the RSButton to display when it is in the Up state. Choose SWITCH1.BMP. Now go back to the Pictures combo box and select Picture. Follow the same procedure, selecting SWITCH2.BMP for the RSButton to display when it is in the Down state. Click on Apply and OK to apply the changes and close the Custom Properties page.
4. Now right click on the second RSButton control and bring up the Custom Properties page.
5. On the Display tab, again select PictureUp from the Pictures Combo Box. This time browse and select SWITCHA.BMP. Go back to the Pictures combo box and select Picture. Follow the same procedure again, choosing SWITCHB.BMP as the Picture bitmap. Click on Apply and OK to apply those changes and close the Custom Properties page.
6. Run your program. Click each of the RSButton controls up and down using your mouse to view the up and down bitmaps. The effect you will see is a pair of switches being flipped up and down.

## Array Links Using RSButton

In the following example you can link a RSButton with a Boolean Excel database and create an array of eight buttons in either the up or down state based on the values read from the database.

1. Place a RSButton control on your form and set **ButtonStyle** = 6 (Diode), **LinkServer** = Excel, **LinkTopic** = [book1]sheet1, **LinkTopic** = R1C1:R8C1 and **LinkMode** = 1(Automatic). These can be set in the Properties window or by opening the Custom Properties page and using the LinkInfo tab.
2. Set **LightOffColor** to red and **LightOnColor** to green.
3. Open Excel and enter random values of 0 and 1 in cells R1C1 through R8C1.
4. Run your program. You will see the RSButton split into eight different buttons.



5. The buttons which are red linked with a value of 0 and are in the Up state. The green buttons are linked with a value of 1 and are in the Down state.

## Using RSTButton to Read from Datasource

With the RSTButton control and Visual Basic's Data control, you can create an application to utilize information from many types of existing databases. Creating a data-aware application with Visual Basic can be done easily through a few steps, and requires very little code.

The first thing you need to do to make a "data-aware" application is to add the Visual Basic Data control to your form. Next, you will have to specify the database you would like to get the information from. Once you have decided on the database, you must load the RSTButton control on the form and set its properties to "bind" the control to the Visual Basic's data control. Depending on the property settings you choose for DataSource, when you run this application you will be able to utilize data coming from your server or from a database.

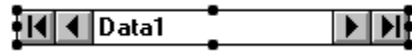
The RSTButton's AdvancedDDE technology combined with Visual Basic's data control gives you seamless access to many standard database formats, including Microsoft Access, Btrieve, dBASE, Microsoft FoxPro, and Paradox.

The following procedure gives you a brief overview of how to "bind" or use the RSTButton control and Visual Basic's data control in your application. We will use the color.mdb sample database that comes with RSTButton.

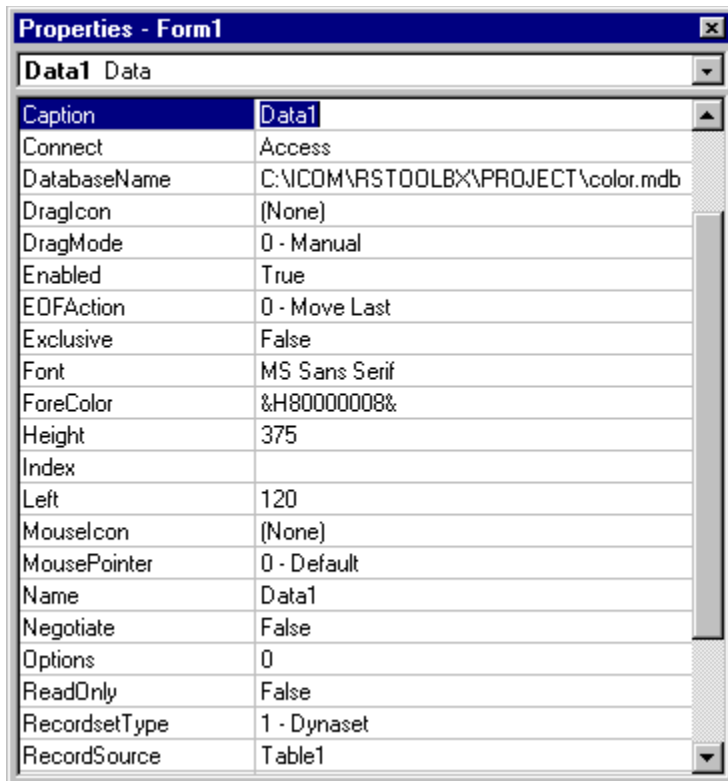
1. Select the RSTButton control in the toolbox and draw a RSTButton control on the form.
2. Select the Visual Basic Data control and draw a control on the form. The control icon looks like this:



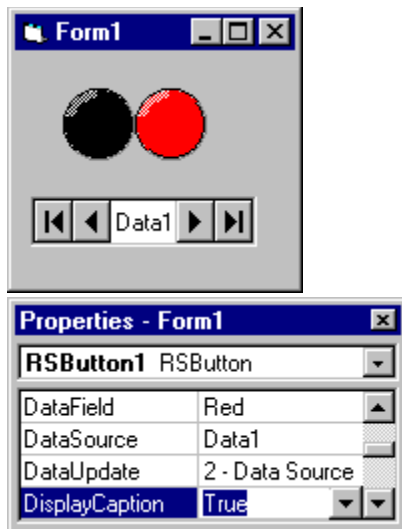
After you have drawn the control on the form, it will look like this, with the default caption as Data1.



3. In the properties window for the Visual Basic Data control, set the DatabaseName property to the filename you want to connect to. You must specify the path to the database only. We will use the example database located in C:\RSWKSHOP\RSSTOOLBX\DEMO\COLOR.MDB. You will set the **DatabaseName** property to this database.
4. For the **Connect** property, you must specify "ACCESS". Then set **RecordSource** = "TABLE1". The RecordSource is the name of the database file.



5. Click once on the RSTButton control and bring up its property window. In the properties window, set the **DataSource** = Data1, the **DataField** = Red, and set the **DataUpdate** = 2- Data Source. Also set **ButtonStyle** = 3 (Horizontal Reverse LED), **LightNotEnabled** = Black, **LightOnColor** = Green, **LightOffColor** = Red, **DownValue** = 100, and **UpValue** = 0.



6. Now run the application. You can use the arrow buttons in the data control to move through the records. When a value of 0 is read from the database the LED will display a red light.



## Using RSButton to Log to the Data Source

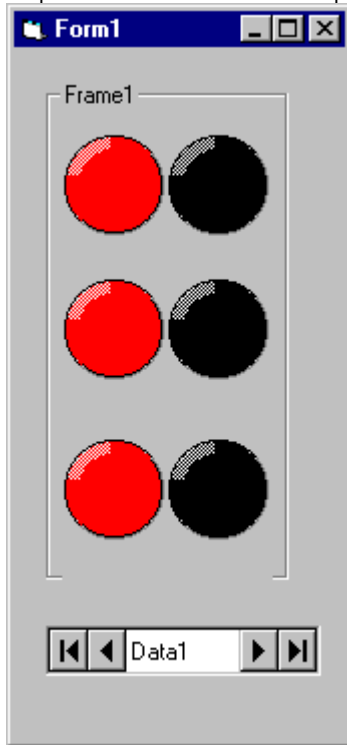
One of the biggest advantages of the RSButton control is its ability to log the RSButton's value to the **DataSource** specified by the Visual Basic Data control.

Before you can log any information to the DataSource, you must have an existing database to log the information. This database must contain at least one entry or record. You may have several field names depending on how many data points you have.

Once you have set up your database, all you need to do is set a few properties, write a few lines of code and run your project.

Follow the steps below to setup the RSButton control for logging. You may want to use the sample database file in the samples directory: C:\RSWKSHOP\IRSTOOLBX\DEMO\COLOR.MDB

1. On a form place a frame and in the frame put three RSButton controls. Also place a Visual Basic data control on the



form.

2. Using the previous example, in the Visual Basic data control, set the **DatabaseName** to color.mdb, the **RecordSource** = TABLE1 and for the **Connect** property, you must specify ACCESS .
3. In each of the RSButton controls, set the properties **GroupExclusive** = True, **DataUpdate** = 2 - Data Source, **DataSource** = Data1 and **UpValue** = 0. For RSButton1 set **DataField** = Red and **DownValue** = 100. For RSButton2 set **DataField** = Green and **DownValue** = 200. For RSButton3 set **DataField** = Blue and **DownValue** = 300.
4. Add the following code:

```
Private Sub RSButton1_Clicks(ByVal Value As Double, ByVal ButtonIndex As Integer)
    Dim sTemp As String
    sTemp = RSButton1.Value
    Data1.Recordset.AddNew
    RSButton1.Value = sTemp
    Data1.Recordset.Update
    Data1.Recordset.MoveLast
End Sub

Private Sub RSButton2_Clicks(ByVal Value As Double, ByVal ButtonIndex As Integer)
    Dim sTemp As String
```

```
sTemp = RSButton2.Value
Data1.Recordset.AddNew
RSButton2.Value = sTemp
Data1.Recordset.Update
Data1.Recordset.MoveLast
End Sub
```

```
Private Sub RSButton3_Click(ByVal Value As Double, ByVal ButtonIndex As Integer)
    Dim sTemp As String
    sTemp = RSButton3.Value
    Data1.Recordset.AddNew
    RSButton3.Value = sTemp
    Data1.Recordset.Update
    Data1.Recordset.MoveLast
End Sub
```

5. Once these changes have been made, simply run the program. If you click on RSButton1, the following values will be put into the database: Red=100, Green=0 and Blue=0. If you click on RSButton2, the following values will be put into the database: Red=0, Green=200 and Blue=0. If you click on RSButton3, the following values will be put into the database: Red=0, Green=0 and Blue=300.

## ArrayItemExclusive Property

<b>Description</b>	Enables/disables the exclusiveness of an RSButton within an array of RSButtons.
<b>Custom</b>	<a href="#">General Tab</a> in the Options List.
<b>Visual Basic</b>	[Form1.]RSButton1.ArrayItemExclusive[=setting]
<b>Remarks</b>	An array of RSButtons is defined as having the same Name property value, along with exclusive Index property values. To use the ArrayItemExclusive property effectively, each RSButton control within the array must have its ArrayItemExclusive property value set to True.
<b>Data Type</b>	Boolean

## BevelBorderEnabled Property

<b>Description</b>	Enables/disables the beveled border of a three dimensional button.
<b>Custom</b>	Set in the Options list box on the <a href="#">General Tab</a> .
<b>Visual Basic</b>	[Form1.]RSButton1.BevelBorderEnabled[=setting]
<b>Remarks</b>	True enables the bevel border. False disables the bevel border.
<b>Data Type</b>	Integer

## ButtonBorderColor Property

<b>Description</b>	Sets the color of the focus border.
<b>Custom</b>	<a href="#">Display Tab.</a>
<b>Visual Basic</b>	[Form1.]RSButton1.ButtonBorderColor[=setting]
<b>Remarks</b>	The focus border is displayed when the RSButton has focus. In order to display the focus border, DrawFocusBorder must be set to True.
<b>Data Type</b>	Color

## ButtonBorderHeight Property

<b>Description</b>	Sets the height of the button's top and bottom borders.
<b>Custom</b>	N/A in Custom property page.
<b>Visual Basic</b>	[Form1.]RSButton1.ButtonBorderHeight[=setting]
<b>Data Type</b>	Integer

## ButtonBorderStyle Property

<b>Description</b>	Selects the button's border style.
<b>Custom</b>	<a href="#">General Tab</a> using the Border Style combo box.
<b>Visual Basic</b>	[Form1.]RSButton1.ButtonBorderStyle[=setting%]
<b>Remarks</b>	Available styles are 0 = Normal; 1 - BorderStays; and 2 - IncreaseBorder.
<b>Data Type</b>	Integer

## ButtonBorderWidth Property

<b>Description</b>	Sets the width of the BorderHighlight and BorderShadow properties for the button control.
<b>Custom</b>	N/A in Custom Properties page.
<b>Visual Basic</b>	[Form1.]RSButton1.ButtonBorderWidth[=setting]
<b>Data Type</b>	Integer



ButtonStyle Property

Description	Sets the style for the RSButton. Possible styles are	
	Settings	Description
	0	Normal
	1	Toggle
	2	Horizontal LED pair
	3	Horizontal Reverse LED pair
	4	Vertical LED pair
	5	Vertical Reverse LED pair
	6	Diode
	7	Check Box
	8	Square
	9	Radio Button
	10	Horizontal Switch
	11	Horizontal Reverse Switch
	12	Vertical Switch
	13	Vertical Reverse Switch
	14	Push Pop
Custom	<a href="#">General Tab</a> using the Button Style Combo Box.	
Visual Basic	[Form1.]RSButton1.ButtonStyle[=setting]	
Data Type	Integer	

## DisplayDownCaption Property

<b>Description</b>	Enables/disables the caption when the button is in the down state.
<b>Custom</b>	Set with "Support button down caption" check box in the Options list on the <a href="#">General Tab</a> .
<b>Visual Basic</b>	[Form1.]RSButton1.DisplayDownCaption[=setting]
<b>Remarks</b>	True enables the down caption. False disables the down caption.
<b>Data Type</b>	Boolean

## DownCaption Property

**Description** Caption that is displayed when the button is in the down state.

**Custom** Set on the [Value Tab](#).

**Visual Basic** [Form1.]RSButton1.DownCaption[=setting]

**Data Type** Text

## DownCaptionColor Property

<b>Description</b>	Sets the color of the caption text when the button is in the down state.
<b>Custom</b>	<a href="#">Display Tab</a> by choosing DownCaptionColor in the properties Combo box and then selecting a color in the colors Combo box.
<b>Visual Basic</b>	[Form1.]RSButton1.DownCaptionColor[=setting]
<b>Data Type</b>	Color

## DownValue Property

<b>Description</b>	The DownValue becomes the Value of the RSButton when in the down state. If the DownValue matches the RSButton's Value then RSButton will go into the down state.
<b>Custom</b>	<a href="#">Value Tab.</a>
<b>Visual Basic</b>	[Form1.]RSButton1.DownValue[=setting]
<b>Data Type</b>	Integer

## DrawFocusBorder Property

<b>Description</b>	Toggles display of the focus border, which is shown when a button has focus.
<b>Custom</b>	Set in the Options list on the <a href="#">General Tab</a>
<b>Visual Basic</b>	[Form1.]RSButton1.DrawFocusBorder[=setting]
<b>Remarks</b>	When set to True, the Button's focus border will be displayed; when set to False, the focus border will not be displayed.
<b>Data Type</b>	Boolean

## GroupExclusive Property

<b>Description</b>	Enables/disables the exclusiveness of an RSButton within a group of RSButtons.
<b>Custom</b>	Set in the Options list on the <a href="#">General tab</a> .
<b>Visual Basic</b>	[Form1.]RSButton1.GroupExclusive[=setting]
<b>Remarks</b>	A group of RSButtons is defined as any number of RSButtons placed on top of the same object. For example, if five RSButtons are placed on top of a Panel on a form, those five RSButtons are considered a group, and will be treated as such if each of their GroupExclusive properties are set to True.
<b>Data Type</b>	Boolean

## LightNotEnabledColor Property

**Description** Sets the color of light (LED) when the button is not enabled.

**Custom** [Display Tab.](#)

**Visual Basic** [Form1.]RSButton1.LightNotEnabledColor[=setting]

**Remarks** Default color is black.

**Data Type** Color



## LightOffColor Property

**Description** Light (LED) color when the button is in the off state.

**Custom** [Display Tab](#).

**Visual Basic** [Form1.]RSButton1.LightOffColor[=setting]

**Remarks** Default color is red.

**Data Type** Color

## LightOnColor Property

<b>Description</b>	Light (LED) color when the button is in the on state.
<b>Custom</b>	<a href="#">Display Tab.</a>
<b>Visual Basic</b>	[Form1.]RSButton1.LightOnColor[=setting]
<b>Remarks</b>	Default color is green.
<b>Data Type</b>	Color

## RoundButton Property

<b>Description</b>	When True the corners of the border around the button, will appear rounded rather than squared.
<b>Custom</b>	Set in the Options list on the <a href="#">General Tab</a> .
<b>Visual Basic</b>	[Form1.]RSButton1.RoundButton[=setting]
<b>Remarks</b>	When set to True the button border's corners will be rounded; when set to False the button's border will be rectangular.
<b>Data Type</b>	Boolean

## ServerCanUpdateValue Property

<b>Description</b>	Allows the server to update the Value.
<b>Custom</b>	Check box in the general section
<b>Visual Basic</b>	[Form1.]RSButton1.ServerCanUpdateValue[=setting]
<b>Remarks</b>	True allows the server to update the value. False disables server update.
<b>Data Type</b>	Boolean

## UpValue Property

<b>Description</b>	UpValue becomes the Value of the RSButton when in the up state. If the UpValue matches the RSButton Value the RSButton will go into the up state.
<b>Custom</b>	Set on the <a href="#">Value Tab</a> using the "Up" text box.
<b>Visual Basic</b>	<code>[Form1.]RSButton1.UpValue[=setting]</code>
<b>Data Type</b>	Integer

## INI File Used for RSButton

RSButton saves some of the information such as the default files names etc. in the RSTOOLS.INI file located in the c:\windows directory. Only the following information should be changed or altered and others section should not be altered for proper operation of the OCX.

The following section sets the default template file for RSButton. The user templates are stored in this file.

[cfg]

cfg=c:\windows\mytempl.rwc

where mytempl.rwc is the default template file you want your templates to be saved in.

The symbol information for the DDE link is set in the following section.

[sym]

*symbolname=server|topic|item*

where the 'server' is the DDE server, 'topic' is the DDE topic and 'item' is the DDE item. Example of the above is

[sym]

subtotal= Excel[[Book1]Sheet1|R6C2

total= Excel[[Book1]Sheet1|R8C2

The requested information could not be found.

## Introduction

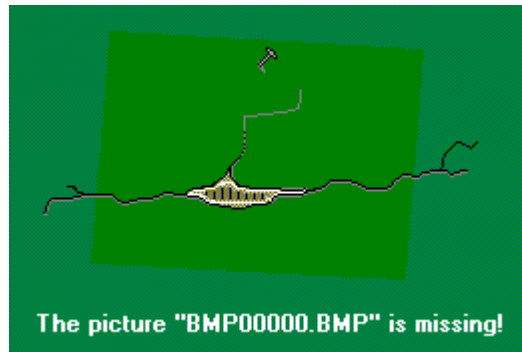
## Basic Concepts

- The RSTools controls are designed to be used with any Dynamic Data Exchange Server and provide enhanced performance when used with those that conform to AdvanceDDEprotocol. If you choose to write your own DDE server, the AdvanceDDE protocol can be obtained from Rockwell Software Inc., by contacting our Technical Support.
- The RSTools controls can be easily configured by setting their properties, without writing a single line of code.
- The RSTools controls can be used as a bound control. They automatically handle adding, updating and displaying data from the Data control, which supports ODBC, Microsoft Access• , dBASE• , Foxpro• , Paradox• , Btrieve and several other database types.
- The RSTools controls also support the Paste Link method from RSLinx or Wintelligent Linx, Excel or other DDE servers for transferring DDE Link data via the clipboard.
- The RSTools controls have built in error notification and handling. If an error has occurred in the data stream, the error can be displayed in the control's Caption property, or handled via code in one of the events.
- With RSTools you may consider using RSJunctionBox - Our flexible, high-speed DDE communications module. RSJunctionBox allows all of our controls to communicate with DDE Servers supporting AdvanceDDE and Excel Table formats.



