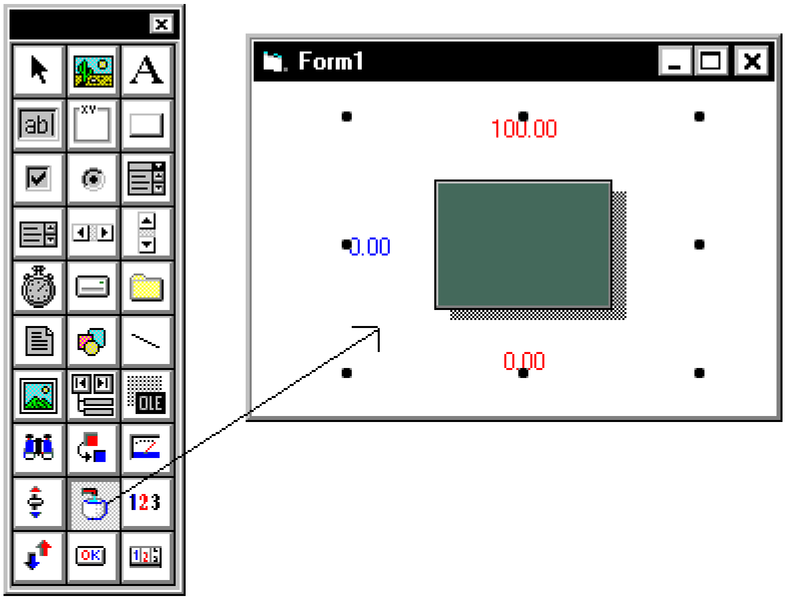
Introduction

Properties Events Methods

Description

The RSVessel control is a free-form dynamic-fill shape control. The free-form shape editor allows addition and removal of nodes/sides, 360 degree rotation, and bezier curve segments. Also provides multiple “fixed” shapes (triangle, pentagon, etc.), 360 degree fill direction, area fill, center in/out, and mirror in/out fill styles, as well as the ability to fill on top of overlay graphics. The RSVessel control can also be configured to act as a button control.



File Name

RSVSL32.OCX

Object Type

RSVessel

Remarks

The RSVessel control has several custom properties that allow you to monitor and control data. The RSVessel 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 RSVessel control, certain files need to be installed in the WINDOWS\SYSTEM directory. Refer to the “Redistributing Files” section of this chapter for specific file names. The Setup Wizard included with Visual Basic provides tools to help create setup programs for installing your applications.

Basic Concepts

- The RSVessel control is designed to be used with any Dynamic Data Exchange Server and can provide enhanced performance when used with those that conform to AdvanceDDEprotocol. In order for RSVessel to communicate via AdvanceDDE or XL_Table protocols, RSJunctionBox, the module that provides RSVessel 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 RSVessel control can be easily configured by setting its properties, without writing a single line of code.
- The RSVessel control can be used as a bound control. It automatically handles adding, updating and displaying data from the Data control, which supports ODBC, Microsoft Access, dBASE, Foxpro, Paradox, Btrieve and several other database types.
- The RSVessel control also supports the Paste Link method from RSLinx or WINtelligent Linx, Microsoft Excel or other servers for transferring DDE Link data via the clipboard.
- The RSVessel 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 its events.

Bound Properties

The RSVessel control has three bound properties: **DataUpdate**, **DataField** and **DataSource**. These properties allow the RSVessel control to be linked to a Visual Basic Data control or Remote Data control, and display field values for the current record in the 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 & Methods

All of the properties, events, and methods for the RSVessel control are listed in the following tables.

Properties

(About)
(Custom)
ActsLikeButton
AllowChangeEvent
AutoColumns
BackColor
BackStyle
BevelHeight
BevelHighlight
BevelShadow
BevelStyle
BevelWidth
BorderBeveled
BorderColor
BorderHighlight
BorderInner
BorderInnerColor
BorderShadow
BorderStyle
BorderWidth
BottomBorder
ButtonBorderWidth
Caption
CaptionBackColor
CaptionColor
CaptionShadow
CaptionShadowColor
CaptionTransparent
CaptionX
CaptionY
Clip
DataChanged
DataField
DataSource
DataUpdate
DataValue
DecimalPlaces
DisplayBorder
DisplayCaption
DisplayCaptionVertically
DisplayPicture
DisplayStartEndValues
DisplayValue
DragIcon
DragMode
DrawDisabledShadow
Enabled
EndValue
ExpressionForRead
ExpressionForWrite
FillColor
FillEndX
FillEndY
FillStartX
FillStartY
FillStyle
FlashEnabled
FlashOn
FlashSpeed
FlashTime
Font
Height
HelpContextID
Index
Left
LeftBorder
LinkErrorDisplay
LinkErrorNumber
LinkErrorString
LinkItem
LinkMode
LinkServer
LinkTip
LinkTipBackColor
LinkTipForeColor
LinkTipText
LinkTopic
Name
NotFilledColor
NotFilledTransparent
NumberOfDataValues
NumberOfSegments
NumbersColor
Picture
PictureStretch
PokeLength
PokeStartIndex
PolyBorderColor
RealPixels
RequestLength
RequestStartIndex
RightBorder
ScreenPriority
Shadow
ShadowOffsetX
ShadowOffsetY
StartValue
Symbol
TabIndex
TabStop
Tag
Top
TopBorder
TrailingZeros
UseInPoke
UseInRequest
Value
ValueBackColor
ValueColor
ValuePadDownLoad
ValueShadow
ValueShadowColor
ValueTransparent
ValueX
ValueY
Visible
WhatsThisHelpID
Width
WriteValue
XYPair

Events

Change
Click
Clicks
DblClick
DragDrop
DragOver
GotFocus
KeyDown
KeyPress
KeyUp
LinkError
LinkItemNotSupported
LinkItemSupported
LinkNotify
LinkOutOfMemory
LinkServerDisconnected
LinkUnableToConnectToServer
LostFocus
MouseDown
MouseMove
MouseUp
PokeCompleted
RequestCompleted

Methods

Container
DoPoke
DoRequest
Drag
LinkPoke
LinkRequest
Move
Object
Parent
SetFocus
ShowWhatsThis
ZOrder

Installing the OCX

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

System Requirements

Before you install RSVessel, 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 free hard disk drive space
- A 3.5’’ floppy drive
- VGA Graphics Card
- Microsoft Visual Basic 4.0, Microsoft Access, or other OLE control containers
- Microsoft Windows NT 3.51 or Microsoft Windows 95

Recommended Requirements

- IBM-compatible Pentium
- 16 MB of RAM or more
- 30 MB free hard disk drive space
- Color (800x600) or (1024 x 768) Monitor
- Microsoft Mouse compatible pointing device (mouse, trackball, touchscreen, etc.)
- Microsoft Visual Basic 4.0, Professional or Enterprise Edition, Microsoft Access, or other OLE control containers
- Microsoft Windows NT 3.51 or Microsoft Windows 95

RSTOOLS.WRI File

The RSTOOLS.WRI file lists any last minute changes to the RSVessel documentation, Help file and to the RSVessel control. To read the file, open the Windows Write application or double-click the RSTOOLS.WRI file in the file manager or Windows Explorer.

Running Setup

When you run the setup program, you will set a path for RSVessel.

To Start Setup:

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

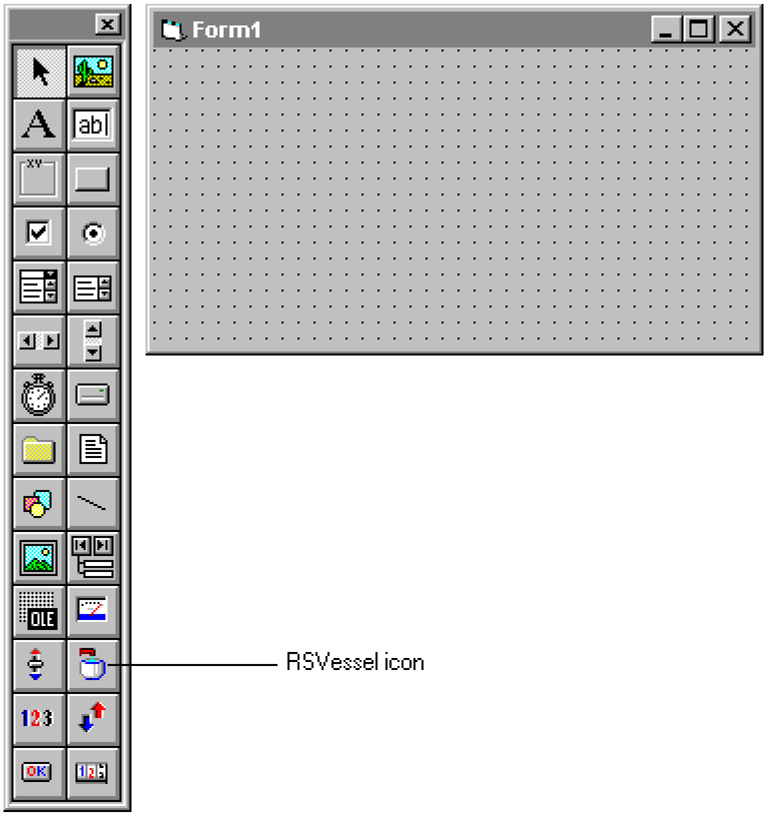
Learning To Use RSVessel

The RSVessel control requires very little programming to create full-featured applications.   Much of the functionality is available by setting standard properties.

Loading RSVessel (Adding RSVessel to your VB Project)

To use RSVessel in Visual Basic, you must add the RSVessel 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 RSVessel” and choose OK.



The RSVessel icon is added to the Visual Basic Toolbox.

Autoloading RSVessel

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

To configure Visual Basic to automatically load RSVessel:

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 RSVessel” 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 RSVessel’’ does not appear in the list of available custom controls, you may have to click the Browse button and manually select RSVSL32.OCX and RSVSLX32.DLL, which are located in the C:\WINDOWS\SYSTEM directory .When attempting to load RSVSLX32.DLL the message ‘‘Unable to load control from RSVSLX32.DLL’’ may appear. This occurs because this DLL contains supporting code for the RSVessel OCX and does not contain the actual control. Choose OK to continue.

Using Help

Comprehensive on-line help is available to assist you as you learn and use the RSVessel control. The complete RSVessel documentation is available through on-line help. In addition, you can receive context-sensitive help for properties, events and methods. The Help file is located in your C:\RSWKSHOP\RSTOOLBX directory.

To access the help contents page:

1. Click the RSVessel icon in the toolbox.
2. Press F1.

To access context-sensitive help for properties:

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

To access context-sensitive help for events:

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

Placing the RSVessel Control on a Form

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

1. Select the RSVessel 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 RSVessel control on the form.
4. When you release the mouse, the new RSVessel control is placed in the location you specified.

-OR-

1. Double click on the RSVessel tool in the Visual Basic toolbox.
2. An RSVessel control will be placed in the center of your form.

Distributing RSVessel 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 can ship the following files with your application:

File	**Description**
RSVSL32.OCX	Code for the RSVessel control
RSTOOL32.DLL	Common Code for the control

If the run time application requires features of the RSJunctionBox (enhanced communications, calculation engine) then in addition to the files above, an RSJunctionBox needs to be installed on the system. The RSJunctionBox (can be purchased separately) installs the following files in the WINDOWS\SYSTEM directory. These files may not be freely distributed and require a special activation key which is automatically installed during setup.

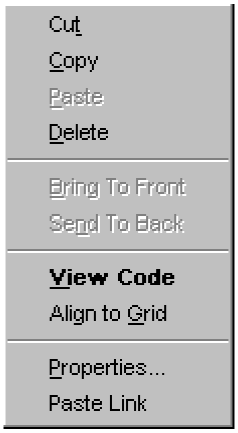
RSJBOX32.DLL	RSJunctionBox module
RSJBP32.DLL	RSJunctionBox Protection DLL
RSCALC32.DLL	Calculation engine for Read/Write expressions

Programming Tools

RSVessel provides over 100 custom and standard properties.   By setting these properties, you can perform a variety of tasks, such as setting colors and fill styles, establishing automatic DDE links, and binding to databases.

Visual Basic Floating Menu

Click the right mouse button anywhere within an RSVessel control to display the floating menu. Once the menu appears, use the left mouse button to select a menu item. Clicking on a menu item that contains three periods brings up additional dialog boxes for performing certain control functions. The “Properties…” menu item will bring up the Custom Property Tab dialog discussed in the next section.

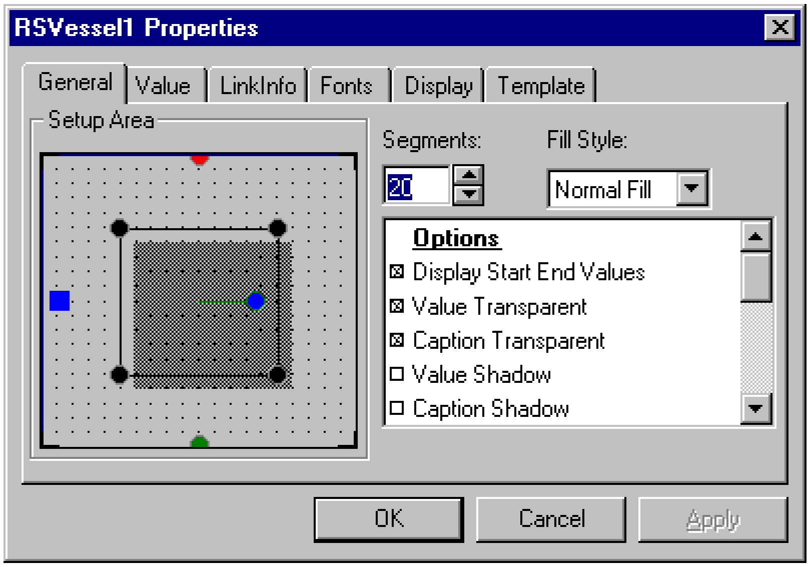


The following table describes the purpose of each item on the floating menu.

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 foreground.
Send To Back	Sends the selected item to the Background.
View Code	Displays the selected item’s code window.
Align to Grid	Aligns the control’s Top and Left properties to the form’s grid.
Properties	Displays the Custom Property Dialog.
Paste Link	Supports pasting a link from a valid DDE source.

Using Custom Property Tabs

The RSVessel 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 control type are grayed. The following illustration shows an example of the RSVessel OCX dialog box.



Segments: 20

Fill Style: Normal Fill

Options

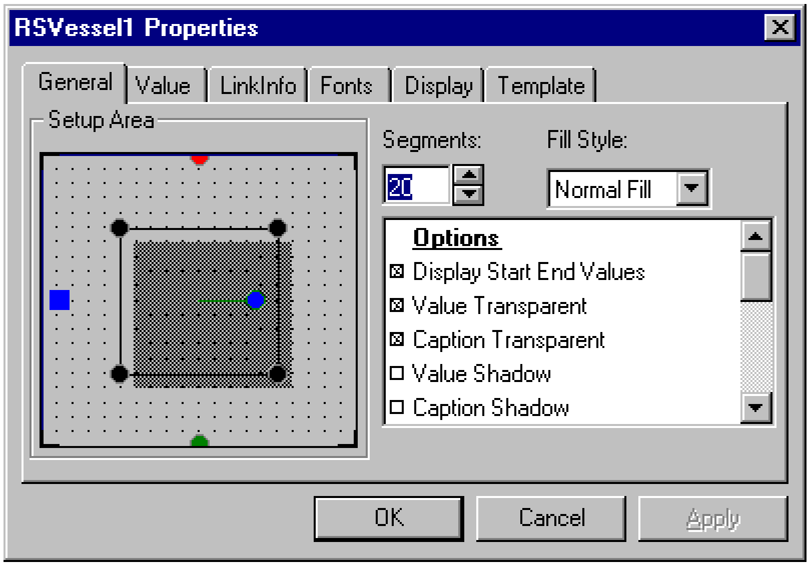
☒ Display Start End Values
☒ Value Transparent
☒ Caption Transparent
☐ Value Shadow
☐ Caption Shadow

Note

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

General Tab

The General tab allows you to make changes to the RSVessel control's more appearance-related properties, such as the RSVessel shape, Fill Style, whether or not to display the control's value, and many more.



RSVessel1 Properties

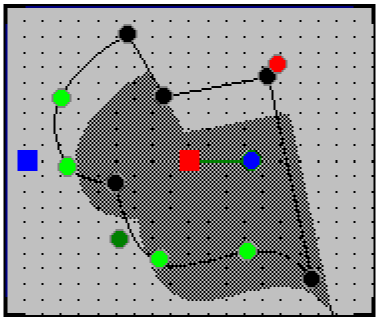
Segments: 20

Fill Style: Normal Fill

Options

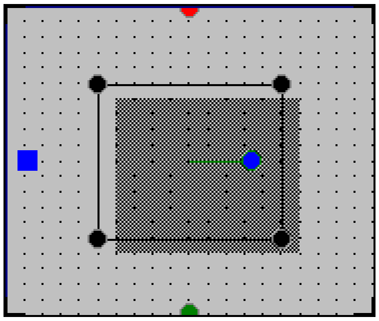
☒ Display Start End Values
☒ Value Transparent
☒ Caption Transparent
☐ Value Shadow
☐ Caption Shadow

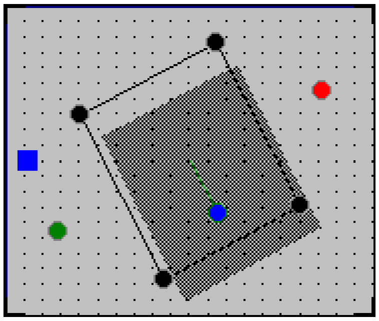
- **OCX Setup Area**.



Use this display to change the attributes of the control. The graphic image shown is made up of a series of nodes and lines connecting those nodes. To change the shape of the graphic image, simply click and drag the nodes within the rectangular setup area. Add nodes by right-clicking and dragging the mouse on top of an existing node; another node will then be created. Delete nodes by double-clicking on them. You can also add bezier curves to the shape by double-clicking on any line between two nodes. Two new green "handles" will appear, and the line that they are on can be "pulled" into a curve. These green "handles" can be moved to change the shape of the curve. These curves are actually a series of straight line segments. The "smoothness" of the curves is determined by the NumberOfSegments property.

The graphic image can be rotated from within the setup area by clicking and dragging the blue rotation handle, as shown below. Note that the fill direction rotates along with the image.



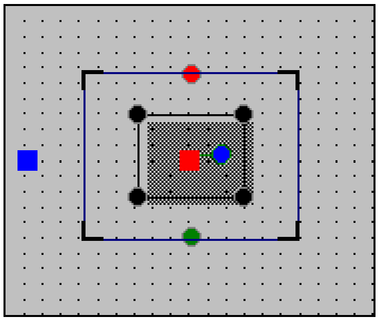


The RSVessel's fill direction is represented by the round green and red indicators in the setup area. The green indicator represents the point where the RSVessel control will begin filling, and the red indicator represents the point where the RSVessel control will stop filling. You can change the fill direction by clicking and dragging those indicators within the setup area. Note that these indicators can be located inside or outside of the image to give the effect of "empty" areas.

If the Value and Caption are being displayed on the RSVessel control, their positions can be adjusted by moving their placeholders within the shape editor. The Value display is represented by a blue square, and the Caption display is represented by a red square. The Value and Caption displays can be turned off and on with the right-click menu described on the next page.

The graphic'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 on the next page.

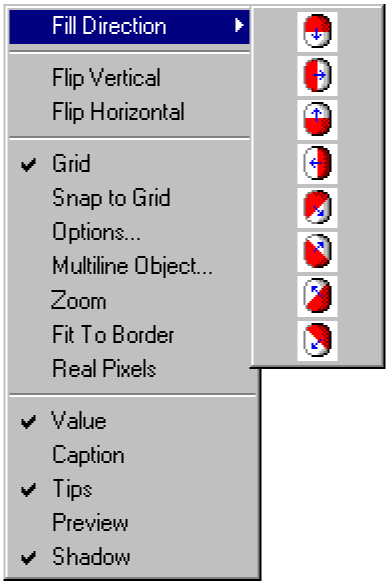
The size of the RSVessel fill area within the control's borders can be adjusted using the sizing handles. Each of the four corners of the Setup Area has a sizing handle, and by clicking and dragging those sizing handles you can adjust the TopBorder, BottomBorder, LeftBorder, and RightBorder properties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



Note

Fit To Border, which appears on the right-click menu and is described below, works within this sizable border, not the control's outer boundary.

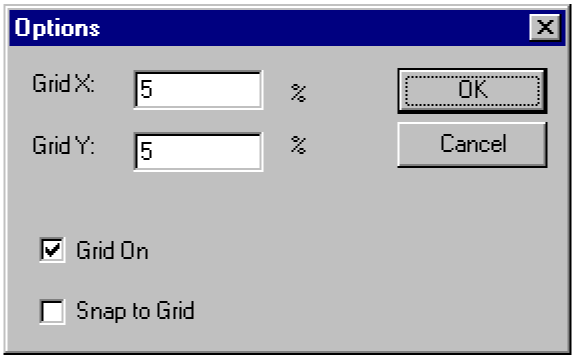
While in the setup area, click the right mouse button to bring up the floating menu.



From this floating menu you can change the Fill Direction for the RSVessel control by choosing one of the icons on the **Fill Direction** menu.

You can invert the RSVessel shape vertically or horizontally with **Flip Vertical** and **Flip Horizontal**.

The OCX setup area can display a grid for alignment; that grid is enabled / disabled with the **Grid** option on the floating menu. With the **Snap to Grid** option you can align the nodes of your graphic shape with the points on the grid. When this option is on, any node that is moved will be repositioned on one of the grid points displayed. The **Options…** option brings up a dialog box that allows you to set the density of the grid points, as well as enable / disable the **Grid** and **Snap to Grid** options.



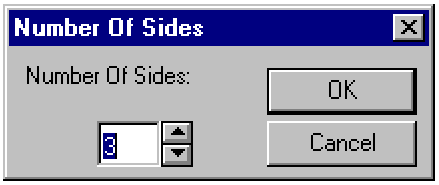
Grid X: 5 %

Grid Y: 5 %

☑ Grid On

☐ Snap to Grid

With the **Multiline Object…** option you can quickly set the number of sides that make up the graphic shape. Valid values are between 3 and 40.



Number Of Sides: 3

The **Zoom** option enlarges the OCX setup area to allow you to make more detailed adjustments to the graphic shape.

The **Fit To Border** option will adjust any outside nodes in your graphic shape to align with the outside border of the grid area, in order to make the fill area as large as possible.

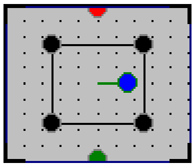
You can quickly enable / disable the **RealPixels** property from this menu. The **RealPixels** property determines whether the position properties will use relative positioning or absolute pixels. When set to True, the position properties will use absolute pixels - for example, if the RSVessel is 200 by 200 pixels, a point at position 50,50 will be drawn at pixel 50,50; when set to False, the position properties will use relative percentage positioning - in the same 200 by 200 pixel RSVessel the point 50,50 will be drawn at pixel 100,100.

From this menu you can also enable / disable the **Value** (DisplayValue), **Caption** (DisplayCaption), **Tips**

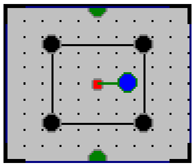
(LinkTip), or **Shadow** (Shadow) properties.

You can also display the **Preview** window, which allows you to view a floating display of your control.

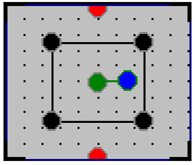
- **Segments Text Box.** The NumberOfSegments property value is changed here - the higher the number, the more rounded any bezier curves in the graphic shape will appear.
- **Fill Style Combo Box.** Set the FillStyle property here - choose from the following:



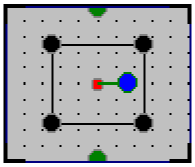
Normal - Green and red indicators set the start and end of the Fill Area.



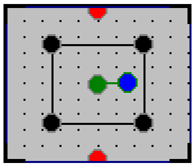
Mirror In - Two green indicators show the direction, and the single red indicator sets the “Mirror Line” where the split fill meets.



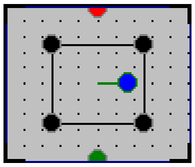
Mirror Out - Single green indicator sets the “Mirror Line”, while two red indicators set the direction. Notice that the red indicators move opposite of each other.



Center In - Two green indicators represent the outer diameter of the fill area, and the single red indicator is the center of the fill circle. Fill occurs in a circular pattern.



Center Out - Red indicator is the point from which the fill emanates and the two green indicators are the outside diameter of the fill area. Fill occurs in a circular pattern.



Area Fill - Allows narrow sections of the fill area to fill faster than wide areas. The rate of fill is proportional based on total area and Start and End values.

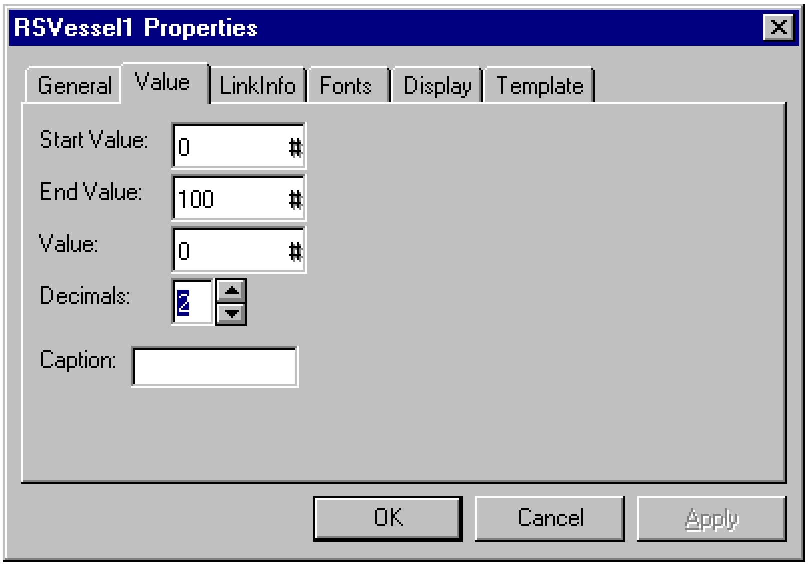
- **General Options List Box.** This list box allows you to set the general options for the control. You can enable or disable these properties from this list box by clicking in the appropriate check box. Following is a list of the options available:

Display Start End Values
Value Transparent
Caption Transparent
Value Shadow
Caption Shadow
Display Picture
Picture Stretch
Display Vessel Border
Shadow
Entry Pad Enabled
Border Beveled
Act Like Button
Trailing Zeros
Write Value
Draw Disabled Shadow
Not Filled Transparent

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

Value Tab

The Value tab allows you to view or edit the Value properties of the RSVessel control, such as StartValue and EndValue, Caption, and the number of DecimalPlaces.



Start Value: 0 #

End Value: 100 #

Value: 0 #

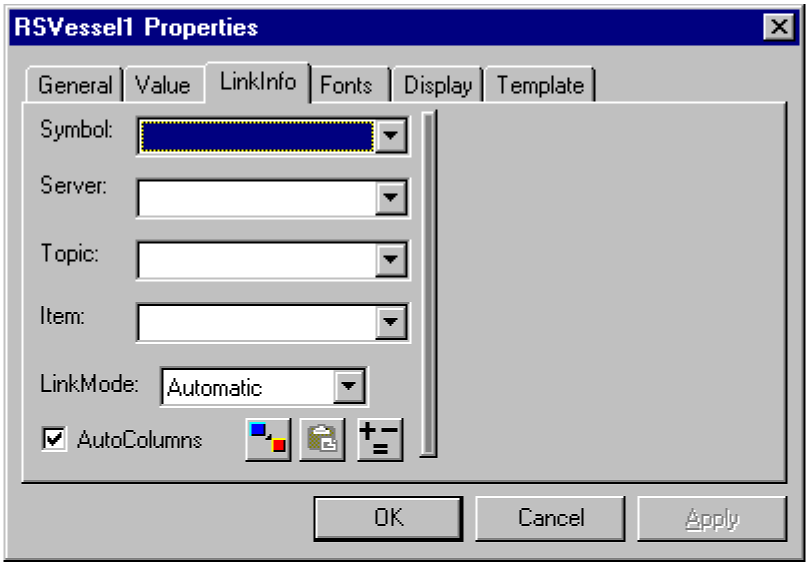
Decimals: 2

Caption:

- **Start Value**. Sets the minimum value for the value range. Either enter the number directly or click on the “#” sign to enter the number using the keypad.
- **End Value**. Sets the maximum value for the value range. Either enter the number directly or click on the “#” sign to enter the number using the keypad.
- **Value**. Sets the default for the Value property. Either enter the number directly or click on the “#” sign to enter the number using the keypad.
- **Decimals**. Sets the number of decimal places to be displayed for the Value property.
- **Caption**. Sets the default RSVessel caption.

LinkInfo Tab

The LinkInfo tab is where you can view or edit DDE link information for the RSVessel control.



LinkMode: Automatic

☑ AutoColumns

- **Symbol.** The Symbol combo box is used to choose a 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, its Server, Topic and Item combo boxes will be automatically filled in with the appropriate data. The *Manage Symbols* section below describes how to define a Symbol.
- **Server.** The Server combo box determines the application or LinkServer name to which the RSVessel control is linked. 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 RSVessel 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 name in the combo box.
- **Item.** The Item combo box determines the LinkItem portion of the data link string, which the RSVessel 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 name 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:

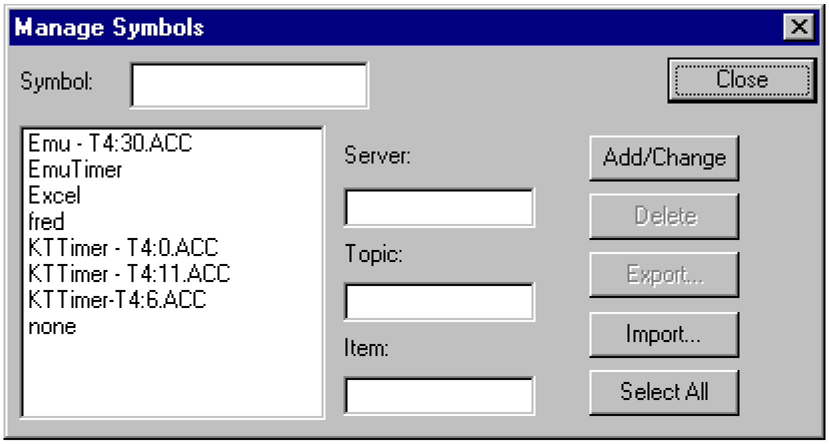
0 = None - No DDE connection is established between the control and **Server.**

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 another form (see below), which is used to define Symbols for DDE links to the RSVessel control.