**Description**

RSTools is a unique line of graphical OLE Custom Controls that let you turn any data into a dynamic picture. You can easily add a professional “custom” look to all interface and data management projects. RSTools are the fastest, easiest-to-use programming tools for developing applications for manufacturing and process control and they give you the power to build graphical database interfaces that tell the story in a flash.

**File Name**

RSBTN32.OCX  
RSCMP32.OCX  
RSDAT32.OCX  
RSGGE32.OCX  
RSSLD32.OCX  
RSVSL32.OCX  
RSWHL32.OCX

**Remarks**

The RSTools controls have several custom properties that allow you to monitor and control data. The RSTools controls also have custom events that allow you to be notified when data has changed, or if data has finished its request or poke.

**Note**

*When you create and distribute applications that use the RSTools controls, you should install the appropriate OCX files in the WINDOWS \SYSTEM subdirectory. A more detailed description of these files are located on page 9 of this chapter. The Setup Wizard included with Visual Basic provides tools to help you write setup programs that install your applications correctly.*

## Bound XE "Bound" Properties

The RSTools controls have three bound properties: **DataUpdate**, **DataField** and **DataSource**. This allows the RSTools controls to be linked to a Visual Basic Data control or Remote Data Control, and display field values for the current record in the recordset. The RSTools control can also write values to the Data control's recordset.

**Note**                      *For more information on using bound controls, refer to Accessing Databases With the Data Control, in the Visual Basic Programmers Guide.*

## **Properties & Events XE "Properties"**

All of the common properties, events and methods for the RSTools controls are described in chapters 5, 6 and 7 of this manual, respectively. The RSTools controls also come with many standard properties. For documentation of these properties, see "Standard Properties, Events, and Methods." in the Microsoft Visual Basic *Custom Control Reference*.

## **Installing the RSTools**

You can install the RSTools controls on your computer using Rockwell Software's SETUP.EXE. The setup program installs all RSTools files, the Help system and other product components from the distribution disks to your hard disk.

## System Requirements

Before you install RSTools, make sure that your computer meets the minimum system requirements. You must have certain hardware and software installed on your computer. The system requirements include:

### *Minimum Requirements*

- IBM-compatible 486 or higher
- 8 MB of RAM
- 10 MB of free Hard Disk space
- A 3.5" floppy drive
- VGA Graphics Card
- Microsoft Visual Basic 4.0, Microsoft Access, or other OLE Container.

Note: This manual references Visual Basic only.

- Microsoft Windows NT 3.51 or Windows 95

***Recommended Requirements***

- IBM compatible Pentium
- 16 MB of RAM or more
- 30 MB of free Hard Disk space
- CD-ROM
- Color (800x600) or (1024 x 768)
- Any Microsoft compatible pointing device (mouse, trackball, touchscreen, etc.)
- Microsoft Visual Basic 4.0, Professional Edition or the Enterprise edition
- Microsoft Windows NT 3.51 or Windows 95

## **RSTOOLS.WRI File**

The RSTOOLS.WRI file lists any last minute changes to the RSTools documentation, Help file and to the RSTools controls. To read the file, open the Windows Write application or double-click the **RSTOOLS.WRI** file using File Manager or Windows Explorer in the following directory C:\RSWKSHOP\RSTOOLBX.

## **Running Setup XE "Running Setup"**

When you run the setup program, you will create a path for RSTools.

### **To Start Setup:**

1. Insert Disk 1 in drive A.
2. From the file menu in Program Manager or File manager, choose Run.
3. Type a:setup
4. Follow the setup instructions on the screen.

The RSTools program group is created in your Program Manager or placed in your Windows 95 Start Menu. The Help file, RSTOOLBX.WRI file and sample applications are also copied to this program group.



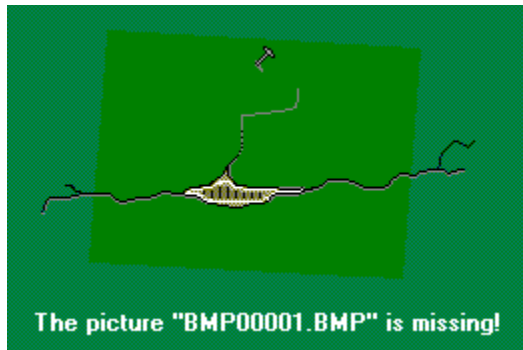
## **Learning To Use RSTools**

The RSTools controls require very little programming to create full-featured applications. Much of the functionality is available by setting standard properties. You can even log data seamlessly through the Visual Basic Data control or Remote Data control with only a few lines of code.

## Loading RTools XE "Loading RToolbox" Controls (Adding RTools to your VB Project)

To use RTools in Visual Basic, you must add the RTools controls to the Visual Basic toolbox.

1. Start Visual Basic.
2. From Visual Basic, select Tools/Custom Controls menu item.
1. Scroll down the list box until one of the "Rockwell Software" OCX tools is visible, select the desired control, and choose OK. (You may have to click the Browse button and manually search for the OCX file, which will be located in the C:\WINDOWS\SYSTEM directory.) The control is added to the project list. A list of the files are on the next page.



1. The RTools icons are added to the Visual Basic Toolbox.

## Autoloading XE "Autoloading" RSTools XE "RSToolbox"

You can configure Visual Basic to automatically load the RSTools controls when you start a new project in Visual Basic.

To configure Visual Basic to automatically load RSTools:

1. Start Visual Basic and open AUTO32LD.VBP.
1. Choose Custom Controls from the Tools menu. (The "Custom Controls" Dialog appears.)
1. Scroll down the list box until the "Rockwell Software" OCX tools are visible, select the desired controls, and choose OK. (You may have to click the Browse button and manually search for the OCX files, which will be located in the C:\WINDOWS\SYSTEM directory). The controls are added to the project list.  
Here is a list of the current OCX files:

<u>1.OCX</u>	<u>Filename</u>	<u>Description</u>
RSButton	RSBTN32.OCX	Rockwell Software Button
RSCmpare	RSCMP32.OCX	Rockwell Software Compare
RSData	RSDAT32.OCX	Rockwell Software Data
RSGauge	RSGGE32.OCX	Rockwell Software Gauge
RSSlider	RSSLD32.OCX	Rockwell Software Slider
RSVessel	RSVSL32.OCX	Rockwell Software Vessel
RSWheel	RSWHL32.OCX	Rockwell Software Wheel

-

1. Choose Save Project from the File menu.

## Using Help XE "Help"

Comprehensive on-line help is available to assist you as you learn and use the RSTools controls. The complete RSTools documentation is available through on-line help. In addition, you can receive context-sensitive help for properties and events. The Help file is located in your RSTools directory.

### To access the help contents page:

1. Click one of the RSTools icons in the Visual Basic toolbox.
2. Press F1.

### To access context-sensitive help for properties:

1. Select an RSTools control on your form.
2. Highlight a property in the properties window.
3. Press F1.

### To access context-sensitive help for events:

1. Double-Click an RSTools control on your form.
2. Pull down the procedure Combo-Box, labeled "Proc:", and select an event from the list.
3. Press F1.

## Placing an RSTools Control on a Form XE "Form"

Creating a new instance of an RSTools control and placing it on a form is as simple as point, click and drag.

1. Select an RSTools control from the Visual Basic toolbox.
2. Position the mouse on the form at the location where you want to draw the control.
3. Click and drag to draw the outline of the RSTools control on the form.
1. When you release the mouse, the new RSTools control is placed in the location you specified.

*Note*      *By double-clicking on the control within the toolbox, the control will automatically be placed in the center of the form.*

## **Distributing RTools Applications XE "Distributing RToolbox Applications"**

Please read the license agreement that was shipped with this package. You are bound by the licensing restrictions contained in that document.

## **Redistributing Files**

You can use all the files accompanying this product for development of an application. You can redistribute the run time version of the software according to the terms of the license agreement.

You will need the individual OCX file for each OCX that you use and the following table of files:

File	Description
RSTOOL32.DLL	Common code for Design & Run modes
RSCALC32.DLL *	Calculation engine for Read/Write expressions
RSJBOX32.DLL *	RSJunctionBox module
RSJBP32.DLL*	RSJunctionBox License Check DLL

*Note*                      *\* These files are optional when distributing applications. The RSCALC32.DLL is used for calculating expressions in run time. The RSJBOX32.DLL is needed only if AdvanceDDE is required during runtime. And the RSJBP32.DLL is the run time license check.*

## INI Files

RSTools supports its own INI file used mostly for testing purposes. It allows you specify which DDE protocol to use; AdvanceDDE, XLTABLE or TEXT mode. You have the options to turn these on or off; the default setting is True (on) for all modes. To create a INI file for RSTools, you must use a text editor and add the following lines:

```
[Values]
;Force Text mode DDE
Support Excel = False
Support PackDDE = False
```

Then save the file in the C:\WINDOWS directory as "WORKLIST.INI". The above example disables the XLTABLE and AdvanceDDE modes, leaving the TEXT mode as the only mode to operate in.



## Sample Applications

In addition to the documentation, the RSTools package includes sample applications that you can load into Visual Basic. These applications are useful, and are excellent tools to learn from. You may use any part of the samples in your own applications.

**Note**      *Throughout the documentation, you will find references to sample code and applications that illustrate programming techniques. Many of the files for these applications are included on your installation disks. If you installed the sample applications, you will find them in the C:\RSWKSHOP\RSTOOLBX\DEMO\ subdirectory, or the drive you choose for the installation. You will also find some code examples in the RSTools help file, that you can copy and paste into your forms and subroutines.*

## Programming Tools

RSTools provide several custom properties and standard properties. By setting these properties, you can perform a variety of tasks, such as hiding and displaying controls, setting DDE properties or binding to a database.

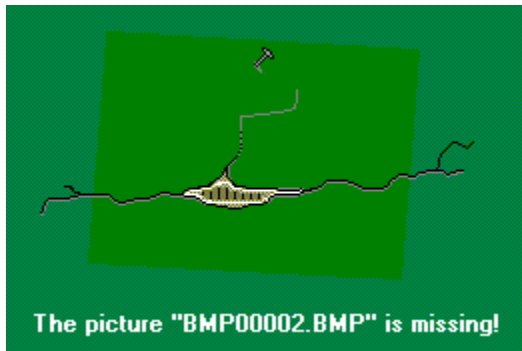
## Contents

- Link Tips
- Excel Blocks
- AdvanceDDE Arrays

## LinkTips

The LinkTips property can be a very useful tool during run time. It allows you to create a help balloon to display useful information. It will also display the property settings for the LinkServer, LinkTopic and LinkItem all at once. The controls have four properties associated with the LinkTips function:

- LinkTip (True\False) - The LinkTip property allows you to turn it on or off during run time.
- LinkTipText - Displays text in a popup window next to the control. (A help balloon).
- LinkTipBackColor (any color value) - The LinkTipBackColor property allows you to change the background color of the LinkTip window.
- LinkTipForeColor (any color value) - The LinkTipForeColor property allows you to change the text color of the LinkTip window.

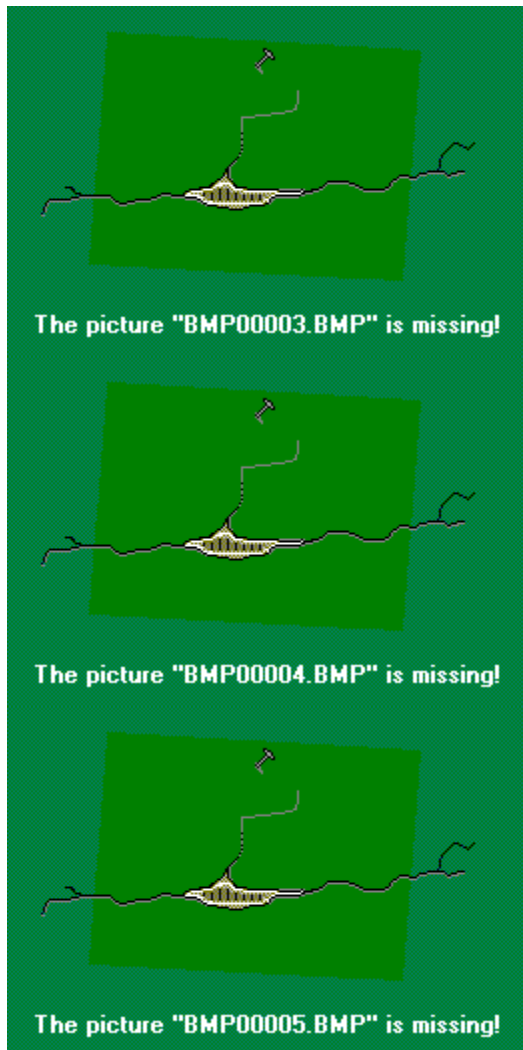


*Note*                      *Refer to the LinkTip property in the property reference for more information on the LinkTip property's attributes.*

## Excel Blocks XE "Excel Blocks"

A very useful tool that the control supports is its ability to poke or request data to Microsoft Excel, one word at a time or as a large block of data. Below is one possible example for configuring a block read from Excel (Using RSData as an example).

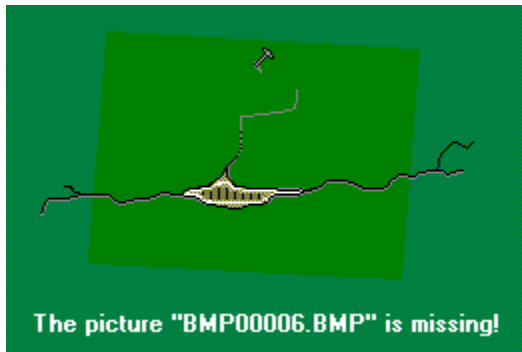
```
rsdata1.LinkServer = "Excel"  
rsdata1.LinkTopic = "[Book1]Sheet1"  
rsdata1.LinkItem = "R1C1:R2C2"    'Read 4 cells of data  
rsdata1.DoRequest = True
```



## AdvanceDDE Arrays (Optional: Requires RSJunctionBox) XE "AdvanceDDE Arrays"

AdvanceDDE arrays are very useful when accessing large amounts of data from a server. The easiest method is to use DDE block arrays as shown below.

```
rsdata1.LinkServer = "icomwdrv"  
rsdata1.LinkTopic = "testsol"  
rsdata1.Linkitem = "N7:0,L10"  
rsdata1.LinkMode = 1
```



*Note*

*This functionality (AdvanceDDE) requires a Rockwell Software serve product, RSServer Toolkit partner server product, or a self-built AdvanceDDE server as the data link source.*

### **Using DDE (Linking)**

DDE stands for “Dynamic Data Exchange”. It is a feature of the Microsoft Windows operating system that enables two applications to “talk” to each other by continuously and automatically exchanging data.

Rockwell Software has created a DDE format called AdvanceDDE. AdvanceDDE packetizes information, which greatly increases the speed at which data can be exchanged.

## Contents

- Sources and Destinations
- Applications, Topics and Items
- Linking DDE within RSTools Controls
- DDE Links
- Link Modes
- Creating Links
- Requesting and Poking Data
- DDE Arrays
- RSTools Control Arrays
- Using DDE Block Arrays with RSTools

## Sources XE "Sources" and Destinations XE "Destinations"

Two applications exchange information using a DDE conversation. This is similar to a conversation between two people. The application that initiates the conversation is called the ***destination (client)*** application, the application that responds to the client is the ***source (server)*** application. An application can be engaged in several conversations at the same time, acting as the destination in some and the source in others. The RSTools controls are designed to be used as clients only.



## Applications XE "Applications" , Topics XE "Topics" and Items XE "Items"

When a destination application begins a DDE conversation, it must specify three things:

1. The name of the source application (program), called the *application*.
2. The subject of the conversation, called the *topic*.
3. The specific data requested, called the *item*.

When a source application receives a request for a conversation concerning a topic it recognizes, it responds and a conversation is started. Once established, a conversation cannot change application or topics. The application and topic remain constant for the duration of the conversation. You must first deactivate any links to change the application or topic.

Data values are passed through *items*. Items are references to data that is meaningful to both applications. Either the destination or the source can change the value of the item without affecting the state of the conversation.

## Application name

Any application that can be a DDE source has a unique *application name*. Some examples are shown below.

<b>Application</b>	<b>DDE application name</b>
RSLinx or Wintelligent Linx	ICOMWDRV
Microsoft Excel	EXCEL
Microsoft Word	WINWORD

To set the application name for a control, you must set its **LinkServer** property to valid server name. One example for the **LinkServer** name could be “ICOMWDRV”; another would be “EXCEL”.

## Topic

The topic defines the subject of a DDE conversation. In the case of Rockwell Software’s RSLinx or Wintelligent Linx, a topic name would most likely be the name of a PLC program running somewhere on your PLC network. If you were using Excel the LinkTopic would be “[Book1]Sheet1”.

You can specify a control’s topic by setting its **LinkTopic** property.

## Item

The item defines the piece of data actually being passed during the DDE conversation. An example would be a programmable controller address from RSLinx or Wintelligent Linx such as N7:0, T4:11.ACC or B3/100, for Excel it could be a spreadsheet cell R1C1, or R1C1:R2C2

You can specify a control’s item by setting its **LinkItem** property.

## DDE Links XE "DDE Links"

A DDE conversation is often called a link because the two applications in the conversation are linked by the data they are exchanging. The RSTools controls support 3 different types of links, set by its LinkMode property:

1. **Automatic (Hot) link:** The source supplies the data to the control every time the data changes, or on a predefined interval.
2. **Manual (Cold) link:** The source supplies data only when it is requested. (See DoRequest in appendix A)
3. **Notify link:** The source notifies the control when the data changes, but supplies data only when it is requested.

### Note

*For more information on Dynamic Data Exchange (DDE), refer to Microsoft Visual Basic's Programmer's Guide.*

## **DDE Formats XE "DDE Formats"**

The RSTools controls supports three different DDE formats, these are; AdvanceDDE, XLTABLE and TEXT mode. When the control attempts to establish a link it first tries the AdvanceDDE mode; if it cannot establish a link with this mode it then tries XLTABLE and then TEXT mode.

## **RSJunctionBox**

RSJunctionBox boosts your communications by adding Rockwell Software's Advanced DDE or Microsoft's Excel Table DDE formats. These enhanced DDE formats provide RSTools with high-volume, high-speed data transfer and extended error detection. When added to your application, RSJunctionBox negotiates with each DDE sever for the best common format (AdvanceDDE, Excel Table, Text).

RSJunctionBox provides each control with notification of important communication events (Change, PokeCompleted, RequestCompleted, LinkError), and utilizes High-speed, real world data access utilizing RSLinx or other AdvanceDDE capable DDE servers. RSJunctionBox is required on each run-time node when the application is utilizing RSTools High Speed (non-Text) DDE communications functionality.

**Linking XE "Linking" DDE XE "DDE" with an RSTools Control**

## The "Best Common Format" Approach

RSTools can communicate through a variety of formats, and utilizes the "Best Common Format" approach when establishing links to source or "server" applications.

Rockwell Software's AdvanceDDE is a binary, packetized data format that allows blocks of contiguous and non-contiguous data to be encoded into a message, sent through the DDE link and decoded by the receiving application. AdvanceDDE also allows the transfer of arrays of data as single items. Many of these array items can be encoded within a single AdvanceDDE message. AdvanceDDE is used when linked to Rockwell Software's RSLinx or Wintelligent Linx family of products or any other applications utilizing this format.

XL\_Table format allows the transfer of a spreadsheet-like blocks of data in a single message. This format also supports specifying an array of cells in a single item. It is used when linked to Microsoft Excel or other applications providing XL\_Table services.

TEXT format is the most common format and is used if no other format is supported by the DDE Server.

## **How AdvanceDDE works**

AdvanceDDE compresses multiple DDE messages into a single message. In TEXT mode each control's data is sent in it's own message. If you have 10 controls using TEXT mode there can be as much as 10 times the message traffic required for AdvanceDDE.



Windows Message Flow to Exchange 5 Requested Data Items  
(TEXT Mode)

## Windows Message Flow to Exchange 10 Requested Data Items (AdvanceDDE Mode)

As can be seen, twice as much data was transferred in the second example with less than half of the messages or overhead.

## LinkModes XE "LinkModes"

The control's **LinkMode** properties are very much like Visual Basic's label control. The LinkMode property has 4 different modes:

- 0-None
- 1-Automatic
- 2-Manual
- 3-Notify

When the **LinkMode** is set to Automatic the control will attempt to initiate a conversation. If the source application specified in the **LinkServer** and **LinkTopic** properties is not running, will generate an error, during run time.

When the **LinkMode** is set to 1-Automatic, and when the data changes, the control receives the new data and a **Change** event is generated for that control. The **AllowChangeEvent** property must be set to True, if you want to use the **Change** event. If it is set to False then the event will not get generated. The **AllowChangeEvent** property can be toggled at run time to disable the event.

If you set the **LinkMode** to 2-Manual or 3-Notify, then the control will not automatically update, and you must use the **LinkRequest** method (**DoRequest** property. See Creating Manual Link).

## **Manual XE "Manual" or "Cold" Links**

Manual Links require that Visual Basic code in the application containing the control requests the data from the source or server. Manual Links are often used to poke or write data back to the server or source application. The controls are designed so that when they are used in control arrays, blocks of data can be Requested or Poked with a single line of Visual Basic code. This is the most efficient way to write data with the RSTools controls.

## **Automatic or "Hot" Links**

Automatic links are the easiest way to move information with the controls, and does not require any Visual Basic code to transfer data . The link can be made entirely by setting properties at design time.

When data in the source or server application changes, it is automatically updated in the control. Automatic Links can be set up as array items. Refer to “Using DDE Block Arrays with RSTools” later in this chapter.

## Creating an Automatic Link XE "Automatic Link"

There are three ways to create an automatic link. These are

- Paste & Set LinkMode
- Design Time - set link properties
- At run time

If you have a valid **LinkServer**, such as RSLinx or Wintelligent Linx, and a valid **LinkTopic** and **LinkItem**, then you can establish an automatic link by setting the **LinkMode** property for that control to Automatic.

The following code for a Form\_Click event attempts to establish an automatic link at run time between RSTools controls and RSLinx or Wintelligent Linx when the form is clicked.

### Paste & Set LinkMode:

1. In RSLinx or Wintelligent Linx, select Copy to Clipboard from the Edit menu.
2. Select a valid Data Table Address and Topic name.
3. Click Ok.
1. Go back to VB, select or place a RSData control (or any RSTools control) on the form. Then select Paste Link.
1. Press Run (F5).

### Design time - Set Link properties:

1. Drop an RSData (or any RSTools control) control on a form, and bring up the properties window for the control. (Or use the LinkInfo custom property tab.)
1. Set the **LinkServer** Property to "ICOMWDRV", "EXCEL" or another DDE server.
2. Set the **LinkTopic** Property to "TESTSOL", or to the name of the topic you configured.
3. Set the **LinkItem** Property to "N7:0", or any valid item name.
4. Set the **LinkMode** Property to 1 - Automatic.
5. Press Run (F5).

### Run time (Using Code):

1. Double click on a form that has a RSData or another RSTools control on it, and add the following code. (If using a control other than RSData, substitute that control's name for *RSData1*.)

```
Sub Form_Click ()  
  
    Const AUTOMATIC = 1, NONE = 0  
  
    'Set LinkMode properties  
  
    rsdata1.LinkMode = NONE  
  
    rsdata1.LinkServer = "ICOMWDRV"  
  
    rsdata1.LinkTopic = "Testsol"  
  
    rsdata1.Linkitem = "T4:0.ACC"  
  
    rsdata1.LinkMode = AUTOMATIC  
  
End Sub
```

1. Press Run (F5).

### Note

*For the above sample to work, you must be running Rockwell Software's RSLinx or Wintelligent Linx and WINtelligent EMULATE 5, running the sample PLC ladder logic program called ICOM5DM7.X5.*

## Creating a Manual Link XE "Manual Link"

There are three ways to create an manual link. Two don't even require you to write any VB code. These are as follows: 1) Paste & Set **LinkMode**, 2) Design Time - set link properties, and 3) At run time.

To create a manual link, you must set the **LinkMode** property to 2-Manual. When **LinkMode** is manual, a conversation will exist, but you must update the link by using the **LinkRequest** method (or **DoRequest** property). By setting the **LinkRequest** method, you will update the control.

The properties **DoPoke** and **DoRequest** are identical to LinkPoke and LinkRequest. They are here strictly for backwards compatibility with the RSData VBX. Use the LinkPoke and LinkRequest methods, not the DoPoke or DoRequest properties. When using LinkPoke and LinkRequest the UseInPoke/UseInRequest properties should be checked.

The following code for a **Form\_Click** event will establish a manual link at run time between an RSData control and Rockwell Software's RSLinx or Wintelligent Linx server when the form is clicked. Each time the form is clicked a request will be made:

```

Sub Form_Click()
    Const MANUAL=2, NONE=0
    'Establish Link with the LinkServer
    If rsdata1.LinkMode = NONE then
        rsdata1.LinkServer = "ICOMWDRV"
        rsdata1.LinkTopic = "TESTSOL"
        rsdata1.LinkItem = "N7:21"
        rsdata1.LinkMode = MANUAL
    End If
    'Retrieve data
    rsdata1.LinkRequest
End Sub

```

**Note**      *For the above sample to work, you must be running Rockwell Software's RSLinx or Wintelligent Linx and WINTelligent EMULATE 5, running the sample PLC ladder logic program called ICOM5DM7.X5.*



## Efficient Requests XE "Requests"

The properties affecting requests are **RequestLength**, **RequestMode**, **RequestStartIndex** and **LinkRequest**. When creating an array of controls, the **LinkRequest** property comes in handy for requesting information into the array from the appropriate **LinkItems**. These properties allow the user to read a block of data by setting a **RequestStartIndex** and a **RequestLength**, and initiating a **LinkRequest**. For example, let's say you want to request information into 10 RSDData (or any RSTools control) controls on a form from a DDE server. Instead of requesting each one separately, set the **RequestLength** to 10 and set the **LinkRequest** property. Once the **LinkRequest** is initiated, the request will start from the RSDData control that was specified in the **RequestStartIndex**.

The following example uses controls RSDData1(0) through RSDData1(9):

```
Sub Form_Click ()  
    rsdata1(0).RequestStartIndex = 0  
    rsdata1(0).RequestLength = 10  
    'Request data  
    rsdata1(0).LinkRequest  
End Sub
```

You can also set which one of the RSDData controls get updated in the array request, by setting the **UseInRequest** property. If any one of the control's **UseInRequest** property is set to false, then that control will not get updated.

*Note                      The above example references the RSDData control, any one of the RSTools controls may be used in the same manner.*

## Efficient Pokes XE "Pokes"

The properties affecting pokes are **PokeLength**, **PokeStartIndex** and **LinkPoke**. When creating an array of controls, the **LinkPoke** property comes in handy for Poking information in the array to the appropriate **LinkItems**. These properties allow the user to set up a block of data by setting a **PokeStartIndex** and a **PokeLength**, and initiating a **LinkPoke**. For example, let's say you want to poke the information in 10 RSDData (or any RSTools control) controls on a form to another application. Instead of poking each one separately, set the **PokeLength** to 10 and set the **LinkPoke** property. Once the **LinkPoke** is initiated, the poke will start from the RSDData control that was specified in the **PokeStartIndex**.

The following example uses controls RSDData1(0) through RSDData1(9):

```
Sub Form_Click ()  
    rsdata1(0).PokeStartIndex = 0  
    rsdata1(0).PokeLength = 10  
    'Poke data  
    rsdata1(0).LinkPoke  
End Sub
```

As with the **UseInRequest** property, you can also set which one of the RSDData controls in the array are included in the DoPoke, by setting the **UseInPoke** property. If any one of the control's **UseInPoke** property is set to false, then that control will not get poked.

*Note*                      *The above example references the RSDData control, any one of the RSTools controls may be used in the same manner.*

## Creating a Notify Link XE "Notify Link"

The Notify Link is 3-Notify, it is similar to 2-Manual except that the source application notifies your application when the data changes. The notification occurs in your application through the **LinkNotify** event.

The sample below will wait until a **LinkNotify** event has occurred, after you click the form, and will then execute a **DoRequest**:

```
Sub Form_Click()  
    Const NOTIFY = 3, NONE = 0  
    'Establish Link with the LinkServer  
    If rsdata1.LinkMode = NONE then  
        rsdata1.LinkServer = "ICOMWDRV"  
        rsdata1.LinkTopic = "TESTSOL"  
        rsdata1.LinkItem = "T4:0.ACC"  
        rsdata1.LinkMode = NOTIFY  
    End if  
End Sub  
  
Sub RSData1_LinkNotify()  
    'Retrieve Data  
    rsdata1.LinkRequest  
    rsdata1.LinkMode = None  
End Sub
```

*Note*

*For more information on the common types of LinkModes and properties, refer to the Visual Basic Programmers Guide on DDE.*

## Working with DDE Arrays

One of the most important functions of the RSTools controls is optimization of DDE communications. Rich array manipulation activities are supported to accomplish this optimization. There are three important concepts to understand:

- Control arrays are the most efficient way to move “non-contiguous” data as all writes or reads to a topic are completed simultaneously in XL\_Table and AdvanceDDE messages.
- Array data types are the most efficient way to move “contiguous” data to/from servers that support them. Entire arrays are moved in one item.
- For highest data density per DDE message, array data types can be used in control arrays.

## RSTools Control Arrays XE "Control Arrays"

An efficient method for requesting or poking data is to use an array of RSData (or any RSTools) controls. For example, if you want to request 20 items of data from a particular server and topic. You could set up a control array on your form in run time, then set the **LinkRequest** method, which will then retrieve all 20 items from that server and topic. If, for instance, you wanted to poke data rather than request data, then all you would have to do is set the **LinkPoke** method to True, instead of the **LinkRequest** property. Try the following example below.

1. Place an RSData control on a blank form.
2. Set the **Index** = 0.
1. Set the **RequestLength** = 20, **RequestMode** = 0 - Start with Data, **RequestStartIndex** = 0 and **UseInRequest** = True.
1. Type in the following code below.

### Sub Form\_Load ()

```
const manual = 2
' set properties for server, topic and item
rsdata1(0).LinkServer = "icomwdrv"
rsdata1(0).LinkTopic = "testsol"
rsdata1(0).LinkItem = "n12:0"
For j% = 1 To 19
    ' load 19 more rsdata control on the form
    Load rsdata1(j%)
    rsdata1(j%).LinkItem = "N12:" & LTrim$(Str$(j%))
    Select Case j%
```

*Continued on next page*

```
Case 10
    rsdata1(j%).Left = rsdata1(j% - 10).Left +
    rsdata1(j% - 10).Width
    rsdata1(j%).Top = rsdata1(0).Top
Case Else
    rsdata1(j%).Left = rsdata1(j% - 1).Left
    rsdata1(j%).Top = rsdata1(j% - 1).Top +
rsdata1(j% - 1).Height
End Select
rsdata1(j%).Visible = True
Next j%
For j% = 0 To 19
    rsdata1(j%).LinkMode = Manual
Next j%
'request the data
```

```
rsdata1(0).LinkRequest
```

**End Sub**

1. After typing in the above code, run the Visual Basic program. It should create 19 more RSData controls on the form. And then fill them with data via the LinkRequest method.

*Note*

*For this example to work properly you must have RSLinx or Wintelligent Linx and WINtelligent EMULATE 5 running with the sample ladder logic program ICOM5DM7.X5.*

The previous example can be modified slightly to do a control array poke. The example below will poke data to the same **LinkItem's** as the previous example. We must first populate each of the controls in the array with data. To do this, create a For/Next loop to fill in random sample data.

1. Place a Button control on the form and set the caption = "Poke Data".
2. Place an RSData (or any RSTools) control on a blank form.
3. Set the **Index** = 0.
4. Set **AlwaysDisplayData** = True
5. Set the **PokeLength** = 20, **PokeStartIndex** = 0 and **UseInPoke** = True.
6. Type in the following code below for each appropriate event.

**Sub Form\_Load ()**

```
Const manual = 2
' set properties for server, topic and item
rsdata1(0).LinkServer = "icomwdrv"
rsdata1(0).LinkTopic = "testsol"
rsdata1(0).LinkItem = "n7:0"
For j% = 1 To 19
    ' load 19 more rsdata control on the form
    Load rsdata1(j%)
    rsdata1(j%).LinkItem = "N7:" & LTrim$(Str$(j%))
    Select Case j%
        Case 10
            rsdata1(j%).Left = rsdata1(j% - 10).Left +
rsdata1(j% - 10).Width
    Continued on next page
```



```

        rsdata1(j%).Top = rsdata1(0).Top
    Case Else
        rsdata1(j%).Left = rsdata1(j% - 1).Left
        rsdata1(j%).Top = rsdata1(j% - 1).Top +
rsdata1(j% - 1).Height
    End Select
    rsdata1(j%).Visible = True
Next j%
End Sub

Sub Command1_Click ()
    'Set the LinkMode to manual for each control
    For j% = 0 To 19
        rsdata1(j%).LinkMode = manual
    Next j%
    'Generate random data for RSData captions
    For j% = 0 To 19
        rsdata1(j%).Caption = Int(Rnd(j% + 1) * 1000)
    Next j%
    'Poke the data
    rsdata1(0).LinkPoke
End Sub

```

Run the program, but before you click the “Poke Data” button, switch to RSLinx or Wintelligent Linx and view the N7:0 data table. This allows you to view the changes in the data table when you click the “Poke Data” button. When both the Visual Basic window and the RSLinx or Wintelligent Linx windows are up, click the “Poke Data” button.

*Note For this example to work properly you must have RSLinx or Wintelligent Linx and WINtelligent EMULATE 5 running with the sample ladder logic program ICOM5DM7.X5.*

## Using DDE Block Arrays XE "DDE Block Arrays" with RSTools

Using DDE block arrays is a very efficient method of retrieving data from a server when using a single control. Each piece of data can then be retrieved out of the DDE block array using the **DataValue(index)** run time-only property. The **NumberOfDataValues** property, allows access to the number of elements in an array item.

Below is an example that retrieves 30 words (Pieces) of data from RSLinx or Wintelligent Linx. Then displays several of the word values in bargraphs and a label that displays how many words are in the Link itself.

1. Place one-RSData control, two-Label controls and two-3DPanel controls on the form.
2. For the RSData control, set the **AllowChangeEvent** = True and **LinkTip**=True.
3. For Label1, set the caption = "Number of Data Values".
1. For Panel3D1, set the FloodColor = Green (&HC000&), FloodShowPct = True and FloodType = 1 - Left To Right.
2. For Panel3D2, set the FloodColor = Red (&HFF&), FloodShowPct = True and FloodType = 1 - Left To Right.
1. Type in the following code.

### Sub Form\_Load ()

```
Const Automatic = 1

' Set properties for the server, topic and item
rsdata1.LinkServer = "icomwdrv"
rsdata1.LinkTopic = "testsol"

' Set LinkItem to T4:0.ACC with a length of 30 words
Continued on next page
rsdata1.LinkItem = "T4:0.acc,L30"

rsdata1.LinkMode = Automatic
```

### End Sub

### Sub RSData1\_Change (NewData As String)

```
' set the floodpercent properties using the DataValue
' If the value of DataValue is greater than 100, then
Divide by 10

panel3d1.FloodPercent = rsdata1.DataValue(0) / 10
panel3d2.FloodPercent = rsdata1.DataValue(3) / 10

' Set label2's caption to the Number of Data
values(i.e. 30)
label2.Caption = rsdata1.NumberOfDataValues
```

### End Sub

1. After typing in the code, run the program.

Notice that as you move the mouse cursor over the RSData control and leave it there for more than one second, the **LinkTip** box is displayed. You should see something similar to this:

```
icomwdrv!testsol!t4:0.acc,l30
```

The **L30** signifies a length of 30 words. You can also change the BackColor and ForeColor of the **LinkTip** box, by setting the **LinkTipBackColor** and **LinkTipForeColor** properties.

*Note*

*For this example to work properly you must have RSLinx or Wintelligent Linx or Wintelligent Linx and WINtelligent EMULATE 5 running with the sample ladder logic program ICOM5DM7.X5.*

### **Binding to the Data Control**

With the RSTools controls and Visual Basic's Data control, you can create an application to display edit, and update (log) information from many types of existing databases. Creating a data-aware application with Visual Basic can be done easily through a few steps, and requires very little code.

The first thing you need to do to make a "data-aware" application is to add the Visual Basic Data control or the Remote Data control to your form. Next, you will have to specify the database you would like to get the information from. Once you have decided on the database, you must place the RSTools controls on the form and set their properties to "bind" to Visual Basic's Data control. Depending on the property settings you choose for DataUpdate, when you run the application you will be able to view data coming from your server, view data from a database, or log data to the database.

The RSTools controls combined with Visual Basic's Data control give you seamless access to many standard databases, including Microsoft Access, Btrieve, dBASE, Microsoft FoxPro, and Paradox. If the Remote Data control is used, ODBC databases such as SQL Server and Oracle are accessible anywhere on a network.