Emu - T4:30.ACC
EmuTimer
Excel
fred
KTTimer - T4:0.ACC
KTTimer - T4:11.ACC
KTTimer-T4:6.ACC
none

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.

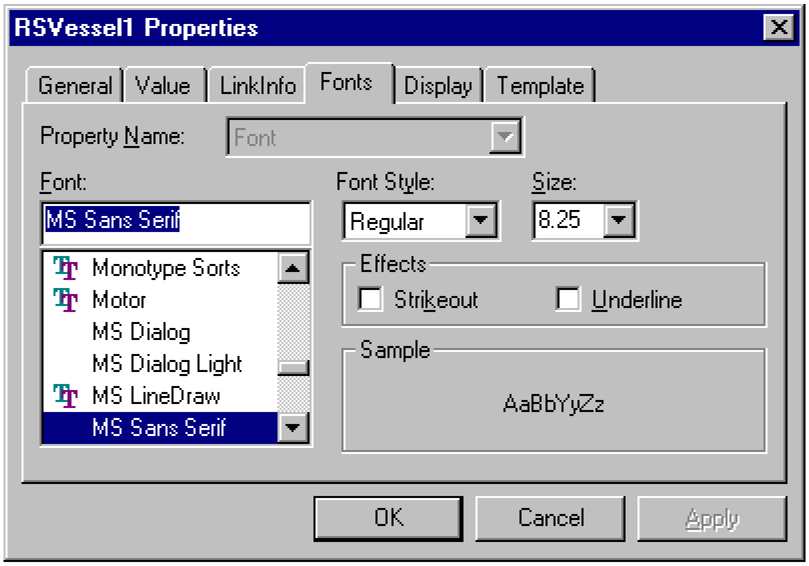
- **Paste Link.** This button is enabled when the user has copied specific DDE Link information from another application (for example, Microsoft Excel) to the clipboard. Pressing this button will then fill in the appropriate Server, Topic, and Item for the DDE Link. To copy the link information, go to the application to which the RSVessel control is to be linked and highlight the appropriate address, then copy it to the clipboard.
- **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. AdvanceDDE 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 RSVessel control's fonts. The font selected applies to the Start and End Value displays, as well as the Value and Caption displays.



Property Name: Font

Font: MS Sans Serif

Font Style: Regular

Size: 8.25

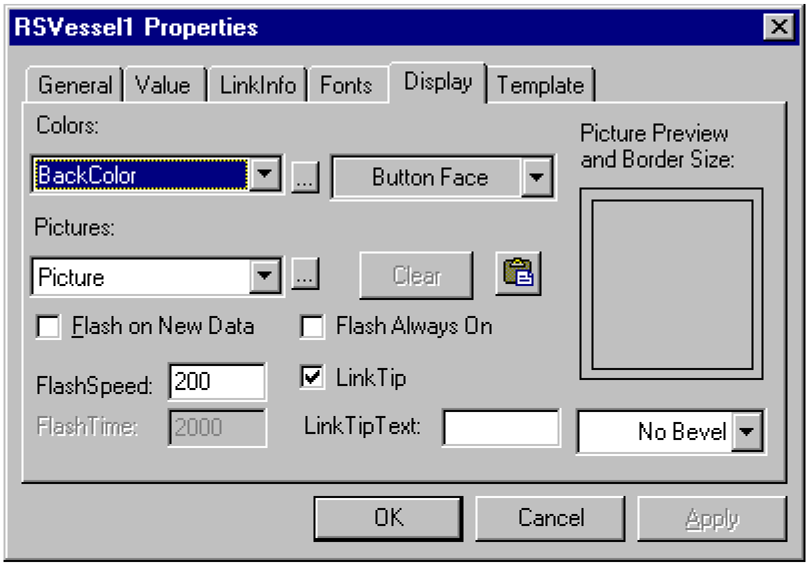
□ Strikeout

□ Underline

- **Property Name.** Selects the property of the control that you want to format. At this time only the **Font** property is available for RSVessel.
- **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 RSVessel control 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: BackColor | Button Face

Pictures: Picture

☐ Flash on New Data

☐ Flash Always On

FlashSpeed: 200

☑ LinkTip

FlashTime: 2000

LinkTipText:

No Bevel

- **Colors.** Setting of color properties involves selecting the color property name (for example, BackColor) and then choosing a color. Both the property name and the color can be selected from the drop-down combo boxes labeled *Colors*.

  Color properties that can be set using Display Tab for the RSVessel control are:

  BackColor
  BevelHighlight
  BevelShadow
  BorderColor
  BorderHighlight
  BorderInnerColor
  BorderShadow
  CaptionBackColor
  CaptionColor
  CaptionShadowColor
  FillColor
  LinkTipBackColor
  LinkTipForeColor
  NotFilledColor
  NumbersColor
  PolyBorderColor
  ValueBackColor
  ValueColor
  ValueShadowColor

  Select a color property by highlighting it in the drop-down combo box and then set the color you want to apply to that property from either the basic color palette (activated by clicking on the ellipsis button) or the drop-down color combo box. The drop-down color combo box 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, then use this drop-down combo box. The properties set to Windows desktop colors will automatically take up the new colors when you change your desktop's colors settings. If the project is compiled and made into an executable, then the executable's control colors would also inherit the desktop colors of the machine it is run on.
- **Picture.** You can assign a picture to the RSVessel control's Picture property with the Picture combo box. The Picture property allows accurate depiction of tanks and other containers by allowing you to place a bitmap in the background and then edit the fill area on top of that picture. Click on the ellipsis button to bring up a File Open dialog box. With this dialog box you can choose any *.bmp, *.wmf, or *.ico 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 the picture, simply click on the **Clear** button. In order for the Picture to be visible on the control, make sure that the DisplayPicture property is set to True.
- **Flash on New Data.** The FlashEnabled property is turned off and on with this check box. When turned on, the RSVessel control will flash when the RSVessel'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 RSVessel control will continually flash at the rate designated by the FlashSpeed property.
- **FlashSpeed.** Set the FlashSpeed property value by entering a number here. FlashSpeed is measured in milliseconds. FlashSpeed is the rate at which the RSVessel control will flash on and off.

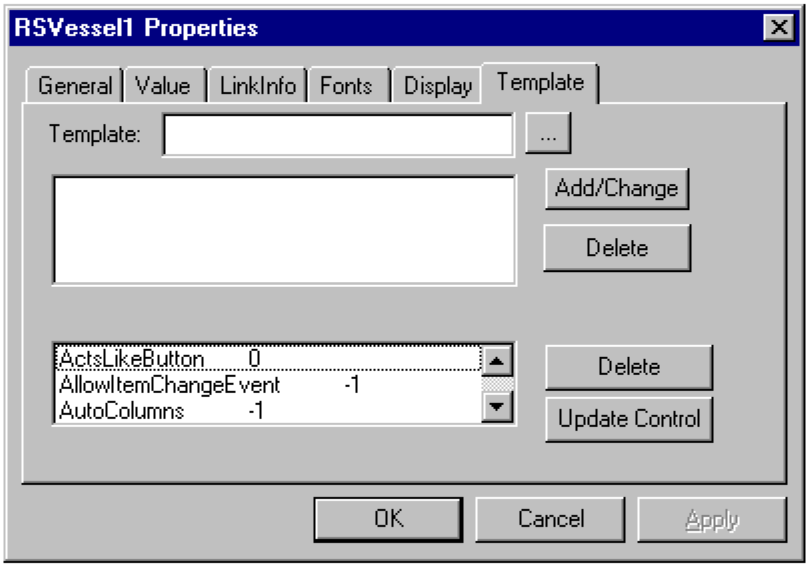
- **FlashTime.** Set the FlashTime property value by entering a number here. FlashTime is measured in milliseconds. FlashTime is the amount of time that the RSVessel control will continue to flash.
- **LinkTip.** RSVessel provides the ability to add tool tips for the control. The tool tip can be turned on by checking this check box. Text for the tool tip is entered in the **Link Tip Text** box below the check box. If no text is entered, the tool tip text defaults to the *Symbol* name being used, and if no symbol was used, to the DDE LinkItem details for the control.
- **Picture Preview and Border Size.** After choosing a bitmap for a picture property, a preview of the bitmap will appear here. The RSVessel bevel width can also be set with this box by clicking and dragging the mouse on one of the corners of the inner border.
- **Bevel Style.** This combo box allows you to select a bevel style for the RSVessel control. The available bevel styles are:

  No Bevel
  Indented
  Marble
  Bevel
  Thick
  Stripe

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

Template Tab

You can save the property settings of a control and apply them to other controls using templates. Once a template of properties has been saved, it can be applied to another RSVessel control and the property settings applicable to that control are changed as per the template. Using this custom property page tab, templates can be Added, Changed and Managed. The property templates are saved in a file with an ".rwc" extension. Each template is saved with a distinct name. A duplicate name will replace any template with the same name in that file.



ActsLikeButton 0
AllowItemChangeEvent -1
AutoColumns -1

- **Template**. This field displays the name of the template to be applied, added or changed. A template file can be selected by clicking on the ellipsis button ... next to the field. Available template 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 can be added or changes in an existing template can be saved by clicking on this button. The template is saved in the selected *.rwc file.
- **Delete Button**. Use this button to delete a template from the list. Only that template is removed from the file; all other templates in that file are left in place.
- **Properties.** This field, below the template list, shows the property names and their settings for the selected template.
- **Delete Property Button**. If you want to delete a property from the template, select the property name 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 custom property tabs or Visual Basic property list and then saved in templates. Property values cannot be set on this property page. A few sample templates are included with the RSVessel control that may be helpful in learning to use and apply templates.

Using RSVessel Custom Events

The RSVessel control comes with a set of eleven custom events that allow you to track and monitor actions performed on the control by users of your application. Events allow you to respond to users' actions and control the operation of the RSVessel control.

Change Event

You can use the **Change** event to create bargraphs, animation sequences, or copy data as it changes to other controls. The following code displays a Progress Bar using the dValue argument.

1. Place an RSVessel control and a ProgressBar control on your form.
2. Set the RSVessel **AllowChangeEvent** = True, **LinkServer** = “EXCEL”, **LinkTopic** = “[Book1]Sheet1”, **LinkItem** =“R1C1”, and **LinkMode** = 1-Automatic.
3. In the ProgressBar1 properties window set **Max** = 100 and **Min** = 0.
4. Type in the following code:

```
Private Sub RSVessel1_Change(ByVal Value As Double, ByVal
VesselIndex As Integer)

    ProgressBar1.Value = Value

End Sub
```

5. Run the program.
6. Go to EXCEL and change the value in Cell 1 of Row 1. The RSVessel value should change accordingly and the Progress Bar will show the value in graph form.

Note

RSVessel can also be linked to an array of cells. For example, try changing the ***LinkItem*** *to R1C1:R2C1. When you run the program, RSVessel will split into two Vessels, one showing the value from Row1 Col1, and the other showing the value from Row2 Col1. VesselIndex will now be the index of the particular Vessel control whose value is being changed. (Row1 Col1 will be VesselIndex 0; Row2 Col1 will be VesselIndex 1.)*

Clicks Event

You can use the **Clicks** event to fire code when the RSVessel control uses its ActsLikeButton property to act as a button. The **Clicks** event is only fired when its ActsLikeButton property is set to True. In the Clicks event, the “dValue” argument will contain the current RSVessel value, and in the case of a control array, the “iButtonIndex” argument will pass the index of the particular RSVessel control that was clicked on. The following code sets both the value of the particular RSVessel control in the array that is clicked on and its corresponding Excel block value to zero.

1. Place an RSVessel control on your form.
2. Start Microsoft Excel. In Row 1, Column 1 through Row 5, Column 1, enter five numeric values.
3. Minimize Excel and return to Visual Basic. Set the RSVessel **ActsLikeButton** = True, **LinkServer** = “Excel”, **LinkTopic** = “[Book1]Sheet1”, **LinkItem** = “R1C1:R5C1”, and **LinkMode** = “1 - Automatic”.
4. Type in the following code:

```
Private Sub RSVessel1_Clicks(ByVal dValue As Double, ByVal
iButtonIndex As Integer)

    RSVessel1.DataValue(iButtonIndex) = 0

End Sub
```

5. Run the program. You will see the RSVessel control split into five separate Vessels, each displaying a different value. If you bring up Excel and view both the worksheet and the Visual Basic project simultaneously, you will notice that each time you click on one of the RSVessel controls, the **Clicks** event is fired and its value changes to zero. The corresponding Excel block’s value will also change to zero. The RSVessel values can be reset by typing a new value into the Excel worksheet.

Note

There are two faster ways to set up the LinkServer, LinkTopic, and LinkItem in this example. Highlight the cells to be linked while in Excel and copy them to the clipboard. Then return to Visual Basic and right-click on the RSVessel control. On the floating menu that appears you will see a “Paste Link” option. Click on that option, and your Excel link will be pasted directly to the RSVessel control. The second method is similar - copy your Excel link to the clipboard, then return to Visual Basic and bring up the RSVessel custom properties page. On the LinkInfo tab, click on the 📋 *(Paste Link) button; this will paste in the Link Server, Topic, and Item directly from the clipboard.*

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.

```
Private Sub RSVessel1_LinkError(ByVal iRet As Integer, ByVal
ErrorString As String)
    Dim Msg As String
    Select Case iRet
        Case 3
            Msg = "Connection failed"
        Case 10
            Msg = "Out of Memory Error"
    End Select
    MsgBox Msg, 48, "Error Report"
End Sub
```

LinkItemNotSupported Event

The **LinkItemNotSupported** event occurs when RSVessel's LinkItem property has an incorrect format.

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

```
Private Sub RSVessel1_LinkItemNotSupported()

    Dim Msg As String

    Msg = "Link Item " & RSVessel1.LinkItem & " Is Not
    Supported"

    MsgBox Msg, 48, "Error Report"

End Sub
```

4. Start Microsoft Excel and run the program. The LinkItemNotSupported event will be fired and the message box will appear because "R0C1" is not a valid LinkItem value.

LinkItemSupported Event

The **LinkItemSupported** is fired when a DDE conversation is initiated and the specified LinkItem is supported by the server.