## Contents

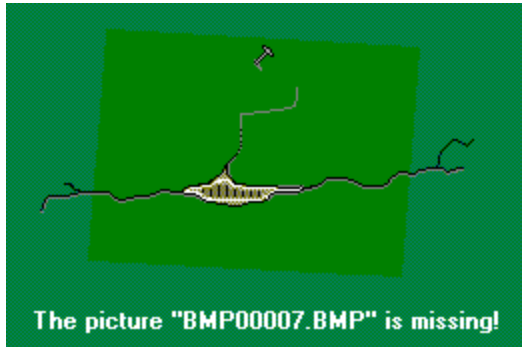
- Quick Start - Using RSTools to read from DataSource.
- Using RSTools to log to DataSource.
- Time-Based Logging• .
- Working with other types of Databases

## Quick Start

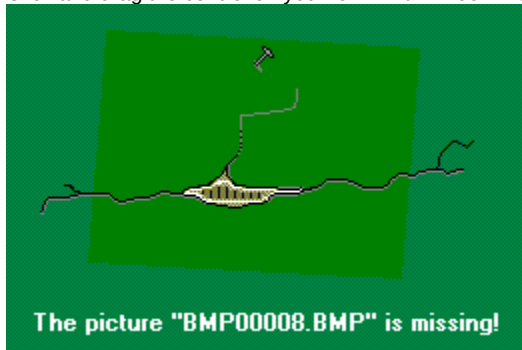
The following procedure gives you a brief overview of how to “bind” or use the RSData control (one of the RSTools controls) and Visual Basic’s data control in your application. Below we will use the COLOR.MDB sample database that comes with RSTools.

### How to use the RSData (or any RSTools control) control as a “Database” tool.

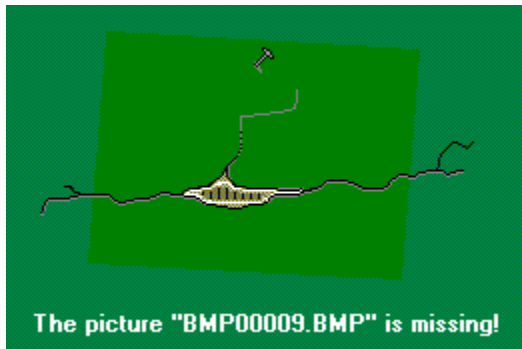
1. Select the RSData control in the toolbox and draw a RSData control on the form. The RSData control icon looks like this:



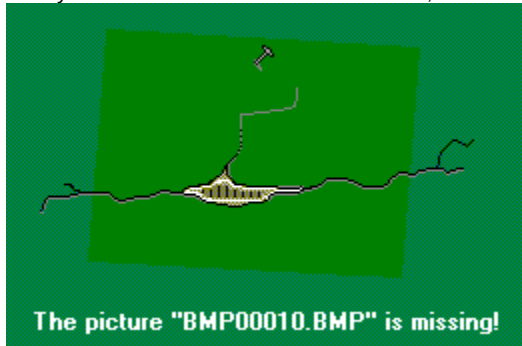
Click and drag the control on your form. It will look like a label control. The default name of the control is RSData1.



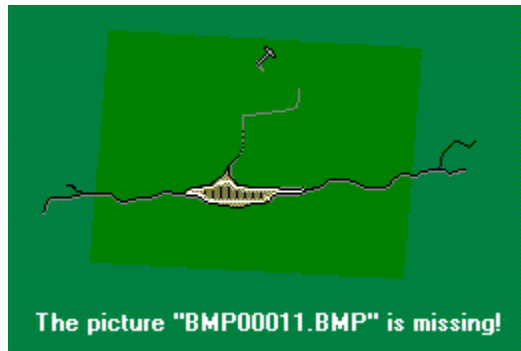
1. Select the Visual Basic Data control and draw a control on the form. The Data control icon looks like this:



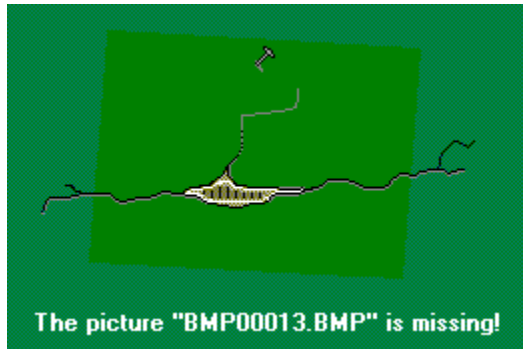
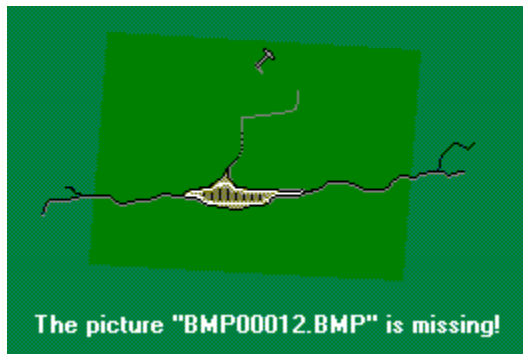
After you have drawn the control on the form, it will look like this, with the default caption as Data1.



1. In the properties window for the Visual Basic Data control, set the DatabaseName property to the filename you want to connect to C:\RSWKSHOP\IRSTOOLBX\DEMO\COLOR.DBF.



1. For the **Connect** property, you must specify "Access" Then set **RecordSource** = "COLORS". The RecordSource is the name of the database table.
1. Click once on the RSData control and bring up it's property window. In the properties window, set the **DataSource** = Data1, the **DataField** = COLORNAME, and set the **DataUpdate** = 2- Data Source.



1. Now run the application. The RSTools control displays the data in the "COLOR" field based on which database record you display. You can use the arrow buttons in the data control to move through the records.

## Working with other types of Databases XE "Databases"

The RSTools controls support all database types that Visual Basic's Data control and Remote Data control supports. These include: Microsoft Access, dBASE, Paradox, FoxPro, and many other ODBC compliant drivers.

If you used any database other than Access, in earlier versions of Microsoft Visual Basic, you would have had to create an initialization file (\*.INI) for your application. This INI file would need to contain ISAM information for the database type you have chosen, and would have the same name as your Visual Basic EXE name. The INI file would be placed in your C:\WINDOWS directory. With Visual Basic 4.0, you no longer have to be concerned with creating your own INI files for the ISAM drivers; Visual Basic will take care of connecting to these databases.

***Note** Refer to the Visual Basic Programmers Guide for more information on other types of databases.*



## Property Reference

# AllowChangeEvent Property

**Applies To** RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel

**Description** Enables the firing of a Change event when data in the control changes.

**Visual Basic** *object.AllowChangeEvent*[ = *setting* %]

**Settings** The following table lists the AllowChangeEvent property settings for the control.

Settings	Description
False	Do not allow the Change event to occur.
True	Allow the Change event to occur.

**Data Type** Integer

# AutoColumns Property

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Creates the correct number of columns to be displayed based on the LinkItem.
<b>Custom</b>	No access via custom property page.
<b>Visual Basic</b>	<i>object</i> .AutoColumns[= <i>setting</i> %]
<b>Remarks</b>	AutoColumns will work only if the LinkItem length divided by the number of columns leaves a remainder of zero. For example, a LinkItem of "C5:0,L11,C2" would display as only one column because eleven divided by two leaves a remainder of one. AutoColumns will only work with AdvanceDDE, which requires RSJunctionBox.
<b>Data Type</b>	Integer

# BackColor Property

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Sets the background color for the control.
<b>Custom</b>	Display section.
<b>Visual Basic</b>	<i>object</i> .BackColor[= <i>setting</i> %]
<b>Settings</b>	Visual Basic uses the Microsoft Windows operating environment red-green-blue (RGB) color scheme. The settings for color are:

<u>Setting</u>	<u>Description</u>
Normal RGB colors	Colors specified by using the Color palette or by using the RGB or QBColor functions in code.
System default colors	Colors specified by system color constants listed in the Visual Basic (VB) object library in the Object Browser. The Windows operating environment substitutes the user's choices as specified in the Control Panel settings.

<b>Data Type</b>	Color
------------------	-------

# BackStyle Property

Applies To	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel	
Description	Determines if the control will be transparent or opaque.	
Custom	No access via custom property page.	
Visual Basic	<i>object</i> .BackStyle[= <i>setting</i> %]	
Settings	<u>Settings</u>	<u>Description</u>
	0	Sets the control to transparent.
	1	Sets the control to opaque.
Data Type	Integer	

# BevelHeight Property

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Sets the height of the top and bottom beveled edges around the control.
<b>Custom</b>	No access via custom property page.
<b>Visual Basic</b>	<i>object</i> . <b>BevelHeight</b> [= <i>setting</i> %]
<b>Data Type</b>	Integer

# BevelHighlight Property

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Sets the highlight color of the control's beveled border.
<b>Custom</b>	Display section.
<b>Visual Basic</b>	<i>object</i> . <b>BevelHighlight</b> [= <i>setting</i> %]
<b>Remarks</b>	BevelHighlight is dependent on the bevel style chosen for the control. If BevelStyle is set to either 1 (indented) or 3 (beveled), the control's outside border will be given a 3 dimensional appearance by displaying a bevel highlight and a bevel shadow.
<b>Data Type</b>	Color

# BevelShadow Property

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Sets the shadow color of the control's beveled border.
<b>Custom</b>	Display section.
<b>Visual Basic</b>	<i>object</i> . <b>BevelShadow</b> [= <i>setting %</i> ]
<b>Remarks</b>	BevelShadow is dependent on the bevel style chosen for the control. If BevelStyle is set to either 1 (indented) or 3 (beveled), the control's outside border will be given a 3 dimensional appearance by displaying a bevel highlight and a bevel shadow.
<b>Data Type</b>	Color



# BevelStyle Property

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel														
<b>Description</b>	Determines the style of the beveled area around the control object. A beveled appearance is one that has a three dimensional look. This property controls the beveled appearance of the outside border around the control.														
<b>Custom</b>	Adjusted in the Display section of the custom properties page. The BevelStyle list box in the lower right hand corner has a drop drop down menu with six styles available for the bevel: <table><tr><th>Settings</th><th>Description</th></tr><tr><td>0</td><td>None</td></tr><tr><td>1</td><td>Indented</td></tr><tr><td>2</td><td>Marble</td></tr><tr><td>3</td><td>Beveled</td></tr><tr><td>4</td><td>Thick</td></tr><tr><td>5</td><td>Stripe</td></tr></table>	Settings	Description	0	None	1	Indented	2	Marble	3	Beveled	4	Thick	5	Stripe
Settings	Description														
0	None														
1	Indented														
2	Marble														
3	Beveled														
4	Thick														
5	Stripe														
<b>Visual Basic</b>	<i>object</i> .BevelStyle[= <i>setting %</i> ]														
<b>Remarks</b>	If a bevel does not appear when this property is set to True, make sure that the BevelWidth and BevelHeight properties are set to a value greater than zero.														
<b>Data Type</b>	Integer														

# BevelWidth Property

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Sets the width of the left and right beveled edges around the control.
<b>Custom</b>	No access via custom property page.
<b>Visual Basic</b>	<i>object</i> . <b>BevelWidth</b> [= <i>setting %</i> ]
<b>Data Type</b>	Integer

# BorderBeveled Property

<b>Applies To</b>	RSCompare, RSVessel
<b>Description</b>	Toggles display of BorderHighlight and BorderShadow.
<b>Custom</b>	Adjusted in the General section of the custom properties page with a checkbox in the Options window.
<b>Visual Basic</b>	<i>object</i> . <b>BorderBeveled</b> [= <i>setting %</i> ]
<b>Remarks</b>	When set to False in the RSVessel control, the BorderHighlight and BorderShadow are not displayed, but PolyBorderColor is displayed. When True, all three are shown. When set to False in the RSCompare control, BorderHighlight and BorderShadow are not displayed, while DownBorderColor, EqualBorderColor, and UpBorderColor are still shown. When set to True, all are shown.
<b>Data Type</b>	Integer

# BorderColor Property

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Determines the color of the border around the control object.
<b>Custom</b>	Display section.
<b>Visual Basic</b>	<i>object</i> . <b>BorderColor</b> [= <i>setting</i> %]
<b>Remarks</b>	The BorderColor property can be set to any color in the palette or to a hex integer value that represents a color.
<b>Data Type</b>	Color

# BorderHighlight Property

<b>Applies To</b>	RSCompare, RSVessel
<b>Description</b>	Sets the Highlight color for the three-dimensional border of the applicable controls' graphic shapes.
<b>Custom</b>	Display section.
<b>Visual Basic</b>	<i>object</i> . <b>BorderHighlight</b> [= <i>setting %</i> ]
<b>Remarks</b>	For the RSCompare control, this property sets the BorderHighlight color for all of its states - Up, Equal, and Down.
<b>Data Type</b>	Color

# BorderInner Property

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Determines if the inner border is displayed around the control object.
<b>Custom</b>	No access via custom property page.
<b>Visual Basic</b>	<i>object</i> . <b>BorderInner</b> [ = <i>setting</i> %]
<b>Remarks</b>	This property is dependent upon the BevelStyle property in that the inner border is displayed only when a BevelStyle other than 0 (none) is chosen.
<b>Data Type</b>	Integer

# BorderInnerColor Property

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Determines the color of the inner border around the control object.
<b>Custom</b>	Display section.
<b>Visual Basic</b>	<i>object</i> . <b>BorderInnerColor</b> [= <i>setting %</i> ]
<b>Remarks</b>	The BorderInnerColor property can be set to any color in the palette or to a hex integer value that represents a color.
<b>Data Type</b>	Color

# BorderShadow Property

<b>Applies To</b>	RSCompare, RSVessel
<b>Description</b>	Sets the Shadow color for the three-dimensional border of the applicable controls' graphic shapes.
<b>Custom</b>	Display section.
<b>Visual Basic</b>	<i>object</i> . <b>BorderShadow</b> [= <i>setting</i> %]
<b>Remarks</b>	For the RSCompare control, this property sets the BorderShadow color for all of its states - Up, Equal, and Down.
<b>Data Type</b>	Color



# BorderStyle Property

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Determines the type of border displayed around the control object.
<b>Custom</b>	No access via custom property page.
<b>Visual Basic</b>	<i>object</i> . <b>BorderStyle</b> [=setting %]
<b>Remarks</b>	There are two border styles available: 0 = None and 1 = Fixed Single.
<b>Data Type</b>	Integer

# BorderWidth Property

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Determines the width of the border displayed around the control object.
<b>Custom</b>	No access via custom property page.
<b>Visual Basic</b>	<i>object</i> . <b>BorderWidth</b> [= <i>setting</i> %]
<b>Remarks</b>	The BorderStyle property needs to be set to 1 (Fixed Single) in order to make changes in BorderWidth visible.
<b>Data Type</b>	Integer

# BottomBorder Property

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Sets the distance between bottom external border of the control and bottom of the control.
<b>Custom</b>	Adjusted in the General section of the custom properties page. Placing the mouse pointer over either of the control graphic's lower corners will cause a sizing pointer to appear. Clicking and holding the left mouse button allows the bottom border for the control to be adjusted to the appropriate position with respect to the bottom external border of the control. Releasing the mouse button will then set the BottomBorder property.
<b>Visual Basic</b>	<i>object</i> . <b>BottomBorder</b> [= <i>setting %</i> ]
<b>Remarks</b>	The BottomBorder property controls the distance between the outside border of the control and the lower edge of the control. Use this property to adjust the space available for displaying a caption or value.
<b>Data Type</b>	Integer

# ButtonBorderWidth Property

<b>Applies To</b>	RSCompare, RSVessel
<b>Description</b>	Sets the width of the BorderHighlight and BorderShadow properties for applicable controls.
<b>Custom</b>	No access via custom property page.
<b>Visual Basic</b>	<i>object</i> . <b>ButtonBorderWidth</b> [= <i>setting</i> %]
<b>Remarks</b>	Sets the width of the three-dimensional border (shown with the BorderHighlight and BorderShadow properties) around the RSVessel and RSCompare controls' graphic shapes. For these two controls there is not a corresponding ButtonBorderHeight property as in the RSButton control.
<b>Data Type</b>	Integer

# ButtonFaceColor Property

<b>Applies To</b>	RSButton, RSSlider, RSWheel
<b>Description</b>	Sets the color for the face of the control's buttons.
<b>Custom</b>	Display section.
<b>Visual Basic</b>	<i>object</i> . <b>ButtonFaceColor</b> [= <i>setting</i> %]
<b>Remarks</b>	For the RSButton control, the Button's (all styles) face color is set with ButtonFaceColor, for the RSSlider control, the face color of the its two scroll buttons is set with ButtonFaceColor, and for the RSWheel control ButtonFaceColor sets the face color for the up and down scroll arrow buttons.
<b>Data Type</b>	Color

# ButtonHighlight Property

<b>Applies To</b>	RSButton, RSSlider, RSWheel
<b>Description</b>	Sets the highlight color for the control's three-dimensional button(s) effect.
<b>Custom</b>	Display section.
<b>Visual Basic</b>	<i>object</i> . <b>ButtonHighlight</b> [= <i>setting</i> %]
<b>Remarks</b>	ButtonHighlight sets the highlight color for the Button control's three-dimensional outside border, sets the highlight color on the scroll buttons for the Slider, and sets the highlight color for the up and down scroll arrow buttons for the Wheel.
<b>Data Type</b>	Color

# ButtonShadow Property

<b>Applies To</b>	RSButton, RSSlider, RSWheel
<b>Description</b>	Sets the shadow color for the control's three-dimensional button(s) effect.
<b>Custom</b>	Display section.
<b>Visual Basic</b>	<i>object</i> . <b>ButtonShadow</b> [= <i>setting</i> %]
<b>Remarks</b>	ButtonShadow sets the shadow color for the Button control's three-dimensional outside border, sets the shadow color on the scroll buttons for the Slider, and sets the shadow color for the up and down scroll arrow buttons for the Wheel.
<b>Data Type</b>	Color

# Caption Property

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Sets the caption to be displayed on the control.
<b>Custom</b>	Set in the Value section of the custom properties page with the "Caption" text box. Text entered in this text box will appear as the control's caption.
<b>Visual Basic</b>	<i>object.Caption</i> [= <i>setting %</i> ]
<b>Remarks</b>	The DisplayCaption property must be set to True in order for the Caption to be shown.
<b>Data Type</b>	String



# CaptionBackColor Property

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Sets the background color of the caption for the control.
<b>Custom</b>	Display section.
<b>Visual Basic</b>	<i>object</i> .CaptionBackColor[= <i>setting</i> %]
<b>Remarks</b>	The CaptionBackColor property can be set to any color in the palette or to a hex integer value representing a color. If the CaptionTransparent property is set to True, CaptionBackColor will not be shown.
<b>Data Type</b>	Color

# CaptionColor Property

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Sets the color of the caption text for the control.
<b>Custom</b>	Display section.
<b>Visual Basic</b>	<i>object</i> .CaptionColor[= <i>setting %</i> ]
<b>Remarks</b>	The CaptionColor property can be set to any color in the palette or to a hex integer value representing a color. The DisplayCaption property must be set to True in order for this property to have any effect.
<b>Data Type</b>	Color

# CaptionShadow Property

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Enables/disables the shadow behind the caption for the control.
<b>Custom</b>	Adjusted in the General section of the custom properties page with a checkbox in the Options window.
<b>Visual Basic</b>	<i>object</i> .CaptionShadow [=setting %]
<b>Remarks</b>	The DisplayCaption property must be True before this property will have any visible effect. When set to True, a shadow will be displayed behind the caption text; when set to False, the shadow will not be displayed.
<b>Data Type</b>	Integer

# CaptionShadowColor Property

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Sets the color of the caption's shadow for the control.
<b>Custom</b>	Display section.
<b>Visual Basic</b>	<i>object</i> .CaptionShadowColor[= <i>setting %</i> ]
<b>Remarks</b>	The CaptionShadowColor property can be set to any color in the palette or to a hex integer value representing a color. The DisplayCaption and the CaptionShadow properties must be set to True before this property will have any visible effect.
<b>Data Type</b>	Color

# CaptionTransparent Property

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Enables/disables display of the CaptionBackColor.
<b>Custom:</b>	Adjusted in the General section of the custom properties page with a checkbox in the Options window.
<b>Visual Basic</b>	<i>object.CaptionTransparent[=setting %]</i>
<b>Remarks</b>	When set to True, the CaptionBackColor is transparent and will not be displayed; when set to False the CaptionBackColor will be displayed. DisplayCaption must be set to True before this property will have any visible effect.
<b>Data Type</b>	Integer

# CaptionX Property

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Sets the horizontal position of the caption with respect to the left edge of the control object.
<b>Custom</b>	Adjusted in the General section of the custom properties page. The red square caption position indicator on the control graphic shows the relative position of the caption. To change the caption position, click, drag, and drop the red square in the desired location. If the red square is not displayed, the DisplayCaption property needs to be set to True.
<b>Visual Basic</b>	<i>object</i> .CaptionX [=setting %]
<b>Remarks</b>	The range for the CaptionX property is from 0 to 100 with zero being the left edge and 100 being the right edge of the control. The DisplayCaption property must be set to True before this property will have any visible effect.
<b>Data Type</b>	Integer

# CaptionY Property

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Sets the vertical position of the caption with respect to the top edge of the control object.
<b>Custom</b>	Adjusted in the General section of the custom properties page. The red, square caption position indicator on the control graphic shows the relative position of the caption. To change the caption position, click, drag, and drop the red square in the desired location. If the red square is not displayed, the DisplayCaption property needs to be set to True.
<b>Visual Basic</b>	<i>object.Control.CaptionY</i> [= <i>setting %</i> ]
<b>Remarks</b>	The range for the CaptionY property is from 0 to 100 with zero being the top edge and 100 being the bottom edge of the control. The DisplayCaption property must be set to True before this property will have any visible effect.
<b>Data Type</b>	Integer

# Clip Property

Applies To	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel						
Description	Returns or sets the contents of the data elements in a control. Not available at design time.						
Visual Basic	<i>object.Clip</i> [= <i>string</i> ]  The Clip property syntax has these parts: <table><tr><th>Part</th><th>Description</th></tr><tr><td>object</td><td>An object expression that evaluates to an object in the Applies To list.</td></tr><tr><td>string</td><td>A string expression containing the element contents.</td></tr></table>	Part	Description	object	An object expression that evaluates to an object in the Applies To list.	string	A string expression containing the element contents.
Part	Description						
object	An object expression that evaluates to an object in the Applies To list.						
string	A string expression containing the element contents.						
Remarks	<p>The string can contain the contents of multiple rows and columns. In the string, a tab character (ANSI character 9) indicates a new element in a row, and a carriage return (ANSI character 13) indicates the beginning of a new row. Use the Chr function to embed these characters in strings. For example, the following line of code puts text into 4 Gauge controls that are displayed in 2 rows by 2 columns:</p> <pre>RSGauge1.Clip = "231" &amp; Chr(9) &amp; "400" &amp; Chr(13) &amp; "278" &amp; Chr(9) &amp; "58"</pre> <p>Compiling a string and setting a control's <b>Clip</b> property is an excellent way to programmatically use a single instance of a control to display multiple values.</p> <p>The clip property works in a similar fashion for the other RSTool controls.</p>						
Data Type	String						



# DataChanged Property

<b>Description</b>	Returns or sets a value indicating that RSData in a control has changed by some process other than by retrieving RSData from the current record.
<b>Visual Basic</b>	<i>object.Control.RSDataChanged</i> [= <i>setting</i> ]
<b>Remarks</b>	The Visual Basic Data control will record changes made to a bound database as you move through its records. In order to avoid recording changes made to the database's records, set the DataChanged property to False in the Data control's Validate event. When the Data control is moved to the next record, the Validate event is fired and if DataChanged is True, changes made to the database are recorded.
<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSWheel, RSVessel

# DataField Property

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Binds the control to a particular field in a database. Used for reading from and writing to a database. The DataSource property must be set prior to this property to enable browsing.
<b>Custom</b>	No access via custom property page.
<b>Visual Basic</b>	<i>object</i> . <b>DataField</b> [= <i>setting %</i> ]
<b>Remarks</b>	The DataField, DataSource, and DataUpdate properties work together with the Visual Basic Data control to bind the RSTools control to a database.
<b>Data Type</b>	String

# DataSource Property

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Binds the control to the particular Visual Basic data control which is bound directly to a database. The available data controls appear in a drop-down list next to the property name in the properties window
<b>Custom</b>	No access via custom property page.
<b>Visual Basic</b>	<i>object.DataSource</i> [= <i>setting %</i> ]
<b>Remarks</b>	DataSource is read/write at design time; not available at run time. The DataField, DataSource, and DataUpdate properties work together with the Visual Basic Data control to bind the RSTool control to a database.
<b>Data Type</b>	String

# DataUpdate Property

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Determines the primary source of displayed data as well as which data triggers a Change event.
<b>Visual Basic</b>	<i>object</i> . <b>DataUpdate</b> [= <i>setting %</i> ]
<b>Remarks</b>	Available settings for the DataUpdate property are 0 = Data Link, which displays only the data specified by the LinkItem; 1 = Data Source, which displays only the data (field in database) that the control is bound to; 2 = Data Link, Log To Data Source, which logs data to the Data Source specified by the data control; and 3 = No Update. When DataUpdate is set to 1 (Data Source), you will be able to view the contents of the field that the control is bound to. When DataUpdate is set to 2 (Data Link, Log to Data Source) and LinkMode is set to Automatic, the data control will add new records to the database whenever the control's value changes.
<b>Data Type</b>	Integer

# DataValue(Index) Property

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSWheel, RSVessel
<b>Description</b>	When requesting an array of data by using the block method (i.e. LinkItem = t4:0.acc, L25), you can use the DataValue property to select the specified element in the array. DataValue(0) for a single item data link is valid as well, even though it is not an array. DataValue(n) is an array property and must have an index.
<b>Visual Basic</b>	<i>object</i> . <b>DataValue</b> ( <i>item number</i> )
<b>Remarks</b>	Specifying array items in your LinkItem string makes it very easy to move large blocks of data with only one control. If you want to use element 25, simply specify that number. (i.e. rsdata1.datavalue(24)).
<b>Data Type</b>	Integer

# DecimalPlaces Property

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Determines the number of decimal places that will be shown when the value is displayed on the control.
<b>Custom</b>	Adjusted in the Value section of the custom properties page by changing the value in the "Decimals" text box. This value can be changed either by typing the number directly or by using the up and down scroll buttons.
<b>Visual Basic</b>	<i>object</i> . <b>DecimalPlaces</b> [= <i>setting</i> ]
<b>Remarks</b>	The range for the DecimalPlaces property value is from 0 to 9. TheDisplayValue property must be set to True before this property will have any visible effect, however the setting affects the <b>Value</b> and <b>DataValue(n)</b> properties whether visible or not.
<b>Data Type</b>	Integer

# DisplayBorder Property

<b>Applies To</b>	RSCompare, RSVessel
<b>Description</b>	Enables/disables all borders for the control's graphic shape.
<b>Custom</b>	Enabled/disabled in the Options window of the General section of the properties page.
<b>Visual Basic</b>	<i>Object</i> . <b>DisplayBorder</b> [= <i>setting</i> ]
<b>Remarks</b>	<p>In the case of the RSCompare control, when set to True a three-dimensional border will be displayed around the control's graphic shapes using the <b>DownBorderColor</b>, <b>EqualBorderColor</b>, <b>UpBorderColor</b>, <b>BorderHighlight</b>, and <b>BorderShadow</b> properties; when set to False none of those properties will be displayed. In the case of the RSVessel control, when set to True, a three-dimensional border around the fill area will be displayed using the <b>PolyBorderColor</b>, <b>BorderHighlight</b>, and <b>BorderShadow</b> properties; when set to False none of those properties will be displayed.</p>
<b>Data Type</b>	Integer

# DisplayCaption Property

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Determines if the caption will be displayed on the control.
<b>Custom</b>	Adjusted in the General section of the custom property page. Place the mouse pointer inside the setup frame of the General Tab and right-click to display the popup property menu. Clicking on "Caption" on this menu will toggle a check mark on and off which represents the True state for the DisplayCaption property. When the DisplayCaption is True, a red square caption position indicator appears on the control graphic in the General section.
<b>Visual Basic</b>	<i>object</i> . <b>DisplayCaption</b> [= <i>setting %</i> ]
<b>Remarks</b>	When set to True the Caption will be displayed; when set to False the Caption will not be displayed.
<b>Data Type</b>	Integer



# DisplayCaptionVertically Property

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Determines if the caption will be displayed vertically on the control.
<b>Custom</b>	Adjusted in the General section of the property page. The red indicator (square) on the control graphic represents the relative position of the caption on the control. Double clicking on the red indicator toggles the caption display between horizontal and vertical. When the caption is vertical the red indicator becomes a rectangle, and when the caption is horizontal the red indicator is a square.
<b>Visual Basic</b>	<i>object</i> . <b>DisplayCaptionVertically</b> [= <i>setting</i> ]
<b>Remarks</b>	When set to True, the caption will be displayed vertically; when set to False, the caption will be displayed horizontally.
<b>Data Type</b>	Integer

# DisplayPicture Property

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Determines if a picture will be displayed on the control.
<b>Custom</b>	Adjusted in the General section of the custom properties page with a checkbox in the Options window.
<b>Visual Basic</b>	<i>object</i> . <b>DisplayPicture</b> [= <i>setting %</i> ]
<b>Remarks</b>	When set to True, the picture designated in the Picture property will be displayed; when set to False, the picture will not be displayed. Because the RSCompare control has three possible picture properties (PictureUp, PictureEqual, PictureDown), this property applies to all three.
<b>Data Type</b>	Integer

# DisplayValue Property

<b>Applies To</b>	RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Determines if the control's current value will be displayed.
<b>Custom</b>	Adjusted in the General section of the custom property page. Place the mouse pointer inside the setup frame of the General Tab and right-click to display the popup property menu. Clicking on "Value" on this menu will toggle a check mark on and off which represents the True state for the DisplayValue property.
<b>Visual Basic</b>	<i>object</i> . <b>DisplayValue</b> [= <i>setting %</i> ]
<b>Remarks</b>	When set to True, the value will be displayed; when set to False the value will not be displayed.
<b>Data Type</b>	Integer

# DrawDisabledShadow Property

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Determines if a shadow will be displayed over the entire control when it is disabled.
<b>Custom</b>	Adjusted in the General section of the custom properties page with a checkbox in the Options window.
<b>Visual Basic</b>	<i>object</i> . <b>DrawDisabledShadow</b> [= <i>setting</i> %]
<b>Remarks</b>	When set to True, a shadow will be displayed when the control's Enabled property is set to False. When set to False, the shadow will not be displayed if the Enabled property is set to False. The shadow is not displayed if the control's Enabled property is set to True.
<b>Data Type</b>	Integer

# EndValue Property

<b>Applies To</b>	RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Sets the maximum for the value range of the control.
<b>Custom</b>	Adjusted in the "End Value" text box in the Value section of the custom properties page.
<b>Visual Basic</b>	<i>object</i> . <b>EndValue</b> [= <i>setting %</i> ]
<b>Remarks</b>	For the RSGauge and RSSlider controls, EndValue applies to both the Scale1 and Scale2 value ranges. For the RSCompare, RSData, and RSWheel controls, the EndValue will be ignored if the UseStartEndValue property is set to False.
<b>Data Type</b>	Double

# ExpressionForRead Property

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Determines the mathematical expression that will be performed on the link item value when the control reads that value.
<b>Custom</b>	Adjusted in the LinkInfo section of the custom property page. When the "Expression" button is pressed another form appears with two input boxes. Input the appropriate mathematical expression into the "Read Expression" text box. The expression must be in the form item[mathematical expression]. For example, "item+5" would add 5 to the LinkItem value.
<b>Visual Basic</b>	<i>object.ExpressionForRead</i> [= <i>setting %</i> ]
<b>Remarks</b>	<p>This property allows you to perform a mathematical function on the link item as it is read by the control. The expression must be in the form item[mathematical expression]. For example, "item+5" would add 5 to the LinkItem value.</p> <p>Optional Calculation/Math module required, RSCALC32.DLL.</p>
<b>Data Type</b>	String

# ExpressionForWrite Property

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Determines the mathematical expression that will be performed on the link item when the control writes that value.
<b>Custom</b>	Adjusted in the LinkInfo section of the custom property page. When the "Expression" button is pressed another form appears with two input boxes. Input the appropriate mathematical expression into the "Write Expression" text box. The expression must be in the form item[mathematical expression]. For example, "item+5" would add 5 to the LinkItem value.
<b>Visual Basic</b>	<i>object</i> . <b>ExpressionForWrite</b> [= <i>setting</i> %]
<b>Remarks</b>	<p>This property allows you to perform a mathematical function on the link item as a write is performed by the control. The expression must be in the form item[mathematical expression]. For example, "item+5" would add 5 to the LinkItem value.</p> <p>Optional Calculation/Math module required, RSCALC32.DLL.</p>
<b>Data Type</b>	String

# FaceBorderColor Property

<b>Applies To</b>	RSGauge, RSSlider, RSWheel
<b>Description</b>	Sets the color of the face border for the control.
<b>Custom</b>	Display section.
<b>Visual Basic</b>	<i>object</i> . <b>FaceBorderColor</b> [= <i>color</i> ]
<b>Remarks</b>	The FaceBorderColor property can be set to any color in the palette or a hex integer value representing a color. The DisplayFace property must be True before this property will have any visible effect.
<b>Data Type</b>	Color



# FaceColor Property

<b>Applies To</b>	RSGauge, RSSlider, RSWheel
<b>Description</b>	Sets the color of the face for the control.
<b>Custom</b>	Display section.
<b>Visual Basic</b>	<i>object</i> . <b>FaceColor</b> [= <i>setting</i> %]
<b>Remarks</b>	The FaceColor property can be set to any color in the palette or a hex integer value representing a color. The DisplayFace property must be True before this property will have any visible effect.
<b>Data Type</b>	Color

# FillColor Property

<b>Applies To</b>	RSGauge, RSVessel
<b>Description</b>	Determines the color of the filled area on the control.
<b>Custom</b>	Display section.
<b>Visual Basic</b>	<i>object.FillColor</i> [= <i>setting %</i> ]
<b>Remarks</b>	The FillColor property can be set to any color in the palette or a hex integer value representing a color. For the Gauge control this property is only used with the LED-style, Vertical and Horizontal Gauge types (4-7).
<b>Data Type</b>	Color

# FlashEnabled Property

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Determines if the control will flash when its value changes.
<b>Custom</b>	Enabled / disabled in the Display section of the properties page with the "Flash on New Data" check box.
<b>Visual Basic</b>	<i>object</i> .FlashEnabled[= <i>setting %</i> ]
<b>Data Type</b>	Integer

# FlashOn Property

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Determines if the control will continuously flash from visible to invisible.
<b>Custom</b>	Enabled / disabled in the Display section of the properties page with the "Flash Always On" check box.
<b>Visual Basic</b>	<i>object</i> .FlashOn [ = setting %]
<b>Remarks</b>	When set to True the control flashes continuously; when set to False the control does not flash. The FlashSpeed property adjusts the rate that the control flashes on and off.
<b>Data Type</b>	Integer

# FlashSpeed Property

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Determines the rate that the control flashes on and off.
<b>Custom</b>	Set on the Display section of the custom properties page with the "FlashSpeed" text box.
<b>Visual Basic</b>	<i>object</i> .FlashSpeed[= <i>setting</i> %]
<b>Remarks</b>	The setting for FlashSpeed is in milliseconds.
<b>Data Type</b>	Long

# FlashTime Property

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Determines how long the control should flash when server sends new data and FlashEnabled is True.
<b>Custom</b>	Set on the Display section of the custom properties page with the "FlashTime" text box.
<b>Visual Basic</b>	<i>object</i> .FlashTime[= <i>setting</i> %]
<b>Remarks</b>	Enabled only when the "Flash On New Data" check box is checked. The setting for FlashTime is in milliseconds.
<b>Data Type</b>	Long

# Font Property

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Determines the font name, style, and size of text that will be used for the control's caption and value.
<b>Custom</b>	Set on the Fonts section of the custom properties page.
<b>Visual Basic</b>	<i>object</i> . <b>Font</b> [= <i>setting</i> ]
<b>Remarks</b>	Available effects on the Fonts section of the custom properties page are Strikeout and Underline. These can be enabled and disabled with check boxes on the Font section.
<b>Data Type</b>	Font

# KnobWidth Property

<b>Applies To</b>	RSGauge, RSSlider
<b>Description</b>	Determines the width of the knob displayed on the control.
<b>Custom</b>	The knob width can be changed on the General section of the custom properties page by clicking the mouse on the knob graphic and drag-dropping the knob outline to the desired width.
<b>Visual Basic</b>	<i>object</i> . <b>KnobWidth</b> [= <i>setting %</i> ]
<b>Remarks</b>	<p>For the Gauge control, KnobWidth will have a visible effect only when the NeedleType property is set to a knob-type setting (3 = Knob; 4 = Knob-Plate; 5 = NeedleKnob; 6 = Needle-Knob-Plate). For the Slider, the KnobWidth property will affect both the button and pointed style knobs.</p> <p>For the Gauge control, KnobWidth is represented as Diameter.</p>
<b>Data Type</b>	Integer



# LeftBorder Property

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Sets the distance between left external border of the control and left edge of the control.
<b>Custom</b>	Adjusted in the General section of the custom properties page. Placing the mouse pointer over either of the control graphic's left corners will cause a sizing pointer to appear. Clicking and holding the left mouse button allows the bottom border for the control to be adjusted to the appropriate position with respect to the left external border of the control. Releasing the mouse button will then set the LeftBorder property.
<b>Visual Basic</b>	<i>object</i> . <b>LeftBorder</b> [= <i>setting %</i> ]
<b>Remarks</b>	The LeftBorder property controls the distance between the outside border of the control and the left edge of the control. Use this property to adjust the space available for displaying a caption or value.
<b>Data Type</b>	Integer