1. Place an RSVessel control on your form.
2. Set the RSVessel **AllowChangeEvent** = True, **LinkServer** = “EXCEL”, **LinkTopic** = “[Book1]Sheet1”, **LinkItem** =“R1C1” and **LinkMode** = 1-Automatic.
3. Type in the following code:

```
Private Sub RSVessel1_LinkItemSupported()
    Dim Msg As String
    Msg = "Link Item " & RSVessel1.LinkItem & " Is Supported"
    MsgBox Msg, 48, "Error Report"
End Sub
```

4. Start Microsoft Excel and run the program. Because R1C1 is a valid LinkItem, the **LinkItemSupported** event will be fired and the message box will appear.

LinkNotify Event

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

```
Private Sub RSVessel1_LinkNotify()
    If Screen.ActiveForm Is Me Then
        Picture1.LinkRequest   'Picture is on active form, so
        update
    Else
        NewDataFlag = True     'Assumed to be a module-level
        variable
    End If
End Sub
```

LinkOutOfMemory

The LinkOutOfMemory event occurs when the DDE client (RSVessel) exhausts its memory resources.

```
Private Sub RSVessel1_LinkOutOfMemory()
    Dim Msg As String
    Msg = "Out of memory to carry on link activities"
    MsgBox Msg, 48, "Link Error"
End Sub
```

LinkServerDisconnected Event

The **LinkServerDisconnected** event occurs when a DDE server is shut down during a DDE conversation.

1. Place an RSVessel control on your form.
2. Set the RSVessel **AllowChangeEvent** = True, **LinkServer** = “EXCEL”, **LinkTopic** = “[Book1]Sheet1”, **LinkItem** =“R1C1” and **LinkMode** = 1-Automatic.
3. Type in the following code:

```
Private Sub RSVessel1_LinkServerDisconnected()
    Dim Msg As String
    Msg = "Link Server Disconnected"
    MsgBox Msg, 48, "Error Report"
End Sub
```

4. Start Microsoft Excel and run the Visual Basic program.
5. While your program is running, close Book1 in Excel. Because the DDE server (Excel) has been shut down, the **LinkServerDisconnected** event is fired, and the message box appears.

LinkUnableToConnectToServer Event

The **LinkUnableToConnectToServer** event occurs when RSVessel attempts to connect to a DDE server that is unavailable.

1. Place an RSVessel control on your form.
2. Set the RSVessel **AllowChangeEvent** = True, **LinkServer** = “EXCEL”, **LinkTopic** = “[Book1]Sheet1”, **LinkItem** = “R1C1”, and **LinkMode** = 1 - Automatic.
3. Type in the following code:

```
Private Sub RSVessel1_LinkUnableToConnectToServer()
    Dim Msg As String
    Msg = "Link Unable to Connect to Server"
    MsgBox Msg, 48, "Link Error"
End Sub
```

4. Run the program without starting Microsoft Excel. Because Visual Basic is unable to connect to the Excel server, the **LinkUnableToConnectToServer** event is fired and the message box is displayed.

PokeCompleted Event

The **PokeCompleted** event occurs when RSVessel completes a DDE Poke.

1. Place an RSVessel control, a Command Button control, and a Slider control on your form.
2. Set the RSVessel **AllowChangeEvent** = True, **LinkServer** = "EXCEL", **LinkTopic** = "Sheet1", **LinkItem** = "R1C1", **LinkMode** = 2 - Manual, **UseInPoke** = True, and **DisplayCaption** = True.
3. Set the Command Button Caption = "Poke Data".
4. Set the Slider Max = 100, Min = 0, and TickFrequency = 10.
5. Type in the following code:

```
Private Sub Command1_Click()
    RSVessel1.LinkPoke
End Sub

Private Sub RSVessel1_PokeCompleted(ByVal iRet As Integer)
    RSVessel1.Caption = "Poke Completed"
End Sub

Private Sub Slider1_Change()
    RSVessel1.Value = Slider1.Value
End Sub
```

6. Start Microsoft Excel and run the program. Move the Slider knob, which will change the RSVessel value. After setting a value, click on the "Poke Data" button to poke the value to Excel. You will see the new value appear in Row 1, Column 1 of Sheet1, and RSVessel's caption will tell you when the Poke has completed successfully.

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

The **RequestCompleted** event occurs when RSVessel completes a DDE Request.

1. Place an RSVessel control and a Command Button control on your form.
2. Set the RSVessel **AllowChangeEvent** = True, **LinkServer** = “EXCEL”, **LinkTopic** = “Sheet1”, **LinkItem** = “R1C1”, **LinkMode** = 2 - Manual, and **UseInRequest** = True.
3. Set the Command Button Caption =   “Request Data”.
4. Type in the following code:

```
Private Sub Command1_Click()
    RSVessel1.LinkRequest
End Sub


Private Sub RSVessel1_RequestCompleted(ByVal iRet As Integer)
    Dim Msg As String
    Msg = "DDE Request completed"
    MsgBox Msg, 48, "Link Message"
End Sub
```

5. Start Microsoft Excel and run the program. Enter a value in Row 1, Column 1 in the Excel worksheet, and then click the “Request Data” button. You will see RSVessel’s value updated, and after the data request is completed the message box will appear, telling you that the **RequestCompleted** event has fired and that the Request was successful.

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.

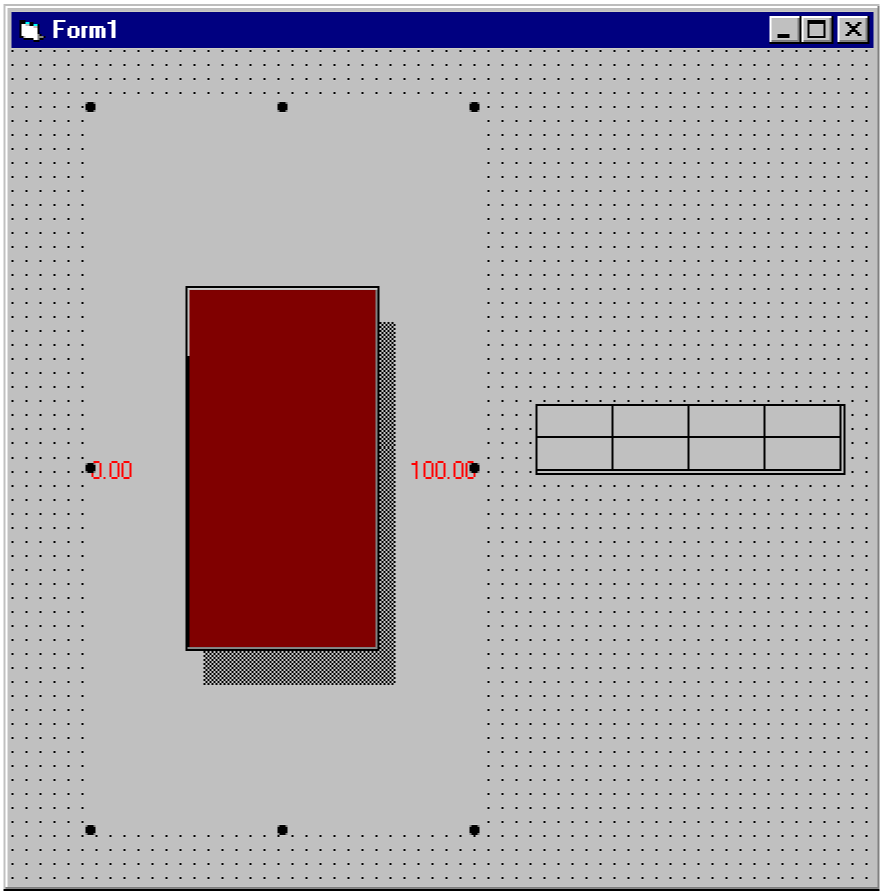
Samples Using RSVessel

The RSVessel control can be integrated easily with other controls on the market today. This chapter discusses using other common controls, including the Microsoft Grid control and the Visual Basic Data control, as well as working with the RSVessel's shape editor feature.

Using RSVessel with the Grid Control

The Grid control can be a very useful tool if you want to copy a block of data. The Grid control is generally used with an array of RSVessel controls or when block DDE is used. In the example below we will use the block DDE method.

1. Place an RSVessel control on a form and set **AllowChangeEvent** = True, **DisplayValue** = False, **FillEndX** = 75, **FillEndY** = 50, **FillStartX** = 25, and **FillStartY** = 50.
2. Set **LinkServer** = “Excel”, **LinkTopic** = “[Book1]Sheet1”, **LinkItem** = “R1C1:R8C1”, and **LinkMode** = “1 - Automatic”.
3. Place a Grid control on the form as shown in the picture below. Set Rows = 2, Cols = 4, FixedRows = 0, and FixedCols = 0.



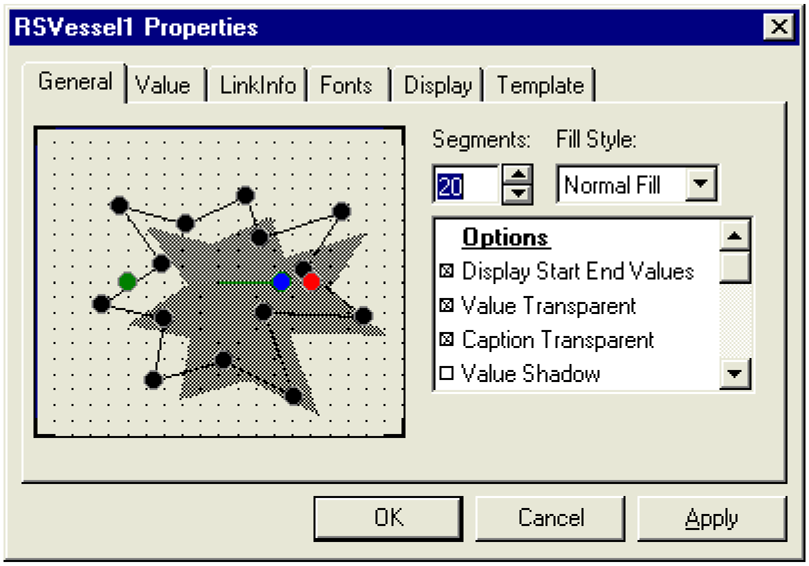
4. Type in the following code:

```
Private Sub RSVessel1_Change(ByVal Value As Double, ByVal
VesselIndex As Integer)
    Dim j As Integer
    For j = 1 To RSVessel1.NumberOfDataValues
        If j < 5 Then
            Grid1.Row = 0
            Grid1.Col = j - 1
        Else
            Grid1.Row = 1
            Grid1.Col = j - 5
        End If
        Grid1.Text = RSVessel1.DataValue(j - 1)
    Next j
End Sub
```

5. Start Microsoft Excel and run the Visual Basic program. As you type values into cells A1 - A8 in Excel Sheet 1, the values will be displayed in the RSVessel control and in the Grid control. Notice that because the RSVessel LinkItem (R1C1:R8C1) is an array of 8 data values, the RSVessel control splits itself into 8 separate controls, each displaying data from a different Excel cell.

Working With the RSVessel Shape Editor

1. The RSVessel control is an extremely visually versatile control. The following example will use the visual properties of RSVessel to create a “tank” control that might be used to display the fill level of a real tank. The example will also show how to use the RSVessel Shape Editor that is provided on the custom properties page.
2. Place an RSVessel control on a new form and set **AllowChangeEvent** = True and **DataUpdate** = “0 - Data Link”.
3. Set RSVessel’s **LinkServer** = “Excel”, **LinkTopic** = “[Book1]Sheet1”, **LinkItem** = “R1C1”, and **LinkMode** = “1 - Automatic”.
4. Click the right mouse button over the RSVessel control and bring up the Custom properties page by choosing “Properties…” on the floating menu that appears.
5. On the General tab of the Custom Properties page you will see a shape editor grid that displays the shape of the RSVessel control. Using your mouse, drag and drop the RSVessel’s nodes within that grid to change the shape of the RSVessel. To add an additional node, right-click on an existing node and a new one will be formed. To delete a node, double-click on it. Make your RSVessel shape look similar to the illustration below.



Segments: 20

Fill Style: Normal Fill

Options

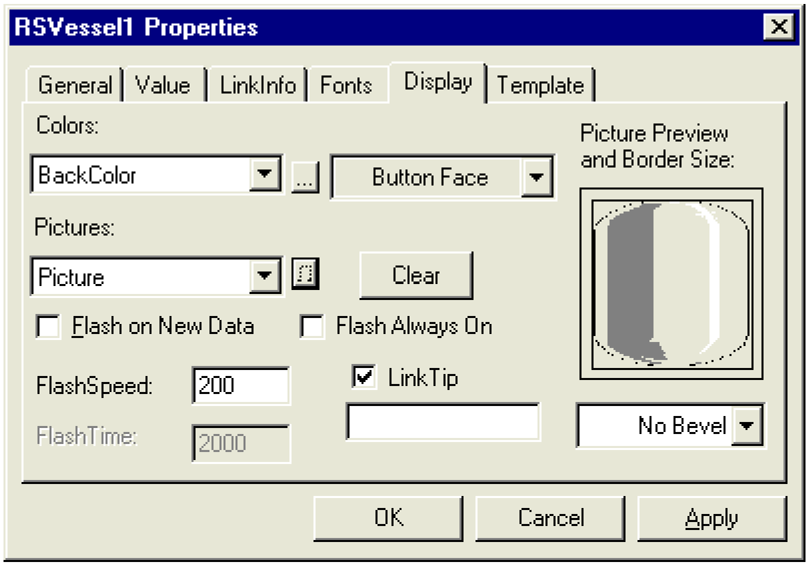
☒ Display Start End Values

☒ Value Transparent

☒ Caption Transparent

☐ Value Shadow

6. The next property to set is the **Picture** property. To set the Picture property, change to the Display tab on the Custom Properties page. Using the ellipsis button, select the “COOLTANK.BMP” bitmap from the C:\RSWKSHOP\RSTOOLBX\DEMO directory. The bitmap will appear in the background of the RSVessel control. The Picture Preview and Display tab should look like this:



Colors: BackColor ... Button Face

Pictures: Picture

☐ Flash on New Data

☐ Flash Always On

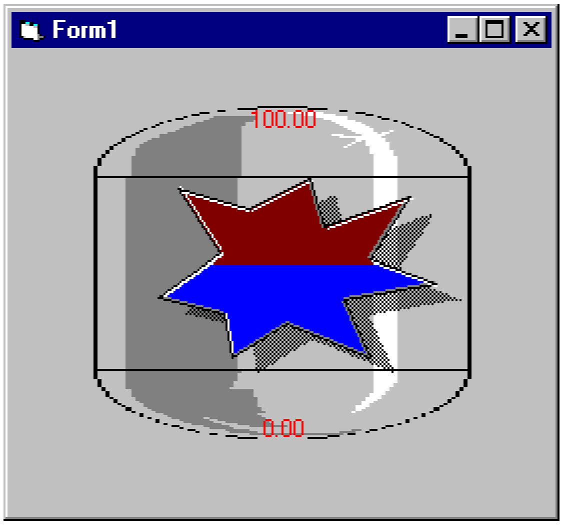
FlashSpeed: 200

FlashTime: 2000

☑ LinkTip

Picture Preview and Border Size: No Bevel

7. Hit the “OK” button to apply your changes. Start Microsoft Excel and run the Visual Basic program. As you change the value in cell A1 in the Excel worksheet you will see the RSVessel fill level going up and down, resembling a window into a filling tank. The RSVessel control should look similar to the illustration below.