# LinkErrorDisplay Property

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Determines if an error message will be displayed in a control if an error has occurred when attempting to establish a DDE conversation with a LinkItem.
<b>Custom</b>	No access via custom property page.
<b>Visual Basic</b>	<i>[Form1.]Control.LinkErrorDisplay[=setting %]</i>
<b>Remarks</b>	When set to True, link error messages will be displayed; when set to False link error messages will not be displayed.
<b>Data Type</b>	Integer

# LinkErrorNumber Property (Run Time Only)

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Returns the error number associated with its LinkItem. Not available at design time and read only at run time.
<b>Custom</b>	No access via custom property page.
<b>Visual Basic</b>	<i>[Form1.]Control.LinkErrorNumber</i>
<b>Remarks</b>	Use the LinkErrorNumber property along with a label control to display the error number. For example: Label1.caption = rsdata1.LinkErrorNumber.
<b>Data Type</b>	Integer

# LinkErrorString Property (Run Time Only)

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Returns the error string associated with its LinkItem. Not available at design time and read only at run time.
<b>Custom</b>	No access via custom property page.
<b>Visual Basic</b>	<i>[Form1.]Control.LinkErrorString</i>
<b>Remarks</b>	Use the LinkErrorString property along with a label control to display the error string. For example: Label1.Caption = rsdata1.LinkErrorString.
<b>Data Type</b>	String

# LinkItem Property

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Sets the item portion of the data link string to which the control is linked.
<b>Custom</b>	Adjusted in the LinkInfo section of the custom property page.
<b>Visual Basic</b>	<i><b>object.LinkItem</b></i> [= <i>setting</i> %]
<b>Remarks</b>	Depending upon which type of DDE link is being established, the LinkItem can have many different formats, for example: "N7:1" is an integer address from a PLC datatable; "T4:0.ACC,L10" is a DDE block array, with a length of 10 items; "B3/0" is a binary address from a PLC datatable; "T4:11.ACC" is a timer address from a PLC datatable; and "r1c1" or "R2C2" are row and column addresses from a Microsoft Excel table.
<b>Data Type</b>	String

# LinkMode Property

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Sets the type of link to be used for a DDE conversation and activates the connection.
<b>Custom</b>	Adjusted in the LinkInfo section of the custom properties page.
<b>Visual Basic</b>	<i>object</i> . <b>LinkMode</b> [= <i>setting</i> %]
<b>Remarks</b>	Available options for the LinkMode property are: 0 = None; 1 = Automatic; 2 = Manual; and 3 = Notify.
<b>Data Type</b>	Integer

# LinkServer Property

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Determines the application or server name that the control is linked to.
<b>Custom</b>	Adjusted in the LinkInfo section of the custom properties page.
<b>Visual Basic</b>	<i>object</i> . <b>LinkServer</b> [= <i>setting</i> %]
<b>Remarks</b>	Depending upon which type of DDE link is being established, the LinkServer can have different formats, for example: "ICOMWDRV" is the DDE server name for WINTelligent Linx and "EXCEL" is the DDE server name for Microsoft Excel.
<b>Data Type</b>	String

# LinkTip Property

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel	
<b>Description</b>	Determines if a pop up LinkTip will be displayed whenever the mouse pointer is positioned above the control. The message in the LinkTip will either be the address that the control is linked to or a message specified by the LinkTipText property.	
<b>Custom</b>	Enabled / disabled in the Display section of the custom properties page with the "Link Tip" check box.	
<b>Visual Basic</b>	<i>object.LinkTip[=setting %]</i>	
<b>Settings</b>	<b><u>Settings</u></b>	<b><u>Description</u></b>
	True	Displays LinkTip window.
	False	Disables LinkTip window.
<b>Remarks</b>	When set to True, the LinkTip popup box will appear whenever the mouse pointer is positioned above the control; when set to False the LinkTip popup box will not appear. If there is not a string value entered for the LinkTipText property, the LinkTip popup message box will display the Link Server, Topic, and Item to which the control is connected.	
<b>Data Type</b>	Integer	



# LinkTipBackColor Property

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Sets the background color of the LinkTip popup box.
<b>Custom</b>	Color is set in the Display section of the custom properties page by choosing LinkTipBackColor in the properties Combo Box and then selecting a color in the color Combo Box.
<b>Visual Basic</b>	<i>object</i> . <b>LinkTipBackColor</b> [= <i>setting</i> ]
<b>Remarks</b>	The LinkTipBackColor property can be set to any color in the palette or a hex integer value representing a color. The LinkTip property must be set to True before this property will have any visible effect.
<b>Data Type</b>	Color

# LinkTipForeColor Property

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Determines the color of the LinkTip message text.
<b>Custom</b>	Color is set in the Display section of the custom properties page by choosing LinkTipForeColor in the properties Combo Box and then selecting a color in the color Combo Box.
<b>Visual Basic</b>	<i>object</i> . <b>LinkTipForeColor</b> [= <i>setting %</i> ]
<b>Remarks</b>	The LinkTipForeColor property can be set to any color in the palette or a hex integer value representing a color. The LinkTip property must be True before this property will have any visible effect.
<b>Data Type</b>	Color

# LinkTipText Property

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Displays a text string that will be displayed in the LinkTip popup box, or the symbol name of a DDE address or the actual DDE address (i.e. LinkServer, LinkTopic, LinkItem).
<b>Custom</b>	No access via custom property page.
<b>Visual Basic</b>	<i>object</i> . <b>LinkTipText</b> [=setting %]
<b>Remarks</b>	This property will only be effective when the LinkTip property is set to True. The LinkTip window has an order of precedence as follows: If there is not any string value entered for this property, then the LinkTip popup window will display the Symbol name associated with the DDE address. If a Symbol name is not used then the actual DDE address will be displayed (i.e. LinkServer, LinkTopic, and LinkItem) to which the control is connected.
<b>Data Type</b>	String

# LinkTopic Property

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Determines the topic portion of a data link string to which the control is linked.
<b>Custom</b>	Adjusted in the LinkInfo section of the custom property page.
<b>Visual Basic</b>	<i>object</i> . <b>LinkTopic</b> [= <i>setting</i> %]
<b>Remarks</b>	Depending upon which type of DDE link is being established, the LinkTopic can have different. Formats, for example: "testsol" would be a DDE topic created for the WINtelligent Linx driver and "sheet1.xls" would be a DDE topic name for Microsoft Excel.
<b>Data Type</b>	String

# MoveRefresh Property

<b>Applies To</b>	RSGauge, RSSlider
<b>Description</b>	Determines if the Windows repaint commands called messages will be acted upon immediately upon mouse movement over the control or if the Windows system will decide when to send the messages and repaint the control.
<b>Custom</b>	No access via custom properties page.
<b>Visual Basic</b>	<i>object</i> . <b>MoveRefresh</b> [= <i>setting %</i> ]
<b>Remarks</b>	When set to True the repaint messages will be acted upon immediately; when set to False the Windows system will decide the most appropriate time to send the message. On a fast machine there will probably not be a visible difference, therefore most users can set this property to False.
<b>Data Type</b>	Integer

# NotFilledColor Property

<b>Applies To</b>	RSGauge, RSVessel
<b>Description</b>	Determines the color of the not-filled area on the control.
<b>Custom</b>	Display section.
<b>Visual Basic</b>	<i>object</i> . <b>NotFilledColor</b> [= <i>setting %</i> ]
<b>Remarks</b>	The NotFilledColor property can be set to any color in the palette or a hex integer value representing a color. For the Gauge control this property is only used with the LED-style, Vertical and Horizontal Gauge types (4-7).
<b>Data Type</b>	Color

# NumberOfDataValues Property (Run Time Only)

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Returns the total number of individual data values represented by the control. Not available at design time. Read only at run time.
<b>Visual Basic</b>	<i>object</i> . <b>NumberOfDataValues</b>
<b>Data Type</b>	Integer

# NumberOfSegments Property

<b>Applies To</b>	RSCompare, RSVessel
<b>Description</b>	Sets the number of line segments to be drawn between the green bezier nodes in the control shape. (Green bezier nodes are visible on the custom properties page.) The higher the number of segments, the more round the control's shape will appear.
<b>Custom</b>	Segments text box on General page.
<b>Visual Basic</b>	<i>object</i> .NumberOfSegments[= <i>setting</i> %]
<b>Data Type</b>	Integer



# NumbersColor Property

<b>Applies To</b>	RSVessel, RSWheel
<b>Description</b>	Sets the color of the Start and End values within the control's display area. This only applies when the DisplayStartEndValues property is set to True.
<b>Custom</b>	Display Section
<b>Visual Basic</b>	<i>object</i> .NumbersColor[= <i>setting</i> %]

**Data TypeColor**

# Picture Property

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Any file of the *.bmp, *.wmf, *.ico format may be displayed on the control by defining the picture file.
<b>Custom</b>	Display Section.
<b>Visual Basic</b>	<i>object</i> . <b>Picture</b> [= <i>filename</i> ]
<b>Remarks</b>	Bitmaps, Windows metafiles, and icon files may be used as pictures for the control. The RSTools controls also support drag and drop of pictures from the optional RSWorkbench Visual Basic Add-In.
<b>Data Type</b>	String

# PictureStretch Property

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Stretches the picture to fit the control boundaries.
<b>Custom</b>	Check box in General Tab.
<b>Visual Basic</b>	<i>object</i> . <b>PictureStretch</b> [= <i>setting %</i> ]
<b>Remarks</b>	True stretches the Picture. False lets the picture be displayed in its original size.
<b>Data Type</b>	Integer

# PictureUp Property

<b>Applies To</b>	RSButton, RSCompare
<b>Description</b>	Sets the filename of a picture to be displayed within the control when the control is not depressed. (Or in the case of the RSCompare control, when the control is in its Up state.)
<b>Custom</b>	Display section.
<b>Visual Basic</b>	<i>object</i> . <b>PictureUp</b> [= <i>file</i> ]
<b>Remarks</b>	Picture files of the format *.bmp, *.wmf, and *.ico may be used. The RSCompare has several other picture properties including PictureUp; PictureDown, PictureEqual and Picture. Refer to the RSCompare documentation for information on these properties.
<b>Data Type</b>	String

# PokeLength Property

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Defines the number of controls in a control array to be used in a LinkPoke starting from the PokeStartIndex in a control array.
<b>Custom</b>	No access via custom property page.
<b>Visual Basic</b>	<i>object.PokeLength</i> [= <i>setting%</i> ]
<b>Remarks</b>	The number of controls to be used in single message transaction if AdvanceDDE is used.
<b>Data Type</b>	Integer

# PokeStartIndex Property

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Sets the starting index value of the controls in a control array to be used in a poke to the DDE server (source).
<b>Custom</b>	No access via custom property page.
<b>Visual Basic</b>	<i>object</i> . <b>PokeStartIndex</b> [= <i>setting</i> %]
<b>Data Type</b>	Integer

# RequestLength Property

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Defines the number of controls in a control array for a LinkRequest from the DDE server (source).
<b>Custom</b>	N/A on the Custom properties page
<b>Visual Basic</b>	<i>object</i> .RequestLength[= <i>setting</i> %]
<b>Remarks</b>	This property applies if the control is part of an array.
<b>Data Type</b>	Integer



# RequestStartIndex Property

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Sets the starting index value of the controls in a control array to be used in a request from the DDE server (source).
<b>Visual Basic</b>	<i>object</i> . <b>RequestStartIndex</b> [= <i>setting</i> %]
<b>Data Type</b>	Integer

# ReverseDirection Property

<b>Applies To</b>	RSGauge, RSSlider, RSWheel
<b>Description</b>	This property reverses the start and end values.
<b>Custom</b>	General section.
<b>Visual Basic</b>	<i>object</i> . <b>ReverseDirection</b> [= <i>setting</i> %]
<b>Remarks</b>	Setting to False uses the defined start and end values. Setting to True reverses the start and end values.
<b>Data Type</b>	Integer

# RightBorder Property

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Sets the distance between right external border of the control and right edge of the control.
<b>Custom</b>	Adjusted in the General section of the custom properties page. Placing the mouse pointer over either of the control graphic's right corners will cause a sizing pointer to appear. Clicking and holding the left mouse button allows the bottom border for the control to be adjusted to the appropriate position with respect to the right external border of the control. Releasing the mouse button will then set the RightBorder property.
<b>Visual Basic</b>	<i>object</i> . <b>RightBorder</b> [= <i>setting %</i> ]
<b>Remarks</b>	The RightBorder property controls the distance between the outside border of the control and the left edge of the control. Use this property to adjust the space available for displaying a caption or value.
<b>Data Type</b>	Integer

# Scale1DecimalPlaces Property

<b>Applies To</b>	RSGauge, RSSlider
<b>Description</b>	Sets or returns the number of decimal places used for the Scale 1 numerals.
<b>Custom</b>	Key in or Use the spin buttons next to 'Decimal' on the Scale Tab.
<b>Visual Basic</b>	<i>object</i> .Scale1DecimalPlaces[= <i>setting</i> %]
<b>Data Type</b>	Integer

# Scale1End Property

<b>Applies To</b>	RSGauge, RSSlider
<b>Description</b>	Sets or returns the End position and value of the Scale 1 markings.
<b>Custom</b>	Key in or click on the # sign to bring up the key pad on 'Place Scale1 @' settings box on the Scale Tab.
<b>Visual Basic</b>	<i>object</i> .Scale1End[= <i>setting</i> %]
<b>Data Type</b>	Double

# Scale1Length Property

<b>Applies To</b>	RSSlider, RSGauge
<b>Description</b>	Determines the physical length of the markings for Scale 1.
<b>Custom</b>	General Section. Can be edited directly on the picture.
<b>Visual Basic</b>	<i>object</i> .Scale1Length[= <i>setting</i> %]
<b>Data Type</b>	Integer

# Scale1Major Property

<b>Applies To</b>	RSSlider, RSGauge
<b>Description</b>	Sets the number of major scale divisions for Scale 1.
<b>Custom</b>	Key in or Use the spin buttons next to 'Major' on the Scale Tab.
<b>Visual Basic</b>	<i>object</i> .Scale1Major[= <i>setting</i> %]
<b>Data Type</b>	Integer

# Scale1MajorColor Property

<b>Applies To</b>	RSSlider, RSGauge
<b>Description</b>	Sets the major scale color of Scale 1.
<b>Custom</b>	Display section.
<b>Visual Basic</b>	<i>object</i> .Scale1MajorColor[= <i>setting</i> %]
<b>Data Type</b>	Color



# Scale1Minor Property

<b>Applies To</b>	RSGauge, RSSlider
<b>Description</b>	Sets the number of minor scale divisions for Scale 1.
<b>Custom</b>	Key in or Use the spin buttons next to ‘Minor’ on the Scale Tab.
<b>Visual Basic</b>	<i>object</i> . <b>Scale1Minor</b> [= <i>setting</i> %]
<b>Data Type</b>	Integer

# Scale1MinorColor Property

<b>Applies To</b>	RSGauge, RSSlider
<b>Description</b>	Sets the minor scale mark color of Scale 1.
<b>Custom</b>	Display section.
<b>Visual Basic</b>	<i>object</i> .Scale1MinorColor[= <i>setting</i> %]
<b>Data Type</b>	Color

# Scale1NumbersVisible Property

<b>Applies To</b>	RSGauge, RSSlider
<b>Description</b>	Determines whether or not Scale 1 numbers are visible.
<b>Custom</b>	Check Box on the Scale Tab.
<b>Visual Basic</b>	<i>object</i> .Scale1NumberVisible[= <i>setting</i> %]
<b>Remarks</b>	True makes the numbers visible. False hides the numbers.
<b>Data Type</b>	Integer

# Scale1Offset Property

<b>Applies To</b>	RSGauge, RSSlider
<b>Description</b>	Defines the offset of the Scale 1 markings from the inside border.
<b>Custom</b>	Double Click on the # mark next to scale in the Setup Area on the General Tab.
<b>Visual Basic</b>	<i>object</i> .Scale1Offset[= <i>setting</i> %]
<b>Data Type</b>	Integer

# Scale1Start Property

<b>Applies To</b>	RSGauge, RSSlider
<b>Description</b>	Determines the Start position and value of the Scale 1 markings on the control.
<b>Custom</b>	Key in or click on the # sign to bring up the key pad on 'Place Scale1 @' settings box on the Scale Tab.
<b>Visual Basic</b>	<i>object.Scale1Start[=setting %]</i>
<b>Data Type</b>	Double

# Scale1String Property

<b>Applies To</b>	RSSlider, RSGauge
<b>Description</b>	Determines the String Caption to be displayed next to Scale 1.
<b>Custom</b>	Not Available on Custom Property Pages.
<b>Visual Basic</b>	<i>object</i> . <b>Scale1String</b> [=setting%]
<b>Data Type</b>	String

# Scale1StringEnabled Property

<b>Applies To</b>	RSSlider, RSGauge
<b>Description</b>	Enables / Disables the display of String caption next to Scale 1.
<b>Custom</b>	Not Available on Custom Property Pages.
<b>Visual Basic</b>	<i>object</i> .Scale1StringEnabled[= <i>setting</i> %]
<b>Data Type</b>	Integer

# Scale1Style Property

<b>Applies To</b>	RSGauge, RSSlider
<b>Description</b>	Defines the position of the Scale 1 numbers on the control.
<b>Custom</b>	Drop down combo box on the Scale Tab.
<b>Visual Basic</b>	<i>object</i> .Scale1Style[= <i>setting</i> %]
<b>Remarks</b>	Valid styles are 0=Next Scale, 1=Inside Border and 2=Outside Border
<b>Data Type</b>	enumScaleStyles



# Scale1TextColor Property

<b>Applies To</b>	RSGauge, RSSlider
<b>Description</b>	Defines the text color for Scale 1.
<b>Custom</b>	Display Section
<b>Visual Basic</b>	<i>object</i> .Scale1TextColor[= <i>setting</i> %]
<b>Remarks</b>	Color can be chosen form the color palette or specified as RGB in Hex.
<b>Data Type</b>	Color

# Scale1TrailingZeros Property

<b>Applies To</b>	RSGauge, RSSlider
<b>Description</b>	Defines the number of trailing zeros for Scale 1.
<b>Custom</b>	Check Box on the Scale Tab.
<b>Visual Basic</b>	<i>object</i> .Scale1TrailingZeros[= <i>setting</i> %]
<b>Data Type</b>	Integer

# Scale1Type Property

<b>Applies To</b>	RSGauge, RSSlider
<b>Description</b>	Specifies the Scale 1 marking type.
<b>Custom</b>	Drop down combo box on the Scale Tab.
<b>Visual Basic</b>	<i>object</i> . <b>Scale1Type</b> [= <i>setting %</i> ]
<b>Remarks</b>	Valid style are 0=Normal, 1=Indented and 2=Bevel
<b>Data Type</b>	Integer

# Scale1Visible Property

<b>Applies To</b>	RSGauge, RSSlider
<b>Description</b>	Specifies if Scale 1 is visible or hidden.
<b>Custom</b>	Check Box on the Scale Tab.
<b>Visual Basic</b>	<i>object</i> .Scale1Visible[= <i>setting</i> %]
<b>Data Type</b>	Integer

# Scale1Width Property

<b>Applies To</b>	RSGauge, RSSlider
<b>Description</b>	Sets the width of the Scale 1 markings.
<b>Custom</b>	Edit directly on the setup area of the General custom properties tab.
<b>Visual Basic</b>	<i>object</i> . <b>Scale1Width</b> [= <i>setting</i> %]
<b>Data Type</b>	Integer

# Scale2DecimalPlaces Property

<b>Applies To</b>	RSGauge, RSSlider
<b>Description</b>	Sets or specifies the number of decimal places used for the Scale 2 numerals.
<b>Custom</b>	Key in or Use the spin buttons next to 'Decimal' on the Scale Tab.
<b>Visual Basic</b>	<i>object</i> .Scale2DecimalPlaces[= <i>setting</i> %]
<b>Data Type</b>	Integer

# Scale2End Property

<b>Applies To</b>	RSGauge, RSSlider
<b>Description</b>	Sets or returns the End position and value of the Scale 2 markings on the control.
<b>Custom</b>	Key in or click on the # sign to bring up the key pad on 'Place Scale1 @' settings box on the Scale Tab.
<b>Visual Basic</b>	<i>object</i> . <b>Scale2End</b> [= <i>setting</i> %]
<b>Data Type</b>	Double

# Scale2EndNumber Property

<b>Applies To</b>	RSGauge, RSSlider
<b>Description</b>	Sets or returns the end number value of the numerals on Scale 2.
<b>Custom</b>	Key in or Use the spin buttons next to 'Scale 2 Numbering' on the Scale Tab.
<b>Visual Basic</b>	<i>object</i> .Scale2EndNumber[= <i>setting</i> %]
<b>Remarks</b>	The Scale2End property value is the physical location of the end scale marking with respect to Scale1; Scale2EndNumber represents the text (number) that will be displayed at that end marking.
<b>Data Type</b>	Double



# Scale2Length Property

<b>Applies To</b>	RSGauge, RSSlider.
<b>Description</b>	Sets the length of the Scale 2 markings.
<b>Custom</b>	General section. Edit directly on the setup area.
<b>Visual Basic</b>	<i>object</i> .Scale2Length[= <i>setting</i> %]
<b>Data Type</b>	Integer

# Scale2Major Property

<b>Applies To</b>	RSGauge, RSSlider
<b>Custom</b>	Key in or Use the spin buttons next to 'Major' on the Scale Tab.
<b>Description</b>	Sets the number of major scale divisions for Scale 2.
<b>Visual Basic</b>	<i>object</i> .Scale2Major[= <i>setting</i> %]
<b>Data Type</b>	Integer

# Scale2MajorColor Property

Applies To	RSGauge, RSSlider
Description	Sets the major scale color for Scale 2.
Custom	Display section.
Visual Basic	<i>object</i> .Scale2MajorColor[= <i>setting</i> %]
Data Type	Color

# Scale2Minor Property

<b>Applies To</b>	RSGauge, RSSlider
<b>Description</b>	Sets the number of scale divisions for the minor scale of Scale 2.
<b>Custom</b>	Key in or Use the spin buttons next to 'Minor' on the Scale Tab.
<b>Visual Basic</b>	<i>object</i> .Scale2Minor[= <i>setting</i> %]
<b>Data Type</b>	Integer

# Scale2MinorColor Property

<b>Applies To</b>	RSGauge, RSSlider
<b>Description</b>	Sets Scale 2 minor scale color.
<b>Custom</b>	Display section.
<b>Visual Basic</b>	<i>object</i> .Scale2MinorColor[= <i>setting</i> %]
<b>Data Type</b>	Color

# Scale2NumbersVisible Property

<b>Applies To</b>	RSGauge, RSSlider
<b>Description</b>	Determines whether Scale 2 numbers are visible or not.
<b>Custom</b>	Check Box on the Scale Tab.
<b>Visual Basic</b>	<i>object</i> .Scale2NumbersVisible[= <i>setting</i> %]
<b>Remarks</b>	True sets the numbers to be visible. False hides the numbers.
<b>Data Type</b>	Integer

# Scale2Offset Property

<b>Applies To</b>	RSGauge, RSSlider
<b>Description</b>	Defines the offset of Scale 2 markings from the outside border.
<b>Custom</b>	No access via custom property page.
<b>Visual Basic</b>	<i>object</i> .Scale2Offset[= <i>setting</i> %]
<b>Data Type</b>	Integer

# Scale2Start Property

<b>Applies To</b>	RSGauge, RSSlider
<b>Description</b>	Sets or returns the Start position and value of the Scale 2 markings on the control.
<b>Custom</b>	Key in or click on the # sign to bring up the key pad on 'Place Scale2 @' settings box on the Scale Tab.
<b>Visual Basic</b>	<i>object</i> .Scale2Start[= <i>setting</i> %]
<b>Data Type</b>	Double



# Scale2StartNumber Property

<b>Applies To</b>	RSGauge, RSSlider
<b>Description</b>	Sets or returns the Scale 2 start number value of the scale numerals.
<b>Custom</b>	Key in or Use the spin buttons next to 'Scale 2 Numbering' on the Scale Tab.
<b>Remarks</b>	The Scale2Start property value is the physical location of the start scale marking with respect to Scale1; Scale2StartNumber represents the text (number) that will be displayed at that start marking.
<b>Visual Basic</b>	<i>object</i> . <b>Scale2StartNumber</b> [= <i>setting %</i> ]
<b>Data Type</b>	Double

# Scale2String Property

<b>Applies To</b>	RSSlider, RSGauge
<b>Description</b>	Determines the String Caption to be displayed next to Scale 1.
<b>Custom</b>	No access via custom property page.
<b>Visual Basic</b>	<i>object</i> . <b>Scale1String</b> [=setting%%]
<b>Data Type</b>	String

# Scale2StringEnabled Property

<b>Applies To</b>	RSSlider, RSGauge
<b>Description</b>	Enables / Disables the display of String caption next to Scale 1.
<b>Custom</b>	No access via custom property page.
<b>Visual Basic</b>	<i>object</i> .Scale1StringEnabled[= <i>setting</i> %]
<b>Data Type</b>	Integer

# Scale2Style Property

<b>Applies To</b>	RSGauge, RSSlider
<b>Description</b>	Defines the position of the Scale 2 numbers on the control.
<b>Custom</b>	Drop down combo box on the Scale Tab.
<b>Visual Basic</b>	<i>object</i> . <b>Scale2Style</b> [= <i>setting</i> %]
<b>Remarks</b>	Valid styles are 0=Next Scale, 1=Inside Border and 2=Outside Border.
<b>Data Type</b>	Integer

# Scale2TextColor Property

<b>Applies To</b>	RSGauge, RSSlider
<b>Description</b>	Define the text color for Scale 2.
<b>Custom</b>	Display section.
<b>Visual Basic</b>	<i>object</i> .Scale2TextColor[= <i>setting</i> %]
<b>Data Type</b>	Color

# Scale2TrailingZeros Property

<b>Applies To</b>	RSGauge, RSSlider
<b>Description</b>	Sets the number of trailing zeros for Scale 2.
<b>Custom</b>	Check Box on the Scale Tab.
<b>Visual Basic</b>	<i>object</i> .Scale2TrailingZeros[= <i>setting</i> %]
<b>Data Type</b>	Integer

# Scale2Type Property

<b>Applies To</b>	RSGauge, RSSlider
<b>Description</b>	Specifies the scale marking type for Scale 2.
<b>Custom</b>	Drop down combo box on the Scale Tab.
<b>Visual Basic</b>	<i>object</i> . <b>Scale2Type</b> [= <i>setting %</i> ]
<b>Remarks</b>	Valid types are 0=Normal, 1=Indented and 2=Bevel
<b>Data Type</b>	Integer

# Scale2Visible Property

<b>Applies To</b>	RSGauge, RSSlider
<b>Description</b>	Decides whether Scale 2 is visible or hidden.
<b>Custom</b>	Check Box on the Scale Tab.
<b>Visual Basic</b>	<i>object</i> . <b>Scale2Visible</b> [= <i>setting %</i> ]
<b>Remarks</b>	True makes the scale visible. False hides it.
<b>Data Type</b>	Integer



# Scale2Width Property

<b>Applies To</b>	RSGauge, RSSlider
<b>Description</b>	Sets the width of the scale markings for Scale 2.
<b>Custom</b>	General Section. Edit the picture directly in the setup area.
<b>Visual Basic</b>	<i>object</i> .Scale2Width[= <i>setting</i> %]
<b>Data Type</b>	Integer

# ScaleBorderColor Property

<b>Applies To</b>	RSGauge, RSSlider
<b>Description</b>	Sets the scale marking border color.
<b>Custom</b>	Display section.
<b>Visual Basic</b>	<i>object</i> .ScaleBorderColor[= <i>setting</i> %]
<b>Remarks</b>	This property is active with a Scale Type of 1. Color can be chosen from the color palette or specified in RGB format.
<b>Data Type</b>	Color

# ScaleHighlight Property

<b>Applies To</b>	RSGauge, RSSlider
<b>Description</b>	Set the scale marking highlight color.
<b>Custom</b>	Display section.
<b>Visual Basic</b>	<i>object</i> . <b>ScaleHighlight</b> [= <i>setting</i> %]
<b>Remarks</b>	This property is active with a Scale Type of 2. Color can be chosen from the color palette or specified in RGB format.
<b>Data Type</b>	Color

# ScaleShadow Property

<b>Applies To</b>	RSGauge, RSSlider
<b>Description</b>	Displays a shadow of the scale.
<b>Custom</b>	Check Box in the General section.
<b>Visual Basic</b>	<i>object</i> .ScaleShadow[= <i>setting %</i> ]
<b>Remarks</b>	This property is active when the Scale Type = 2. True displays the Shadow and False disables Shadow display.
<b>Data Type</b>	Color

# ScreenPriority Property

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Determines if the Windows dispatch commands called messages will be acted upon immediately or the Windows system will decide when to send the messages.
<b>Visual Basic</b>	<i>object</i> .ScreenPriority[= <i>setting</i> %]
<b>Remarks</b>	When set to True the messages will be acted on immediately. When set to False the Windows system decides the most appropriate time to send the message.
<b>Data Type</b>	Integer

# Shadow Property

<b>Applies To</b>	RSCompare, RSGauge, RSSlider, RSVessel
<b>Description</b>	Specifies if the control's knob or graphic shadow is visible or hidden.
<b>Custom</b>	Check box in General section.
<b>Visual Basic</b>	<i>object</i> . <b>Shadow</b> [= <i>setting</i> %]
<b>Remarks</b>	True sets the shadow visible. False hides the shadow.
<b>Data Type</b>	Integer

# ShadowOffsetX Property

<b>Applies To</b>	RSCompare, RSGauge, RSSlider, RSVessel
<b>Description</b>	Sets the horizontal offset of the control's knob or graphic shadow.
<b>Custom</b>	General section. Edit the picture directly in the setup area.
<b>Visual Basic</b>	<i>object</i> .ShadowOffsetX[= <i>setting</i> %]
<b>Data Type</b>	Integer

# ShadowOffsetY Property

<b>Applies To</b>	RSCompare, RSGauge, RSSlider, RSVessel
<b>Description</b>	Sets the vertical offset of the control's knob or graphic shadow.
<b>Custom</b>	General section. Edit the picture directly in the setup area.
<b>Visual Basic</b>	<i>object</i> .ShadowOffsetY[= <i>setting</i> %]
<b>Data Type</b>	Integer



# StartValue Property

<b>Applies To</b>	RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Specifies the starting value for the control.
<b>Custom</b>	Adjusted in the “Start Value” text box in the Value section of the custom properties page.
<b>Visual Basic</b>	<i>object.StartValue</i> [= <i>setting</i> %]
<b>Remarks</b>	For the RSGauge and RSSlider controls, StartValue applies to both the Scale1 and Scale2 value ranges. For the RSCompare, RSData, and RSWheel controls, the StartValue will be ignored if the UseStartEndValue property is set to False.
<b>Data Type</b>	Double

# Symbol Property

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Serves as an alias name for the server, topic and item to be used in a DDE link.
<b>Custom</b>	LinkInfo Section.
<b>Visual Basic</b>	<i>object.Symbol</i> [= <i>setting %</i> ]
<b>Remarks</b>	<p>To establish a symbol go to the LinkInfo tab on the control's custom property page. Select manage symbols and enter the symbol name, server, topic and item. After applying this data to the control, enter the Symbol name on the property sheet. The Linkserver, LinkTopic and LinkItem properties will be updated according to the symbol name entered.</p> <p>See also: <i>LinkTip</i> property.</p>
<b>Data Type</b>	String

# TabIndex Property

<b>Applies To</b>	RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Sets the tab index for the control.
<b>Custom</b>	No access via custom property page.
<b>Visual Basic</b>	<i>object</i> . <b>TabIndex</b> [= <i>setting</i> %]
<b>Remarks</b>	TabIndex is always one less than the total number of controls.
<b>Data Type</b>	Integer

# TabStop Property

<b>Applies To</b>	RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Adds or removes the control from the form's Tab order at run time.
<b>Custom</b>	No access via custom property page.
<b>Visual Basic</b>	<i>object</i> . <b>TabStop</b> [= <i>setting</i> %]
<b>Remarks</b>	Setting to True adds the control to the form's tab order. False removes the control from the tab order.
<b>Data Type</b>	Integer

# Tag Property

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Sets or returns an expression that stores any extra data needed in the application.
<b>Custom</b>	No access via custom property page.
<b>Visual Basic</b>	<i>object</i> . <b>Tag</b> [= <i>setting</i> %]
<b>Remarks</b>	This is a user defined property that does not affect other VB properties.
<b>Data Type</b>	String

# Top Property

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Determines the distance between the top edge of a control and the top edge of its container.
<b>Custom</b>	No access via custom property page.
<b>Visual Basic</b>	<i>object</i> . <b>Top</b> [= <i>setting %</i> ]
<b>Data Type</b>	Single

# TopBorder Property

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Determines the distance between the top external border of the control and the top of the control.
<b>Custom</b>	Adjusted in the General section of the custom properties page. Placing the mouse pointer over either of the control graphic's lower corners will cause a sizing pointer to appear. Clicking and holding the left mouse button allows the top border for the control to be adjusted to the appropriate position with respect to the top external border of the control. Releasing the mouse button will then set the TopBorder property.
<b>Visual Basic</b>	<i>object</i> . <b>TopBorder</b> [= <i>setting</i> %]
<b>Data Type</b>	Integer

# TrailingZeros Property

<b>Applies To</b>	RSCompare, RSData, RSGauge, RSSlider, RSVessel
<b>Description</b>	Determines if the value is displayed with trailing insignificant zeros.
<b>Custom</b>	Check box in General section.
<b>Visual Basic</b>	<i>object.TrailingZeros</i> [= <i>setting</i> %]
<b>Remarks</b>	True displays the value with trailing zeros. False displays the value without trailing zeros.
<b>Data Type</b>	Integer



# UseInPoke Property

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	A flag to signify that the control should be used in a LinkPoke or DoPoke.
<b>Custom</b>	No access via custom property page.
<b>Visual Basic</b>	<i>object.UseInPoke</i> [= <i>setting %</i> ]
<b>Remarks</b>	True sets the control to be used in a poke. False sets the control not to be used. This property may be set for each element of a control array, which allows individual controls to be “skipped” in a block write action.
<b>Data Type</b>	Integer

# UseInRequest Property

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	A flag to signify that the control should be used in a LinkRequest or DoRequest.
<b>Custom</b>	No access via custom property page.
<b>Visual Basic</b>	<i>object</i> .UseInRequest[= <i>setting %</i> ]
<b>Remarks</b>	True sets the control to be used in a request. False sets the control not to be used. This property may be set for each element of a control array, which allows individual controls to be “skipped” in a block read action.
<b>Data Type</b>	Integer

# UseStartEndValue Property

<b>Applies To</b>	RSCompare, RSData, RSWheel
<b>Description</b>	Enables / disables use of a minimum and maximum value range for the applicable controls, with the minimum and maximum being the values of the StartValue and EndValue properties.
<b>Custom</b>	Adjusted in the General section of the custom properties page with a checkbox in the Options window.
<b>Visual Basic</b>	<i>object</i> .UseStartEndValue[= <i>setting</i> %]
<b>Remarks</b>	When set to True, the control will not allow its Value to fall outside of the StartValue to EndValue range. When set to False, the control will allow Values outside of that range.
<b>Data Type</b>	Integer

# Value Property

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Contains the value of the control at runtime.
<b>Custom</b>	Value section.
<b>Visual Basic</b>	<i>object.Value</i> [= <i>setting %</i> ]
<b>Remarks</b>	<p>Since the RSTools controls are capable of displaying multiple LinkItem addresses with an array of controls, you can preview an array of controls at design time with the Value property by setting the Value to a comma-separated string of values. For instance, a Value setting of “0,1,2,3,4” will display an array of 5 controls within the one instance of the RSTools control, each with the corresponding Value displayed.</p> <p>See also: <i>Clip</i> property.</p>
<b>Data Type</b>	String

# ValueBackColor Property

<b>Applies To</b>	RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Sets the back color for the control's value.
<b>Custom</b>	Display section.
<b>Visual Basic</b>	<i>object.ValueBackColor</i> [= <i>setting</i> %]
<b>Remarks</b>	Is visible only if the VALUE is set to transparent.
<b>Data Type</b>	Color

# ValueColor Property

<b>Applies To</b>	RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Sets the foreground color of the control's value.
<b>Custom</b>	Display section.
<b>Visual Basic</b>	<i>object</i> . <b>ValueColor</b> [= <i>setting</i> %]
<b>Data Type</b>	Color

# ValuePadDownload Property

<b>Applies To</b>	RSCompare, RSData, RSVessel
<b>Description</b>	Enables / disables the number entry pad from being activated when the control is clicked on at run time.
<b>Custom</b>	Check box in the General section.
<b>Visual Basic</b>	<i>object.ValuePadDownload</i> [= <i>setting</i> %]
<b>Data Type</b>	Integer

# ValueShadow Property

<b>Applies To</b>	RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Turns On/Off the value shadow.
<b>Custom</b>	Check box in General section.
<b>Visual Basic</b>	<i>object.ValueShadow</i> [= <i>setting</i> %]
<b>Remarks</b>	True displays the shadow and False hides it.
<b>Data Type</b>	Integer



# ValueShadowColor Property

<b>Applies To</b>	RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Determines the color of the value's shadow.
<b>Custom</b>	Display section.
<b>Visual Basic</b>	<i>object</i> .ValueShadowColor[= <i>setting</i> %]
<b>Data Type</b>	Color

# ValueTransparent Property

<b>Applies To</b>	RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Determines if the value background is transparent or opaque.
<b>Custom</b>	Check box in General section.
<b>Visual Basic</b>	<i>object</i> .ValueTransparent[= <i>setting %</i> ]
<b>Remarks</b>	True sets it to transparent. False sets it to opaque.
<b>Data Type</b>	Integer

# ValueX Property

<b>Applies To</b>	RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Sets the horizontal position of the Value display within the control.
<b>Custom</b>	General section. Edit the picture directly.
<b>Visual Basic</b>	<i>object.ValueX</i> [= <i>setting %</i> ]
<b>Remarks</b>	The position can be set from 0 to 100. The CenterOnKnob property of the RSSlider should be False to display the value somewhere other than the knob.
<b>Data Type</b>	Integer

# ValueY Property

<b>Applies To</b>	RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Sets the vertical position of the Value display within the control.
<b>Custom</b>	General section. Edit the picture directly.
<b>Visual Basic</b>	<i>object.ValueY</i> [=setting %]
<b>Remarks</b>	The position can be set from 0 to 100. The CenterOnKnob property for the RSSlider should be False to display the value somewhere other than the knob.
<b>Data Type</b>	Integer

# Visible Property

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Determines if the control is visible at run-time.
<b>Visual Basic</b>	<i>object</i> . <b>Visible</b> [= <i>setting</i> %]
<b>Remarks</b>	True sets the control to be visible and False hides it.
<b>Data Type</b>	Integer

# WhatsThisHelpID Property

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Sets or returns the associated WhatsThisHelp context number.
<b>Visual Basic</b>	<i>object</i> . <b>WhatsThisHelpID</b> [= <i>number</i> ]
<b>Remarks</b>	These context numbers are associated with Windows help files.
<b>Data Type</b>	Long

# Width Property

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Sets or returns the width of the control.
<b>Custom</b>	General section. Edit the picture directly.
<b>Visual Basic</b>	<i>object.</i> <b>Width</b> [= <i>setting %</i> ]
<b>Data Type</b>	Single

# WriteStyle Property

<b>Applies To</b>	RSButton, RSGauge, RSSlider, RSWheel
<b>Description</b>	Sets the write style to the DDE server.
<b>Custom</b>	Value section.
<b>Visual Basic</b>	<i>object</i> . <b>WriteStyle</b> [= <i>setting</i> %]
<b>Remarks</b>	Options are: 0 - ReadOnly; 1- Continuous, 2- Release.
<b>Data Type</b>	Integer



# WriteValue Property

<b>Applies To</b>	RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Determines if the value will be downloaded to a database or LinkItem.
<b>Custom</b>	The WriteValue is used when a person programmatically changes a value.
<b>Visual Basic</b>	<i>object</i> . <b>WriteValue</b> [= <i>setting</i> %]
<b>Data Type</b>	Integer

## **Event Reference**

This chapter provides a complete alphabetical reference for the RSTools common controls' events. Refer to Chapter 2, Programming Tools, for additional information about using events.