Binding RSVessel to the Data Control

With the RSVessel control and Visual Basic's Data control, you can create an application to display, edit, and update (log) information to and 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 have to specify the database you would like to read the information from. Once you have decided on the database, you must load the RSVessel control on the form and set its properties to "bind" the control to Visual Basic's Data control. Depending upon the property settings you choose for DataSource, when you run this application you will be able to view data coming from your server, view data from a database, or log data to the database.

The RSVessel combined with Visual Basic's Data control give you seamless access to many standard database formats, including Microsoft Access, Btrieve, dBASE, Microsoft FoxPro, and Paradox.

The following steps will take you through building a data-aware application using the Visual Basic Data control and the RSVessel control. We will use the DBFLOG.DBF database that comes with RSVessel.

How to Use the RSVessel Control as a Database Tool

1. Select the RSVessel control in the Toolbox and draw an RSVessel on a new form.
2. Draw a Visual Basic Data control on the form. In the Data control's properties window, set DatabaseName = "C:\RSWKSHOP\RSTOOLBX\DEMO\", Connect = "dBASE IV;", and RecordSource = "DBFLOG".
3. Set the RSVessel control's **DataSource** = "Data1", **DataField** = "FIELD_1", **DataUpdate** = "2 - Data Source", and **EndValue** = 500.
4. Now run the program. The RSVessel control displays the data in the "FIELD_1" field based on which database record you display. You can use the Data control's arrow buttons to scroll through the records in the database.

Using the RSVessel Value Pad to Log to the Data Source

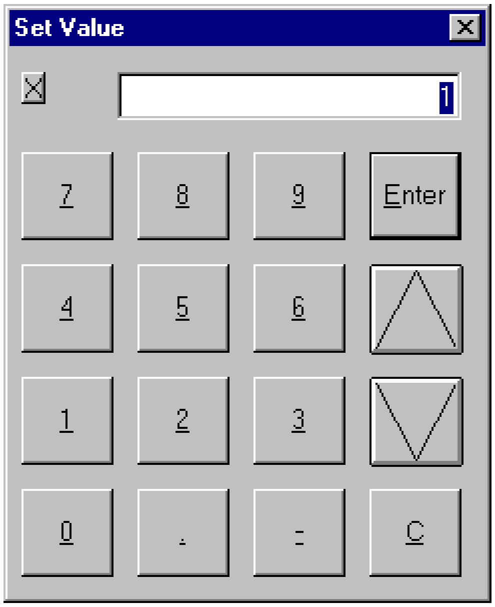
One of the biggest advantages of the RSVessel control is its ability to log the value displayed to the DataSource specified by the Visual Basic Data control . All of this can be accomplished with just a few lines of code.

Before you can log any information to the Data Source, you must have an existing database to log the information. This database must contain at least one entry or record. In the database you must specify a field name for the data point you would like to log. You may have several Field names, depending upon how many data points you have.

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

Follow the steps below to set up the RSVessel control for data logging. We will use the MEDREC.MDB sample database that comes with Microsoft Access, which is included with the RSVessel control.

1. Place a Visual Basic Data control on the form and set Connect = “Access”, DatabaseName = “C:\RSWKSHOP\RSTOOLBX\DEMO\MEDREC.MDB”, and RecordSource = “PATIENTS”.
2. Place an RSVessel control on the form and, using the Template custom properties tab, apply the “Pints of Blood” template to the control. Notice that the **ValuePadDownLoad** property is set to True and the **EndValue** property is set to 5 in this template. Set **DataSource** = “Data1”, **DataUpdate** = “2 = DataSource”, and **DataField** = “Private Blood Bank.
3. Add two Label controls to the form. Set Label1’s DataSource to “Data1” and DataField to “First Name”. Set Label2’s Data Source to “Data1” and DataField to “Last Name”.
4. Run the program. First notice that as you scroll through the database using the arrow buttons on the Data control you see the patients’ names and Private Blood Bank totals displayed in the two Labels and the RSVessel control. Now click on the RSVessel control. The Number Pad shown below appears and allows you to enter a new value for Pints of Blood.



If you try to enter a value that falls outside of the RSVessel control’s value range (StartValue to EndValue), you will get a Message Box prompting you to enter a number that falls within the range. Click on a value from 0 to 5 and then click on the “Enter” key to enter the new value. You can see your changes by scrolling through the database. Changes are written to the database as soon as you scroll to the next record.

Working With the XYPair Property

The XYPair property is what defines the shape that the RSVessel fill are will take. By setting the value of the XYPair property at run time you can change the appearance of the RSVessel control in a number of ways. The following example will use a Visual Basic Timer control, along with the XYPair property, to make an RSVessel control appear to “spin” in a manner similar to a propeller.

1. Place a Timer control and an RSVessel control on a new form. Set the Timer’s Interval property to 50.
2. Type in the following code:

```
Dim j As Integer     ‘ General declarations section.
Private Sub Timer1_Timer()
Select Case j
    Case 0
        RSVessel1.XYPair =
"50,40,25,25,39,50,25,75,50,60,75,75,60,50,75,25"
    Case 1
        RSVessel1.XYPair =
"55,40,30,20,39,45,20,64,50,60,69,79,60,50,80,35"
    Case 2
        RSVessel1.XYPair =
"60,40,45,14,39,40,15,50,45,60,55,85,60,55,85,50"
End Select
j = j + 1
If j = 3 Then j = 0
End Sub
```

3. Run the program. The star-shaped RSVessel control will appear to “spin”, although what is really changing is the value of the XYPair property. The best way to configure your values for XYPair is to use the Shape Editor on the custom properties tab to shape the RSVessel control, then copy the value for XYPair out of the Visual Basic properties window.

Property List

(About)
(Custom)
ActsLikeButton
AllowChangeEvent
AutoColumns
BackColor
BackStyle
BevelHeight
BevelHighlight
BevelShadow
BevelStyle
BevelWidth
BorderBeveled
BorderColor
BorderHighlight
BorderInner
BorderInnerColor
BorderShadow
BorderStyle
BorderWidth
BottomBorder
ButtonBorderWidth
Caption
CaptionBackColor
CaptionColor
CaptionShadow
CaptionShadowColor
CaptionTransparent
CaptionX
CaptionY
Clip
DataChanged
DataField
DataSource
DataUpdate
DataValue
DecimalPlaces
DisplayBorder
DisplayCaption
DisplayCaptionVertically
DisplayPicture
DisplayStartEndValues
DisplayValue
DragIcon
DragMode
DrawDisabledShadow
Enabled
EndValue
ExpressionForRead
ExpressionForWrite
FillColor
FillEndX
FillEndY
FillStartX

FillStartY
FillStyle
FlashEnabled
FlashOn
FlashSpeed
FlashTime
Font
Height
HelpContextID
Index
Left
LeftBorder
LinkErrorDisplay
LinkErrorNumber
LinkErrorString
LinkItem
LinkMode
LinkServer
LinkTip
LinkTipBackColor
LinkTipForeColor
LinkTipText
LinkTopic
Name
NotFilledColor
NotFilledTransparent
NumberOfDataValues
NumberOfSegments
NumbersColor
Picture
PictureStretch
PokeLength
PokeStartIndex
PolyBorderColor
RealPixels
RequestLength
RequestStartIndex
RightBorder
ScreenPriority
Shadow
ShadowOffsetX
ShadowOffsetY
StartValue
Symbol
TabIndex
TabStop
Tag
Top
TopBorder
TrailingZeros
UseInPoke
UseInRequest
Value
ValueBackColor
ValueColor
ValuePadDownLoad

ValueShadow
ValueShadowColor
ValueTransparent
ValueX
ValueY
Visible
WhatsThisHelpID
Width
WriteValue
XYPair

(About) Property

Description	Displays the *About RSVessel* dialog box and revision number. (Design time only.)

(Custom) Property

Description	Displays the RSVessel custom properties page. (Design time only.)

ActsLikeButton Property

Description Allows the RSVessel control to function as a button.

Custom General section, with a checkbox in the Options window.

Visual Basic [Form1.]Control.ActsLikeButton[=setting]

Remarks If both ActsLikeButton and ValuePadDownLoad are set to True, ValuePadDownLoad is ignored and the control acts like a button (does not display the value pad when clicked).

Data Type Boolean

DataValue Property

Description

Allows access to the element specified by an index in a DDE block array.

DisplayBorder Property

Description

Enables/disables all borders for the RSVessel fill area.

Custom

Enabled/disabled in the Options window of the General section of the custom properties page.

Visual Basic

[Form1.]RSVessel1.DisplayBorder[=setting]

Remarks

The following table lists the DisplayBorder settings.

Setting	Description
True	RSVessel will display a three-dimensional border around the fill area using the PolyBorderColor, BorderHighlight, and BorderShadow properties.
False	RSVessel will not display any of the border properties.

Data Type

Boolean

DisplayStartEndValues Property

Description

Determines whether RSVessel's start and end values will be displayed on the control.

Custom

Enabled / disabled in the Options window of the General section of the properties page.

Visual Basic

[Form1.]RSVessel1.DisplayStartEndValues[=setting]

Remarks

The following table lists the DisplayStartEndValues settings.

Setting	Description
True	StartValue and EndValue are displayed at the beginning and ending fill points of the RSVessel fill area. Beginning and ending fill points are as set with the FillStartX, FillStartY, FillEndX, and FillEndY properties.
False	StartValue and EndValue are not displayed on the RSVessel fill area.

Data Type

Integer

FillColor Property

Description

Sets the color for the portion of the RSVessel control that has been filled.

Custom

Color is set in the Display section of the properties page by choosing FillColor in the properties Combo Box and then selecting a color in the colors Combo Box.

Visual Basic

[Form1.]RSVessel1.FillColor[=setting]

Remarks

As the value of the RSVessel goes up, the portion of the fill area that is below that value will be represented with this color.

Data Type

Color

FillEndX Property

Description

Sets the horizontal coordinate for the point which determines the end of the fill area.

Custom

Set on the General section of the properties page by moving the red Fill Direction End control to the desired position on the fill shape. The X coordinate of the Fill Direction End control is the FillEndX value.

Visual Basic

[Form1.]RSVessel1.FillEndX[=setting]

Remarks

The FillEndX and FillEndY properties work together with the FillStartX and FillStartY properties to determine the points at which RSVessel's fill will begin and end.

Data Type

Integer

FillEndY Property

Description

Sets the vertical coordinate for the point which determines the end of the RSVessel fill area.

Custom

Set on the General section of the properties page by moving the red Fill Direction End control to the desired position on the fill shape. The Y coordinate of the Fill Direction End control is the FillEndY value.

Visual Basic

[Form1.]RSVessel1.FillEndY[=setting]

Remarks

The FillEndX and FillEndY properties work together with the FillStartX and FillStartY properties to determine the positions at which RSVessel's fill will begin and end.

Data Type

Integer

FillStartX Property

Description

Sets the horizontal coordinate for the point that determines the beginning of the fill area.

Custom

Set on the General section of the properties page by moving the green Fill Direction Begin control to the desired position on the fill shape. The X coordinate of the Fill Direction Begin control is the FillStartX value.

Visual Basic

[Form1.]RSVessel1.FillStartX[=setting]

Remarks

The FillStartX and FillStartY properties work together with the FillEndX and FillEndY properties to determine the positions at which RSVessel's fill will begin and end.

Data Type

Integer

FillStartY Property

Description

Sets the vertical coordinate for the point which determines the beginning of the fill area.

Custom

Set on the General section of the properties page by moving the green Fill Direction Begin control to the desired position on the fill shape. The Y coordinate of the Fill Direction Begin control is the FillStartY value.

Visual Basic

[Form1.]RSVessel1.FillStartY[=setting]

Remarks

The FillStartX and FillStartY properties work together with the FillEndX and FillEndY properties to determine the positions at which RSVessel's fill will begin and end.

Data Type

Integer

FillStyle Property

Description

Sets the filling style for the RSVessel fill shape.

Custom

Set on the General section of the properties page with the Fill Style Combo Box. Choices are Normal Fill, Mirror In, Mirror Out, Center In, Center Out, and Area Fill.

Visual Basic

[Form1.]RSVessel1.FillStyle[=setting]

Remarks

The following table lists the FillStyle settings.

Setting	Description
0	Normal Fill - fills from the Fill Direction Begin control to the Fill Direction End control.
1	Mirror In - fills from outer edges of fill area to center.
2	Mirror Out - fills from center of fill area to outer edges.
3	Center In - fills in a circular pattern from the edges of the fill area in to the center.
4	Center Out - fills in a circular pattern from the center to the edges of the fill area.
5	Area Fill - fills in a Normal style, but fills faster in thinner sections of the fill area and slower in the wider sections.

Data Type

Integer

NotFilledTransparent Property

Description Allows the portion of the RSVessel fill area that is not filled to be transparent.

Custom Enabled / disabled in the Options window of the General section of the properties page.

Visual Basic [Form1.]RSVessel1.NotFilledTransparent[=setting]

Remarks The following table lists the NotFilledTransparent settings.

Setting	Description
True	The portion of the RSVessel fill area that is not filled will not display a color, but instead the picture or background color that is behind the fill area.
False	The RSVessel NotFilledColor property setting will be displayed in the portion of the fill area that is not filled.

Data Type Boolean

PolyBorderColor Property

Description Sets the color for the border around the RSVessel fill area.

Custom Color is set in the Display section of the properties page by choosing PolyBorderColor in the properties Combo Box and then selecting a color in the colors Combo Box.

Visual Basic [Form1.]RSVessel1.PolyBorderColor[=setting]

Remarks Will only be displayed when the RSVessel's DisplayBorder property is set to True.

Data Type Color

RealPixels Property

Description

Determines whether the position properties will use relative positioning or absolute pixels.

Custom

Set on the General properties page on the right-click floating menu. When a checkmark appears next to the “RealPixels” menu item, the property value is True. Toggle on and off by clicking the left mouse button on the “RealPixels” menu choice.

Visual Basic

[Form1.]RSVessel1.RealPixels[=setting]

Remarks

The following table lists the RealPixels property settings.

Setting	Description
True	The position properties will use absolute pixels - for example, if the RSVessel is 200 by 200 pixels, a point at position 50,50 will be drawn at pixel 50,50.
False	The position properties will use relative percentage positioning - in the same 200 by 200 pixel RSVessel the point 50,50 will be drawn at pixel 100,100.

Data Type

Boolean

XYPair Property

Description

Sets the values for the points which, when put together with lines drawn between them, will make up the graphic shape for the fill area of the RSVessel control.

Custom

XYPair is set on the General section of the properties page by moving or adding/deleting nodes from the RSVessel graphic window. The comma-separated XY coordinates (0,0 to 100,100 starting with 0,0 at the upper left corner) of the nodes make up the XYPair value.

Visual Basic

[Form1.]RSVessel1.XYPair[=setting]

Remarks

A line is drawn between each pair of nodes, beginning and ending at the same node in the sequence, in order to determine the shape of the RSVessel fill area.

Data Type

String

Event List

Change
Click
Clicks
DblClick
DragDrop
DragOver
GotFocus
KeyDown
KeyPress
KeyUp
LinkError
LinkItemNotSupported
LinkItemSupported
LinkNotify
LinkOutOfMemory
LinkServerDisconnected
LinkUnableToConnectToServer
LostFocus
MouseDown
MouseMove
MouseUp
PokeCompleted
RequestCompleted

Method List

Container
DoPoke
DoRequest
Drag
LinkPoke
LinkRequest
Move
Object
Parent
SetFocus
ShowWhatsThis
ZOrder

INI File Used for RSVessel

RSVessel saves some of its information such as the default file names, etc. in the RSTOOLS.INI file located in the C:\WINDOWS directory. Only the following information should be changed or altered, and any other section should not be altered for proper operation of the OCX.

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

```
[cfg]
cfg = c:\windows\mytempls.rwc
```

where mytempls.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. An example of the above is:

```
[sym]
subtotal = Excel | [Book1]Sheet1 | R6C2
total = Excel | [Book1]Sheet1 | R8C2
```

Refer to Microsoft Visual Basic Help for more information on this property.

Enabled Property

Refer to Visual Basic Help on this property.

Height Property

Refer to Visual Basic Help on this property.

Left Property

Refer to Visual Basic Help on this property.

DragIcon Property

Refer to Visual Basic Help on this property.

DragMode Property

Refer to Visual Basic Help on this property.

HelpContextID Property

Refer to Visual Basic Help on this property.

Name Property

Refer to Visual Basic Help on this property.

Index Property

Refer to Visual Basic Help on this property.

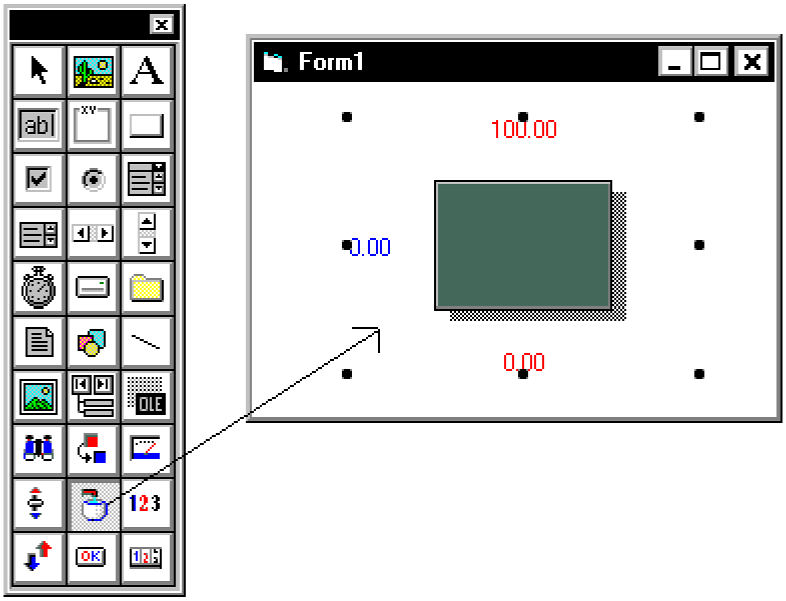
Introduction

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.

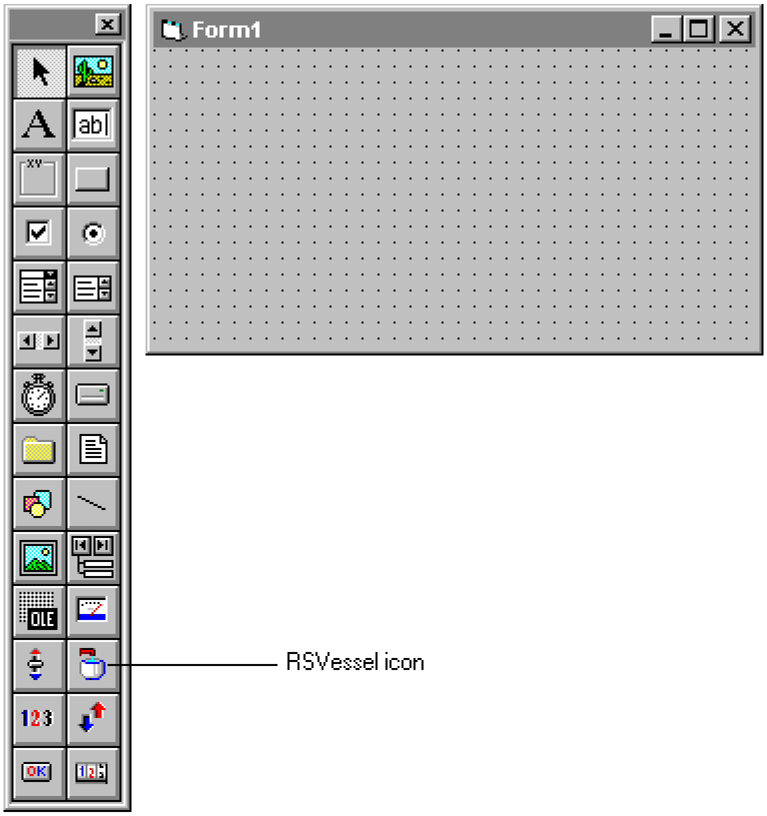
Learning To Use RSTools

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 RSTools XE "Loading RSToolbox" Controls (Adding RSTools to your VB Project)

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

1. Start Visual Basic.
2. From Visual Basic, select Tools/Custom Controls menu item.
3. 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 RSTools 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.)
2. 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:

1.OCX	Filename	Description
RSButton	RSBTN32.OCX	Rockwell Software Button
RSCompare	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.
4. 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 RSTools Applications XE "Distributing RSToolbox 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 you're 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

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:

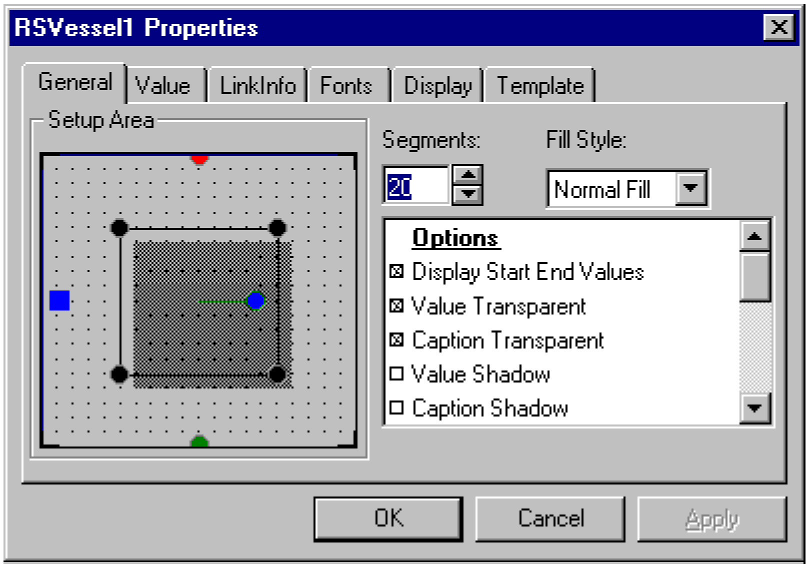
- LinkTip (True\False) - The LinkTip property allows you to turn it on or off during run time.
- LinkTipText - Displays text in a popup window next to the control. (A help balloon).
- LinkTipBackColor (any color value) - The LinkTipBackColor property allows you to change the background color of the LinkTip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

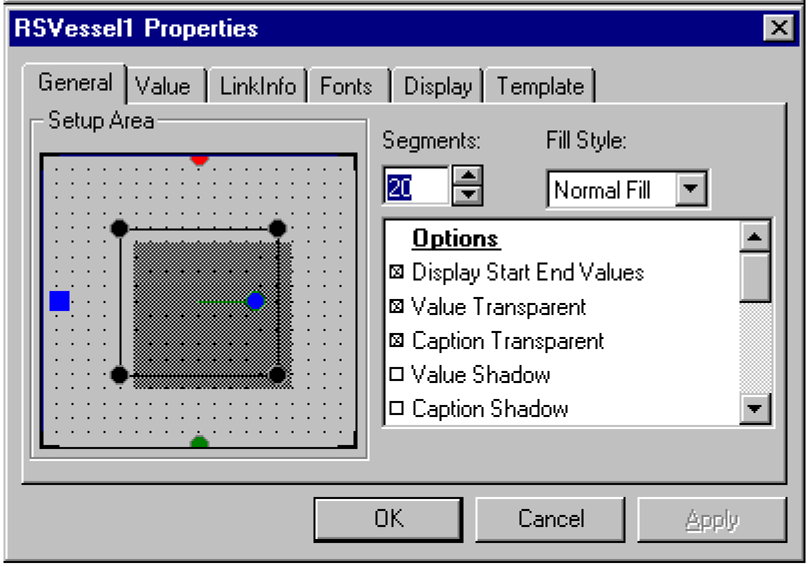


Segments: 20

Fill Style: Normal Fill

Options

☒ Display Start End Values
☒ Value Transparent
☒ Caption Transparent
☐ Value Shadow
☐ Caption Shadow

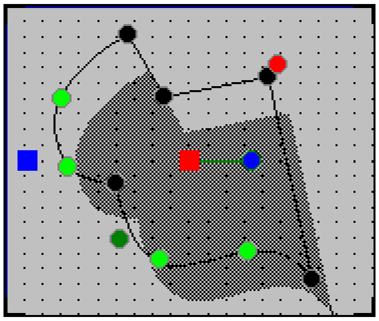


Segments: 20

Fill Style: Normal Fill

Options

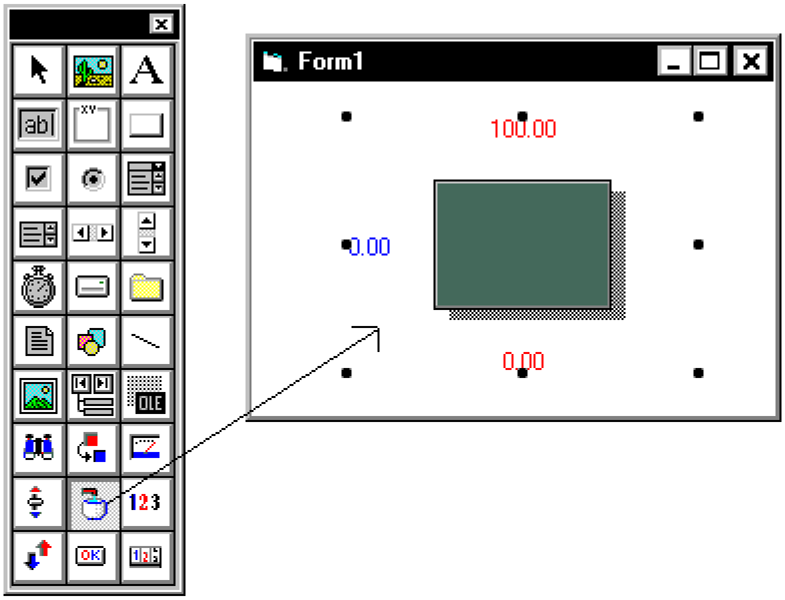
☒ Display Start End Values
☒ Value Transparent
☒ Caption Transparent
☐ Value Shadow
☐ Caption Shadow



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 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.

Application	DDE application name
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 it's 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.
4. Go back to VB, select or place a RSData control (or any RSTools control) on the form. Then select Paste Link.
5. 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.)
2. Set the **LinkServer** Property to “ICOMWDRV”, “EXCEL” or another DDE server.
3. Set the **LinkTopic** Property to “TESTSOL”, or to the name of the topic you configured.
4. Set the **LinkItem** Property to “N7:0”, or any valid item name.
5. Set the **LinkMode** Property to 1 - Automatic.
6. 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 RSData (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 RSData control that was specified in the **RequestStartIndex**.

The following example uses controls RSData1(0) through RSData1(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 RSData 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 RSData 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 RSData (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 RSData control that was specified in the **PokeStartIndex**.

The following example uses controls RSData1(0) through RSData1(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 RSData 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 RSData 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.
3. Set the **RequestLength** = 20, **RequestMode** = 0 - Start with Data, **RequestStartIndex** = 0 and **UseInRequest** = True.
4. 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”.
4. For Panel3D1, set the FloodColor = Green (&HC000&), FloodShowPct = True and FloodType = 1 - Left To Right.
5. For Panel3D2, set the FloodColor = Red (&HFF&), FloodShowPct = True and FloodType = 1 - Left To Right.
6. 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.

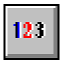
Contents

Quick Start

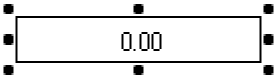
The following procedure gives you a brief overview of how to “bind” or use the RSData control (one of the RSTools controls) and Visual Basic’s data control in your application. Below we will use the COLOR.MDB sample database that comes with RSTools.

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

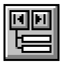
1. Select the RSData control in the toolbox and draw a RSData control on the form. The RSData control icon looks like this:



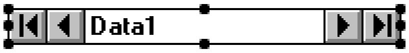
Click and drag the control on your form. It will look like a label control. The default name of the control is RSData1.



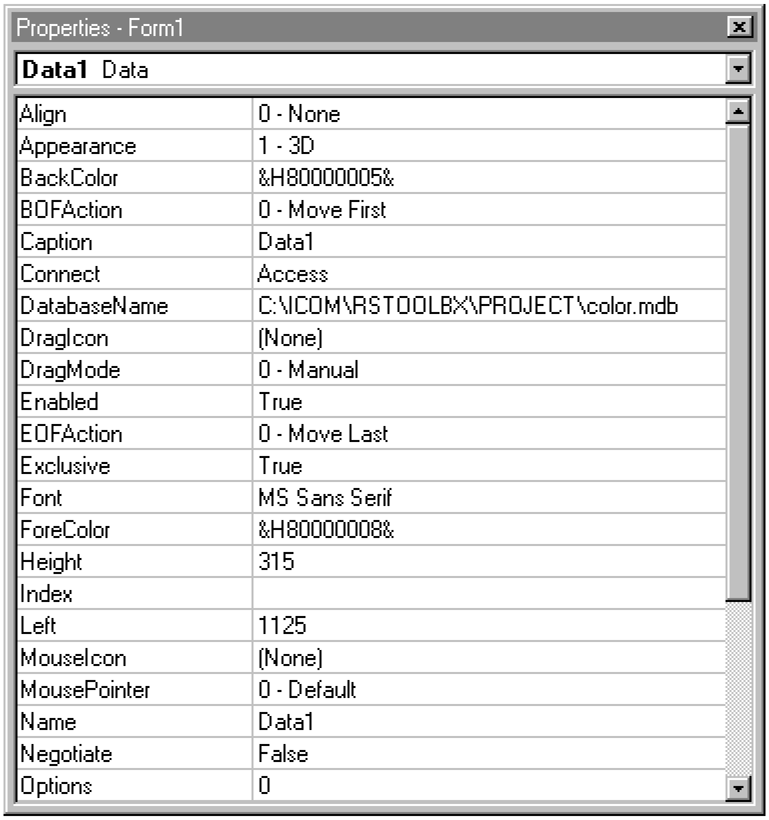
1. Select the Visual Basic Data control and draw a control on the form. The Data control icon looks like this:



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



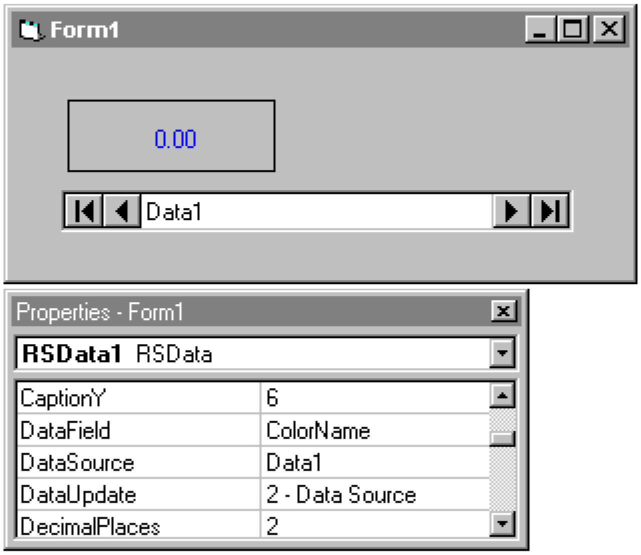
1. In the properties window for the Visual Basic Data control, set the DatabaseName property to the filename you want to connect to C:\RSWKSHOP\RSTOOLBX\DEMO\COLOR.DBF.