# Change Event

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Indicates that the contents of a control's Value property have changed.
<b>Visual Basic</b>	<b>Private Sub <i>object</i>_Change(<i>index</i> As Integer)</b>
<b>Remarks</b>	<p>The Change event procedure can synchronize or coordinate data display among controls.</p> <p>For additional information, refer to the description of the <b>Change</b> event in the Microsoft Visual Basic Language Reference Manual.</p>

# Click Event

<b>Applies To</b>	RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Occurs when the user presses and then releases a mouse button over an object.
<b>Visual Basic</b>	<b>Private Sub <i>object</i>_Click(<i>[index As Integer]</i>)</b>
<b>Remarks</b>	<p>Typically, you attach a Click event procedure to a CommandButton control, Menu object, or PictureBox control to carry out commands and command-like actions. For the other applicable controls, use this event to trigger actions in response to a change in the control.</p> <p>For additional information, refer to the description of the <b>Click</b> event in the Microsoft Visual Basic Language Reference Manual.</p>

# Clicks Event

<b>Applies To</b>	RSButton, RSCompare, RSVessel
<b>Description</b>	The Clicks event occurs when the user presses and releases the mouse button while the pointer is in the control's window and the control's ActsLikeButton property is set to True.
<b>Visual Basic</b>	<b>Sub RSButton1_Click</b> ([ByVal dValue As Double, ByVal iButtonIndex As Integer])
<b>Remarks</b>	Differs from the click event in that when the RSVessel (RSButton or RSCompare) control is linked to an array of data items, the index of the RSVessel that is clicked on within that array is passed to the event as well as that RSVessel's value. The ActsLikeButton property must be set to True for this event to fire

# DblClick Event

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Occurs when the user presses and releases a mouse button and then presses and releases it again over an object.
<b>Visual Basic</b>	<b>Private Sub <i>object</i>_DblClick (<i>index As Integer</i>)</b>
<b>Remarks</b>	<p>The argument Index uniquely identifies a control if it's in a control array. You can use a DblClick event procedure for an implied action, such as double-clicking an icon to open a window or document.</p> <p>For additional information, refer to the description of the DblClick event in the Microsoft Visual Basic Language Reference Manual.</p>

# DragDrop Event

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Occurs when a drag-and-drop operation is completed as a result of dragging a control over a form or control and releasing the mouse button or using the Drag method with its action argument set to 2 (Drop).
<b>Visual Basic</b>	<b>Private Sub <i>object</i>_DragDrop(<i>[index As Integer]</i>,<i>source As Control</i>, <i>x As Single</i>, <i>y As Single</i>)</b>
<b>Remarks</b>	<p>Use a DragDrop event procedure to control what happens after a drag operation is completed.</p> <p>For additional information, refer to the description of the DragDrop event in the Microsoft Visual Basic Language Reference Manual.</p>

# DragOver Event

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Occurs when a drag-and-drop operation is in progress. You can use this event to monitor the mouse pointer as it enters, leaves, or rests directly over a valid target. The mouse pointer position determines the target object that receives this event.
<b>Visual Basic</b>	<b>Private Sub <i>object</i>_DragOver(<i>[index As Integer,]</i>source As Control, x As Single, y As Single, state As Integer)</b>
<b>Remarks</b>	<p>Use a DragOver event procedure to determine what happens after dragging is initiated and before a control drops onto a target.</p> <p>For additional information, refer to the description of the DragOver event in the Microsoft Visual Basic Language Reference Manual.</p>



# EndMove Event

<b>Applies To</b>	RSGauge, RSSlider
<b>Description</b>	Occurs when the user releases the mouse button after using the mouse pointer to position the RSGauge needle or RSSlider knob to a new value.
<b>Visual Basic</b>	<b>Private Sub <i>object</i>_EndMove ([byVal <i>Value</i> As Double, ByVal <i>Index</i> As Integer])</b>
<b>Remarks</b>	<p>The EndMove event can be used to dictate what should happen after the RSGauge needle or RSSlider knob has been moved to a new value using the mouse pointer.</p> <p><b>Note:</b> This event is unavailable when using a RSGauge type with fill boxes unless you apply a needle to it.</p>

# GotFocus Event

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Occurs when an object receives the focus, either by user action, such as tabbing to or clicking the object, or by changing the focus in code using the SetFocus method.
<b>Visual Basic</b>	<b>Private Sub <i>object</i>_GotFocus(<i>[index As Integer]</i>)</b>
<b>Remarks</b>	<p>Typically, you use a GotFocus event procedure to specify the actions that occur when a control or form first receives the focus.</p> <p>For additional information, refer to the description of the GotFocus event in the Microsoft Visual Basic Language Reference Manual.</p>

# KeyDown, KeyUp Events XE " KeyUp Event"

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Occur when the user presses (KeyDown) or releases (KeyUp) a key while an object has the focus.
<b>Visual Basic</b>	<b>Private Sub <i>object</i>_KeyDown(<i>[index As Integer],[keycode As Integer, shift As Integer]</i>)</b>  <b>Private Sub <i>object</i>_KeyUp(<i>[index As Integer],[keycode As Integer, shift As Integer]</i>)</b>
<b>Remarks</b>	For both events, the object with the focus receives all keystrokes. For additional information, refer to the description of the KeyDown,KeyUp events in the Microsoft Visual Basic Language Reference Manual.

# KeyPress Event

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Occurs when the user presses and releases an ANSI key.
<b>Visual Basic</b>	<b>Private Sub <i>object</i>_KeyPress(<i>[index As Integer]</i>,<i>[keyascii As Integer]</i>)</b>
<b>Remarks</b>	The object with the focus receives the event. For additional information, refer to the description of the KeyPress event in the Microsoft Visual Basic Language Reference Manual.

# LinkError Event

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Occurs when there is an error during a DDE conversation. This event is recognized only as the result of a DDE-related error that occurs when no Visual Basic code is being executed. The error number is passed as an argument.
<b>Visual Basic</b>	<b>Private Sub <i>object</i>_LinkError(ByVal <i>iRet</i> As Integer, ByVal <i>ErrorString</i> As String)</b>
<b>Remarks</b>	Use a LinkError event procedure to notify the user of the particular error that has occurred. For additional information, refer to the description of the LinkError event in the Microsoft Visual Basic Language Reference Manual.

# LinkItemNotSupported Event

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Occurs when the control's LinkItem has an incorrect format.
<b>Visual Basic</b>	<b>Private Sub </b> <i>object</i> <b>_LinkItemNotSupported ( )</b>

# LinkItemSupported Event

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Occurs when the control's LinkItem has a correct format.
<b>Visual Basic</b>	<b>Private Sub </b> <i>object</i> <b>_LinkItemSupported ( )</b>

# LinkNotify Event

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Occurs when the source has changed the data defined by the DDE link if the LinkMode property of the destination control is set to 3 (Notify).
<b>Visual Basic</b>	<b>Private Sub <i>object</i>_LinkNotify([ByVal <i>index</i> As Integer])</b>
<b>Remarks</b>	Typically, in the LinkNotify event your code notifies the user, gets the new data immediately, or defers getting the data until later. You can use the LinkRequest method to obtain the new data from the source. For additional information, refer to the description of the LinkNotify event in the Microsoft Visual Basic Language Reference Manual.



# LinkOutOfMemory Event

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Occurs when the client (control) has exhausted its memory resources.
<b>Visual Basic</b>	<b>Private Sub <i>object</i>_DbClick ([ByVal <i>Index</i> As Integer])</b>

# LinkServerDisconnected Event

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Occurs when the control is connected to a server and that server becomes unavailable..
<b>Visual Basic</b>	<b>Private Sub </b> <i>object</i> <b>_LinkServerDisconnected ( )</b>

# LinkUnableToConnectToServer Event

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Occurs when the control attempts to connect to a server that is unavailable.
<b>Visual Basic</b>	<b>Private Sub </b> <i>object</i> <b>_LinkUnableToConnectToServer ( )</b>

# LostFocus Event

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Occurs when an object loses the focus, either by user action, such as tabbing to or clicking another object, or by changing the focus in code using the SetFocus method.
<b>Visual Basic</b>	<b>Private Sub <i>object</i>_LostFocus(<i>[index As Integer]</i>)</b>
<b>Remarks</b>	A LostFocus event procedure is primarily useful for verification and validation updates. Using LostFocus can cause validation to take place as the user moves the focus from the control. For additional information, refer to the description of the Click event in the Microsoft Visual Basic Language Reference Manual.

# MouseDown, MouseUp Events

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Occur when the user presses (MouseDown) or releases (MouseUp) a mouse button.
<b>Visual Basic</b>	<b>Private Sub <i>object</i> _MouseDown(<i>index</i> As Integer,<i>button</i> As Integer, <i>shift</i> As Integer, <i>x</i> As Single, <i>y</i> As Single)</b>  <b>Private Sub <i>object</i> _MouseUp(<i>index</i> As Integer,<i>button</i> As Integer, <i>shift</i> As Integer, <i>x</i> As Single, <i>y</i> As Single)</b>
<b>Remarks</b>	Use a MouseDown or MouseUp event procedure to specify actions that will occur when a given mouse button is pressed or released. For additional information, refer to the description of the MouseDown, MouseUp events in the Microsoft Visual Basic Language Reference Manual.

# MouseMove Event

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	Occurs when the user moves the mouse.
<b>Visual Basic</b>	<b>Private Sub <i>object</i> _MouseMove(<i>[index As Integer,]</i>button As Integer, <i>shift As Integer, x As Single, y As Single</i>)</b>
<b>Remarks</b>	The MouseMove event is generated continually as the mouse pointer moves across objects. For additional information, refer to the description of the MouseMove event in the Microsoft Visual Basic Language Reference Manual.

# PokeCompleted Event

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	The PokeCompleted event occurs when a poke action has completed
<b>Visual Basic</b>	<b>Private Sub <i>object</i>_PokeCompleted ([ByVal <i>iRet</i> As Integer])</b>
<b>Remarks</b>	The argument iRet returns error numbers

# RequestCompleted Event

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
<b>Description</b>	The RequestCompleted event occurs when a request action has completed.
<b>Visual Basic</b>	<b>Private Sub <i>object</i>_RequestCompleted ([ByVal <i>iRet</i> as Integer])</b>
<b>Remarks</b>	The argument iRet returns error numbers



# StartMove Event

<b>Applies To</b>	RSGauge, RSSlider
<b>Description</b>	Occurs when the RSSlider's knob or RSGauges Needle begins to move.
<b>Visual Basic</b>	<b>Private Sub </b> <i>object</i> <b>_StarMove([ByVal </b> <i>Value</i> <b> As Double, ByVal </b> <i>Index</i> <b> as Integer])</b>

## Methods Reference

This chapter provides a reference for the RSTools common methods. Refer to the individual component manuals in Chapter 2, Programming Tools, for additional information about using methods.

# Container Method

<b>Applies To</b>	RSButton, RSCompare, RSData, EventMaster, RSGauge, RSSlider, RSWheel
<b>Description</b>	Returns or sets the container of a control.
<b>Visual Basic</b>	<i><b>object.Control.Container</b></i> [= <i>setting</i> ]
<b>Remarks</b>	Not available at design time. A control's parent (container) may be changed at run time with the Container method.

# DoPoke Method

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSWheel, RSVessel
<b>Description</b>	Writes the values of all controls in an array to the server in a DDE conversation. To work successfully the control's UseInPoke property must be set to True.
<b>Visual Basic</b>	<i>object.Control</i> . <b>DoPoke</b>
<b>Remarks</b>	The control's LinkMode property must be set to 2 - Manual. See also: <i>PokeStartIndex</i> and <i>PokeLength</i> properties, <i>LinkPoke</i> method.

# DoRequest Method

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSWheel, RSVessel
<b>Description</b>	Requests the source application in a DDE conversation to update an array of control values. LinkMode should be set to either None(0) or Manual(2). To work successfully, the control's UseInRequest property must be set to True.
<b>Visual Basic</b>	<i>object.Control.DoRequest</i>
<b>Remarks</b>	<p>The control's LinkMode property must be set to 2 - Manual.</p> <p>See also: <i>RequestStartIndex</i> and <i>RequestLength</i> properties, <i>LinkRequest</i> Method.</p>

# Drag Method

**Applies To** RSButton, RSCompare, RSData, RSGauge, RSSlider, RSWheel, RSVessel

**Description** Begins, ends, or cancels a drag operation of any control except the Line, Menu, Shape, Timer, or CommonDialog controls.

**Visual Basic** *object*.**Drag** action

**Settings** The settings for action are:

<u>Constant</u>	<u>Value</u>	<u>Description</u>
vbCancel	0	Cancels drag operation.
vbBeginDrag	1	Begins dragging object.
vbEndDrag	2	Ends dragging and drop object.

**Remarks** Using the Drag method to control a drag-and-drop operation is required only when the DragMode property of the object is set to Manual (0). However, you can use Drag on an object whose DragMode property is set to Automatic (1 or vbAutomatic).

For more information refer to the Microsoft Visual Basic Language Reference.

# LinkPoke Method

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSWheel, RSVessel
<b>Description</b>	Transfers the value of a control to the source application in a DDE conversation. To work successfully the control's UseInPoke property must be set to True.
<b>Visual Basic</b>	<i>object</i> . <b>LinkPoke</b>
<b>Remarks</b>	<p>Typically, information in a DDE conversation flows from source to destination. However, LinkPoke allows a destination object to supply RSData to the source.</p> <p>See also: <i>PokeStartIndex</i> and <i>PokeLength</i> properties, <i>DoPoke</i> method.</p>

# LinkRequest Method

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSWheel, RSVessel
<b>Description</b>	Requests the source application in a DDE conversation to update the value of the control. LinkMode should be set to either None(0) or Manual(2). To work successfully the control's UseInRequest property must be set to True.
<b>Visual Basic</b>	<i>object</i> . <b>LinkRequest</b>
<b>Remarks</b>	<p>LinkRequest causes the source application to send the most current RSData to object.</p> <p>See also: <i>RequestStartIndex</i> and <i>RequestLength</i> properties, <i>DoRequest</i> method.</p>



# Move Method

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSWheel, RSVessel
<b>Description</b>	Moves an MDIForm, Form, or control.
<b>Visual Basic</b>	<i>object</i> . <b>Move</b> left, top, width, height
<b>Remarks</b>	<p>Only the left argument is required. However, to specify any other arguments, you must specify all arguments that appear in the syntax before the argument you want to specify.</p> <p>For more information refer to the Microsoft Visual Basic Language Reference.</p>

# Object Method

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSWheel, RSVessel, RSSlider
<b>Description</b>	Returns an object in a control.
<b>Visual Basic</b>	<i>object.Control.</i> <b>Object</b> [.property .method][=value]
<b>Remarks</b>	For more information refer to the Microsoft Visual Basic Language Reference.

# Parent Method

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSWheel, RSVessel
<b>Description</b>	Returns the container on which an object is located.
<b>Visual Basic</b>	<i>object.Control</i> . <b>Parent</b>
<b>Remarks</b>	For more information refer to the Microsoft Visual Basic Language Reference.

# SetFocus Method

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSWheel, RSVessel
<b>Description</b>	Moves the focus to the specified control or form.
<b>Visual Basic</b>	<i>object</i> . <b>SetFocus</b>
<b>Remarks</b>	<p>The object (control) that can receive the focus. After invoking the SetFocus method, any user input is directed to the specified control.</p> <p>For more information refer to the Microsoft Visual Basic Language Reference.</p>

# ShowWhatsThis Method

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSWheel, RSVessel
<b>Description</b>	Displays a selected topic in a Help file using the What's This popup provided by Windows 95 Help.
<b>Visual Basic</b>	<i>object</i> . <b>ShowWhatsThis</b>
<b>Remarks</b>	The ShowWhatsThis method is very useful for providing context-sensitive Help from a context menu in your application. The method displays the topic identified by the WhatsThisHelpID property of the object specified in the syntax.

# ZOrder Method

<b>Applies To</b>	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSWheel, RSVessel	
<b>Description</b>	Places a specified MDIForm, Form, or control at the front or back of the z-order within its graphical level.	
<b>Visual Basic</b>	<i>object</i> . <b>ZOrder</b> position	
<b>Remarks</b>	The ZOrder method syntax has these parts:	
	<u>Part</u>	<u>Description</u>
	object	Optional. An object expression that evaluates to an object in the Applies To list. If object is omitted, the form with the focus is assumed to be object.
	position	Optional. Integer indicating the position of object relative to other instances of the same object. If position is 0 or omitted, object is positioned at the front of the z-order. If position is 1, object is positioned at the back of the z-order.