Data1 Data

Align	0 - None
Appearance	1 - 3D
BackColor	&H80000005&
BOFAction	0 - Move First
Caption	Data1
Connect	Access
DatabaseName	C:\ICOM\RSTOOLBX\PROJECT\color.mdb
DragIcon	(None)
DragMode	0 - Manual
Enabled	True
EOFAction	0 - Move Last
Exclusive	True
Font	MS Sans Serif
ForeColor	&H80000008&
Height	315
Index	
Left	1125
MouseIcon	(None)
MousePointer	0 - Default
Name	Data1
Negotiate	False
Options	0

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.



CaptionY	6
DataField	ColorName
DataSource	Data1
DataUpdate	2 - Data Source
DecimalPlaces	2

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

Applies To

RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel

Description

Creates the correct number of columns to be displayed based on the LinkItem.

Custom

No access via custom property page.

Visual Basic

*objec*___t___**.AutoColumns[=**_setting %_**]**

Remarks

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.

Data Type

Integer

BackColor Property

Applies To

RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel

Description

Sets the background color for the control.

Custom

Display section.

Visual Basic

object.**BackColor[**=*setting %***]**

Settings

Visual Basic uses the Microsoft Windows operating environment red-green-blue (RGB) color scheme. The settings for color are:

Setting	Description
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.

Data Type

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	*object*.**BackStyle[=***setting* **%]**

Settings

Settings	Description
0	Sets the control to transparent.
1	Sets the control to opaque.

Data Type Integer

BevelHeight Property

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

Description Sets the height of the top and bottom beveled edges around the control.

Custom No access via custom property page.

Visual Basic *object*.**BevelHeight[=**_setting %_**]**

Data Type Integer

BevelHighlight Property

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

Description Sets the highlight color of the control's beveled border.

Custom Display section.

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

Remarks 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.

Data Type Color

BevelShadow Property

Applies To

RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel

Description

Sets the shadow color of the control's beveled border.

Custom

Display section.

Visual Basic

object.**BevelShadow[**=*setting %***]**

Remarks

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.

Data Type

Color

BevelStyle Property

Applies To

RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel

Description

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.

Custom

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:

Settings	Description
0	None
1	Indented
2	Marble
3	Beveled
4	Thick
5	Stripe

Visual Basic

object.**BevelStyle[=***setting %***]**

Remarks

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.

Data Type

Integer

BevelWidth Property

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

Description
Sets the width of the left and right beveled edges around the control.

Custom
No access via custom property page.

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

Data Type
Integer

BorderBeveled Property

Applies To

RSCompare, RSVessel

Description

Toggles display of BorderHighlight and BorderShadow.

Custom

Adjusted in the General section of the custom properties page with a checkbox in the Options window.

Visual Basic

object.**BorderBeveled[=**_setting %_**]**

Remarks

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.

Data Type

Integer

BorderColor Property

Applies To

RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel

Description

Determines the color of the border around the control object.

Custom

Display section.

Visual Basic

*object***.BorderColor[=***setting %***]**

Remarks

The BorderColor property can be set to any color in the palette or to a hex integer value that represents a color.

Data Type

Color

BorderHighlight Property

Applies To

RSCompare, RSVessel

Description

Sets the Highlight color for the three-dimensional border of the applicable controls' graphic shapes.

Custom

Display section.

Visual Basic

object.**BorderHighlight[**=*setting %***]**

Remarks

For the RSCompare control, this property sets the BorderHighlight color for all of its states - Up, Equal, and Down.

Data Type

Color

BorderInner Property

Applies To

RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel

Description

Determines if the inner border is displayed around the control object.

Custom

No access via custom property page.

Visual Basic

object.**BorderInner[** = *setting* **%]**

Remarks

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.

Data Type

Integer

BorderInnerColor Property

Applies To

RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel

Description

Determines the color of the inner border around the control object.

Custom

Display section.

Visual Basic

object.**BorderInnerColor[=**_setting %_**]**

Remarks

The BorderInnerColor property can be set to any color in the palette or to a hex integer value that represents a color.

Data Type

Color

BorderShadow Property

Applies To

RSCompare, RSVessel

Description

Sets the Shadow color for the three-dimensional border of the applicable controls' graphic shapes.

Custom

Display section.

Visual Basic

object.**BorderShadow[=***setting %***]**

Remarks

For the RSCompare control, this property sets the BorderShadow color for all of its states - Up, Equal, and Down.

Data Type

Color

BorderStyle Property

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

Description Determines the type of border displayed around the control object.

Custom No access via custom property page.

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

Remarks There are two border styles available: 0 = None and 1 = Fixed Single.

Data Type Integer

BorderWidth Property

Applies To

RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel

Description

Determines the width of the border displayed around the control object.

Custom

No access via custom property page.

Visual Basic

object.**BorderWidth[=***setting* **%]**

Remarks

The BorderStyle property needs to be set to 1 (Fixed Single) in order to make changes in BorderWidth visible.

Data Type

Integer

BottomBorder Property

Applies To

RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel

Description

Sets the distance between bottom external border of the control and bottom of the control.

Custom

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.

Visual Basic

object.**BottomBorder[=***setting* **%]**

Remarks

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.

Data Type

Integer

ButtonBorderWidth Property

Applies To

RSCompare, RSVessel

Description

Sets the width of the BorderHighlight and BorderShadow properties for applicable controls.

Custom

No access via custom property page.

Visual Basic

object.**ButtonBorderWidth[**=*setting %*]

Remarks

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.

Data Type

Integer

ButtonFaceColor Property

Applies To

RSButton, RSSlider, RSWheel

Description

Sets the color for the face of the control's buttons.

Custom

Display section.

Visual Basic

*object***.ButtonFaceColor[=***setting %***]**

Remarks

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.

Data Type

Color

ButtonHighlight Property

Applies To

RSButton, RSSlider, RSWheel

Description

Sets the highlight color for the control's three-dimensional button(s) effect.

Custom

Display section.

Visual Basic

*object***.ButtonHighlight[**=*setting* %**]**

Remarks

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.

Data Type

Color

ButtonShadow Property

Applies To	RSButton, RSSlider, RSWheel
Description	Sets the shadow color for the control's three-dimensional button(s) effect.
Custom	Display section.
Visual Basic	*object***.ButtonShadow[**=*setting %***]**
Remarks	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.
Data Type	Color

Caption Property

Applies To

RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel

Description

Sets the caption to be displayed on the control.

Custom

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.

Visual Basic

*object***.Caption[=***setting %***]**

Remarks

The DisplayCaption property must be set to True in order for the Caption to be shown.

Data Type

String

CaptionBackColor Property

Applies To

RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel

Description

Sets the background color of the caption for the control.

Custom

Display section.

Visual Basic

object.**CaptionBackColor[=**_setting %_**]**

Remarks

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.

Data Type

Color

CaptionColor Property

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

Description Sets the color of the caption text for the control.

Custom Display section.

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

Remarks 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.

Data Type Color

CaptionShadow Property

Applies To

RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel

Description

Enables/disables the shadow behind the caption for the control.

Custom

Adjusted in the General section of the custom properties page with a checkbox in the Options window.

Visual Basic

object.**CaptionShadow [**=*setting %*]

Remarks

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.

Data Type

Integer

CaptionShadowColor Property

Applies To

RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel

Description

Sets the color of the caption's shadow for the control.

Custom

Display section.

Visual Basic

*object***.CaptionShadowColor[**=*setting %***]**

Remarks

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.

Data Type

Color

CaptionTransparent Property

Applies To

RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel

Description

Enables/disables display of the CaptionBackColor.

Custom:

Adjusted in the General section of the custom properties page with a checkbox in the Options window.

Visual Basic

object**.CaptionTransparent[=**_setting %_**]**

Remarks

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.

Data Type

Integer

CaptionX Property

Applies To

RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel

Description

Sets the horizontal position of the caption with respect to the left edge of the control object.

Custom

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.

Visual Basic

object.**CaptionX [=***setting %***]**

Remarks

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.

Data Type

Integer

CaptionY Property

Applies To

RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel

Description

Sets the vertical position of the caption with respect to the top edge of the control object.

Custom

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.

Visual Basic

*object***.Control.CaptionY[=** *setting %***]**

Remarks

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.

Data Type

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

*object.***Clip** *[= string]*

The Clip property syntax has these parts:

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

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:

```
RSGauge1.Clip = "231" & Chr(9) & "400" & Chr(13) & "278" & Chr(9) & "58"
```

Compiling a string and setting a control’s **Clip** property is an excellent way to programmatically use a single instance of a control to display multiple values.

The clip property works in a similar fashion for the other RSTool controls.

Data Type

String

DataChanged Property

Description

Returns or sets a value indicating that RSData in a control has changed by some process other than by retreiving RSData from the current record.

Visual Basic

object.*Control.***RSDataChanged***[=setting]*

Remarks

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.

Applies To

RSButton, RSCompare, RSData, RSGauge, RSSlider, RSWheel, RSVessel

DataField Property

Applies To

RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel

Description

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.

Custom

No access via custom property page.

Visual Basic

object.**DataField[**=*setting %*]

Remarks

The DataField, DataSource, and DataUpdate properties work together with the Visual Basic Data control to bind the RSTools control to a database.

Data Type

String

DataSource Property

Applies To

RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel

Description

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

Custom

No access via custom property page.

Visual Basic

object.**DataSource[**=*setting %*]

Remarks

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.

Data Type

String

DataUpdate Property

Applies To

RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel

Description

Determines the primary source of displayed data as well as which data triggers a Change event.

Visual Basic

object.**DataUpdate[=**_setting %_**]**

Remarks

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.

Data Type

Integer

DataValue(Index) Property

Applies To

RSButton, RSCompare, RSData, RSGauge, RSSlider, RSWheel, RSVessel

Description

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.

Visual Basic

*object***.DataValue(***item number***)**

Remarks

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)).

Data Type

Integer

DecimalPlaces Property

Applies To

RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel

Description

Determines the number of decimal places that will be shown when the value is displayed on the control.

Custom

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.

Visual Basic

*object***.DecimalPlaces[=***setting***]**

Remarks

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 **Value** and **DataValue(n)** properties whether visible or not.

Data Type

Integer

DisplayBorder Property

Applies To

RSCompare, RSVessel

Description

Enables/disables all borders for the control's graphic shape.

Custom

Enabled/disabled in the Options window of the General section of the properties page.

Visual Basic

Object.**DisplayBorder[**=*setting***]**

Remarks

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 **DownBorderColor**, **EqualBorderColor**, **UpBorderColor**, **BorderHighlight**, and **BorderShadow** 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 **PolyBorderColor**, **BorderHighlight**, and **BorderShadow** properties; when set to False none of those properties will be displayed.

Data Type

Integer

DisplayCaption Property

Applies To

RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel

Description

Determines if the caption will be displayed on the control.

Custom

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.

Visual Basic

object.**DisplayCaption[**=*setting %***]**

Remarks

When set to True the Caption will be displayed; when set to False the Caption will not be displayed.

Data Type

Integer

DisplayCaptionVertically Property

Applies To

RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel

Description

Determines if the caption will be displayed vertically on the control.

Custom

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.

Visual Basic

*obect***.DisplayCaptionVertically[=***setting***]**

Remarks

When set to True, the caption will be displayed vertically; when set to False, the caption will be displayed horizontally.

Data Type

Integer

DisplayPicture Property

Applies To

RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel

Description

Determines if a picture will be displayed on the control.

Custom

Adjusted in the General section of the custom properties page with a checkbox in the Options window.

Visual Basic

*object***.DisplayPicture[=***setting %***]**

Remarks

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.

Data Type

Integer

DisplayValue Property

Applies To

RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel

Description

Determines if the control's current value will be displayed.

Custom

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.

Visual Basic

object.**DisplayValue[**=*setting %***]**

Remarks

When set to True, the value will be displayed; when set to False the value will not be displayed.

Data Type

Integer

DrawDisabledShadow Property

Applies To

RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel

Description

Determines if a shadow will be displayed over the entire control when it is disabled.

Custom

Adjusted in the General section of the custom properties page with a checkbox in the Options window.

Visual Basic

object.**DrawDisabledShadow[=**_setting_ **%]**

Remarks

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.

Data Type

Integer

EndValue Property

Applies To

RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel

Description

Sets the maximum for the value range of the control.

Custom

Adjusted in the "End Value" text box in the Value section of the custom properties page.

Visual Basic

object.**EndValue[**=*setting %***]**

Remarks

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.

Data Type

Double

ExpressionForRead Property

Applies To

RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel

Description

Determines the mathematical expression that will be performed on the link item value when the control reads that value.

Custom

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.

Visual Basic

object.**ExpressionForRead[**=*setting %***]**

Remarks

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.

Optional Calculation/Math module required, RSCALC32.DLL.

Data Type

String

ExpressionForWrite Property

Applies To

RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel

Description

Determines the mathematical expression that will be performed on the link item when the control writes that value.

Custom

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.

Visual Basic

object.**ExpressionForWrite[=***setting* **%]**

Remarks

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.

Optional Calculation/Math module required, RSCALC32.DLL.

Data Type

String

FaceBorderColor Property

Applies To	RSGauge, RSSlider, RSWheel
Description	Sets the color of the face border for the control.
Custom	Display section.
Visual Basic	*object***.FaceBorderColor[=***color***]**
Remarks	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.
Data Type	Color

FaceColor Property

Applies To

RSGauge, RSSlider, RSWheel

Description

Sets the color of the face for the control.

Custom

Display section.

Visual Basic

*object***.FaceColor[=***setting %***]**

Remarks

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.

Data Type

Color

FillColor Property

Applies To

RSGauge, RSVessel

Description

Determines the color of the filled area on the control.

Custom

Display section.

Visual Basic

object.**FillColor[=***setting %***]**

Remarks

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).

Data Type

Color

FlashEnabled Property

Applies To

RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel

Description

Determines if the control will flash when its value changes.

Custom

Enabled / disabled in the Display section of the properties page with the "Flash on New Data" check box.

Visual Basic

*object***.FlashEnabled[=**_setting %_**]**

Data Type

Integer

FlashOn Property

Applies To

RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel

Description

Determines if the control will continuously flash from visible to invisible.

Custom

Enabled / disabled in the Display section of the properties page with the "Flash Always On" check box.

Visual Basic

object.**FlashOn [=** *setting %*]

Remarks

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.

Data Type

Integer

FlashSpeed Property

Applies To

RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel

Description

Determines the rate that the control flashes on and off.

Custom

Set on the Display section of the custom properties page with the "FlashSpeed" text box.

Visual Basic

object **.FlashSpeed[=** *setting %* **]**

Remarks

The setting for FlashSpeed is in milliseconds.

Data Type

Long

FlashTime Property

Applies To

RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel

Description

Determines how long the control should flash when server sends new data and FlashEnabled is True.

Custom

Set on the Display section of the custom properties page with the "FlashTime" text box.

Visual Basic

*object***.FlashTime[**=*setting %***]**

Remarks

Enabled only when the "Flash On New Data" check box is checked. The setting for FlashTime is in milliseconds.

Data Type

Long

Font Property

Applies To

RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel

Description

Determines the font name, style, and size of text that will be used for the control's caption and value.

Custom

Set on the Fonts section of the custom properties page.

Visual Basic

object.**Font[**=*setting***]**

Remarks

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.

Data Type

Font

KnobWidth Property

Applies To

RSGauge, RSSlider

Description

Determines the width of the knob displayed on the control.

Custom

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.

Visual Basic

object.**KnobWidth[**=*setting %***]**

Remarks

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.

For the Gauge control, KnobWidth is represented as Diameter.

Data Type

Integer

LeftBorder Property

Applies To

RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel

Description

Sets the distance between left external border of the control and left edge of the control.

Custom

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.

Visual Basic

object.**LeftBorder[=**_setting %_**]**

Remarks

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.

Data Type

Integer

LinkErrorDisplay Property

Applies To

RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel

Description

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.

Custom

No access via custom property page.

Visual Basic

*[Form1.]Control.***LinkErrorDisplay***[=setting %]*

Remarks

When set to True, link error messages will be displayed; when set to False link error messages will not be displayed.

Data Type

Integer

LinkErrorNumber Property (Run Time Only)

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

LinkErrorString Property (Run Time Only)

Applies To

RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel

Description

Returns the error string associated with its LinkItem. Not available at design time and read only at run time.

Custom

No access via custom property page.

Visual Basic

*[Form1.]Control.***LinkErrorString**

Remarks

Use the LinkErrorString property along with a label control to display the error string. For example: Label1.Caption = rsdata1.LinkErrorString.

Data Type

String

LinkItem Property

Applies To

RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel

Description

Sets the item portion of the data link string to which the control is linked.

Custom

Adjusted in the LinkInfo section of the custom property page.

Visual Basic

***object*.LinkItem[=*setting %*]**

Remarks

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.

Data Type

String

LinkMode Property

Applies To

RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel

Description

Sets the type of link to be used for a DDE conversation and activates the connection.

Custom

Adjusted in the LinkInfo section of the custom properties page.

Visual Basic

object.**LinkMode[=***setting %***]**

Remarks

Available options for the LinkMode property are: 0 = None; 1= Automatic; 2 = Manual; and 3 = Notify.

Data Type

Integer

LinkServer Property

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

Description
Determines the application or server name that the control is linked to.

Custom
Adjusted in the LinkInfo section of the custom properties page.

Visual Basic
object.**LinkServer[=**_setting %_**]**

Remarks
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.

Data Type
String

LinkTip Property

Applies To

RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel

Description

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.

Custom

Enabled / disabled in the Display section of the custom properties page with the "Link Tip" check box.

Visual Basic

object.**LinkTip[**=*setting %*]

Settings

Settings	Description
True	Displays LinkTip window.
False	Disables LinkTip window.

Remarks

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.

Data Type

Integer

LinkTipBackColor Property

Applies To

RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel

Description

Sets the background color of the LinkTip popup box.

Custom

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.

Visual Basic

*object***.LinkTipBackColor[**=*setting***]**

Remarks

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.

Data Type

Color

LinkTipForeColor Property

Applies To

RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel

Description

Determines the color of the LinkTip message text.

Custom

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.

Visual Basic

object.**LinkTipForeColor[=***setting %***]**

Remarks

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.

Data Type

Color

LinkTipText Property

Applies To

RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel

Description

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).

Custom

No access via custom property page.

Visual Basic

*object***.LinkTipText [=***setting %***]**

Remarks

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.

Data Type

String

LinkTopic Property

Applies To	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
Description	Determines the topic portion of a data link string to which the control is linked.
Custom	Adjusted in the LinkInfo section of the custom property page.
Visual Basic	*object***.LinkTopic[=***setting %***]**
Remarks	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.
Data Type	String

MoveRefresh Property

Applies To

RSGauge, RSSlider

Description

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.

Custom

No access via custom properties page.

Visual Basic

object.**MoveRefresh[=***setting %***]**

Remarks

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.

Data Type

Integer

NotFilledColor Property

Applies To

RSGauge, RSVessel

Description

Determines the color of the not-filled area on the control.

Custom

Display section.

Visual Basic

*object***.NotFilledColor[=***setting %***]**

Remarks

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).

Data Type

Color

NumberOfDataValues Property (Run Time Only)

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

NumberOfSegments Property

Applies To RSCompare, RSVessel

Description 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.

Custom Segments text box on General page.

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

Data Type Integer

NumbersColor Property

Applies To RSVessel, RSWheel

Description 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.

Custom Display Section

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

Data TypeColor

Picture Property

Applies To

RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel

Description

Any file of the *.bmp,*.wmf, *.ico format may be displayed on the control by defining the picture file.

Custom

Display Section.

Visual Basic

*object***.Picture[=***filename***]**

Remarks

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.

Data Type

String

PictureStretch Property

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

Description Stretches the picture to fit the control boundaries.

Custom Check box in General Tab.

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

Remarks True streches the Picture. False lets the picture be displayed in its original size.

Data Type Integer

PictureUp Property

Applies To

RSButton, RSCompare

Description

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.)

Custom

Display section.

Visual Basic

*object***.PictureUp[=***file***]**

Remarks

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.

Data Type

String

PokeLength Property

Applies To

RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel

Description

Defines the number of controls in a control array to be used in a LinkPoke starting from the PokeStartIndex in a control array.

Custom

No access via custom property page.

Visual Basic

object.**PokeLength[**=*setting*%**]**

Remarks

The number of controls to be used in single message transaction if AdvanceDDE is used.

Data Type

Integer

PokeStartIndex Property

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

Description Sets the starting index value of the controls in a control array to be used in a poke to the DDE server (source).

Custom No access via custom property page.

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

Data Type Integer

RequestLength Property

Applies To

RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel

Description

Defines the number of controls in a control array for a LinkRequest from the DDE server (source).

Custom

N/A on the Custom properties page

Visual Basic

*object***.RequestLength[=***setting***%]**

Remarks

This property applies if the control is part of an array.

Data Type

Integer

RequestStartIndex Property

Applies To

RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel

Description

Sets the starting index value of the controls in a control array to be used in a request from the DDE server (source).

Visual Basic

*object***.RequestStartIndex[=***setting*%**]**

Data Type

Integer

ReverseDirection Property

Applies To

RSGauge, RSSlider, RSWheel

Description

This property reverses the start and end values.

Custom

General section.

Visual Basic

object.**ReverseDirection[=***setting***%]**

Remarks

Setting to False uses the defined start and end values. Setting to True reverses the start and end values.

Data Type

Integer

RightBorder Property

Applies To

RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel

Description

Sets the distance between right external border of the control and right edge of the control.

Custom

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.

Visual Basic

object.**RightBorder[=**_setting %_**]**

Remarks

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.

Data Type

Integer

Scale1DecimalPlaces Property

Applies To RSGauge, RSSlider

Description Sets or returns the number of decimal places used for the Scale 1 numerals.

Custom Key in or Use the spin buttons next to ‘Decimal’ on the Scale Tab.

Visual Basic *object*.**Scale1DecimalPlaces[**=*setting*%**]**

Data Type Integer

Scale1End Property

Applies To RSGauge, RSSlider

Description Sets or returns the End position and value of the Scale 1 markings.

Custom Key in or click on the # sign to bring up the key pad on ‘Place Scale1 @’ settings box on the Scale Tab.

Visual Basic *object***.Scale1End[=***setting* **%]**

Data Type Double

Scale1Length Property

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

Scale1Major Property

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

Scale1MajorColor Property

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

Scale1Minor Property

Applies To RSGauge, RSSlider

Description Sets the number of minor scale divisions for Scale 1.

Custom Key in or Use the spin buttons next to ‘Minor’ on the Scale Tab.

Visual Basic *object*.**Scale1Minor[=***setting***%]**

Data Type Integer

Scale1MinorColor Property

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

Scale1NumbersVisible Property

Applies To RSGauge, RSSlider

Description Determines whether or not Scale 1 numbers are visible.

Custom Check Box on the Scale Tab.

Visual Basic *object***.Scale1NumberVisible[=***setting* **%]**

Remarks True makes the numbers visible. False hides the numbers.

Data Type Integer

Scale1Offset Property

Applies To RSGauge, RSSlider

Description Defines the offset of the Scale 1 markings from the inside border.

Custom Double Click on the # mark next to scale in the Setup Area on the General Tab.

Visual Basic *object***.Scale1Offset[=***setting* %**]**

Data Type Integer

Scale1Start Property

Applies To RSGauge, RSSlider

Description Determines the Start position and value of the Scale 1 markings on the control.

Custom Key in or click on the # sign to bring up the key pad on ‘Place Scale1 @’ settings box on the Scale Tab.

Visual Basic *object*.**Scale1Start[=***setting* %**]**

Data Type Double

Scale1String Property

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

Scale1StringEnabled Property

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

Scale1Style Property

Applies To RSGauge, RSSlider

Description Defines the position of the Scale 1 numbers on the control.

Custom Drop down combo box on the Scale Tab.

Visual Basic *object*.**Scale1Style[**=*setting* **%]**

Remarks Valid styles are 0=Next Scale, 1=Inside Border and 2=Outside Border

Data Type enumScaleStyles

Scale1TextColor Property

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

Scale1TrailingZeros Property

Applies To	RSGauge, RSSlider
Description	Defines the number of trailing zeros for Scale 1.
Custom	Check Box on the Scale Tab.
Visual Basic	_object_**.Scale1TrailingZeros[=**_setting_ **%]**
Data Type	Integer

Scale1Type Property

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

Scale1Visible Property

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

Scale1Width Property

Applies To

RSGauge, RSSlider

Description

Sets the width of the Scale 1 markings.

Custom

Edit directly on the setup area of the General custom properties tab.

Visual Basic

*object.***.Scale1Width[=***setting***%]**

Data Type

Integer

Scale2DecimalPlaces Property

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

Scale2End Property

Applies To

RSGauge, RSSlider

Description

Sets or returns the End position and value of the Scale 2 markings on the control.

Custom

Key in or click on the # sign to bring up the key pad on ‘Place Scale1 @’ settings box on the Scale Tab.

Visual Basic

object.**Scale2End[=***setting* **%]**

Data Type

Double

Scale2EndNumber Property

Applies To

RSGauge, RSSlider

Description

Sets or returns the end number value of the numerals on Scale 2.

Custom

Key in or Use the spin buttons next to ‘Scale 2 Numbering’ on the Scale Tab.

Visual Basic

object.**Scale2EndNumber[=***setting* **%]**

Remarks

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.

Data Type

Double

Scale2Length Property

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

Scale2Major Property

Applies To RSGauge, RSSlider

Custom Key in or Use the spin buttons next to ‘Major’ on the Scale Tab.

Description Sets the number of major scale divisions for Scale 2.

Visual Basic *object*.**Scale2Major[=***setting* **%]**

Data Type Integer

Scale2MajorColor Property

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

Scale2Minor Property

Applies To RSGauge, RSSlider

Description Sets the number of scale divisions for the minor scale of Scale 2.

Custom Key in or Use the spin buttons next to ‘Minor’ on the Scale Tab.

Visual Basic *object*.**Scale2Minor[=***setting* **%]**

Data Type Integer

Scale2MinorColor Property

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

Scale2NumbersVisible Property

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

Scale2Offset Property

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

Scale2Start Property

Applies To

RSGauge, RSSlider

Description

Sets or returns the Start position and value of the Scale 2 markings on the control.

Custom

Key in or click on the # sign to bring up the key pad on ‘Place Scale2 @’ settings box on the Scale Tab.

Visual Basic

*object***.Scale2Start[=***setting* **%]**

Data Type

Double

Scale2StartNumber Property

Applies To

RSGauge, RSSlider

Description

Sets or returns the Scale 2 start number value of the scale numerals.

Custom

Key in or Use the spin buttons next to ‘Scale 2 Numbering’ on the Scale Tab.

Remarks

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.

Visual Basic

object.**Scale2StartNumber[=***setting* **%]**

Data Type

Double

Scale2String Property

Applies To RSSlider, RSGauge

Description Determines the String Caption to be displayed next to Scale 1.

Custom No access via custom property page.

Visual Basic *object*.**Scale1String** [=*setting%*%]

Data Type String

Scale2StringEnabled Property

Applies To RSSlider, RSGauge

Description Enables / Disables the display of String caption next to Scale 1.

Custom No access via custom property page.

Visual Basic *object*.**Scale1StringEnabled**[=*setting* %]

Data Type Integer

Scale2Style Property

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

Scale2TextColor Property

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

Scale2TrailingZeros Property

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

Scale2Type Property

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

Scale2Visible Property

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

Scale2Width Property

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

ScaleBorderColor Property

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

ScaleHighlight Property

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

ScaleShadow Property

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

ScreenPriority Property

Applies To

RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel

Description

Determines if the Windows dispatch commands called messages will be acted upon immediately or the Windows system will decide when to send the messages.

Visual Basic

object.**ScreenPriority[=***setting* **%]**

Remarks

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.

Data Type

Integer

Shadow Property

Applies To RSCompare, RSGauge, RSSlider, RSVessel

Description Specifies if the control’s knob or graphic shadow is visible or hidden.

Custom Check box in General section.

Visual Basic _object_**.Shadow[=**_setting_ %**]**

Remarks True sets the shadow visible. False hides the shadow.

Data Type Integer

ShadowOffsetX Property

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

ShadowOffsetY Property

Applies To RSCompare, RSGauge, RSSlider, RSVessel

Description Sets the vertical offset of the control's knob or graphic shadow.

Custom General section. Edit the picture directly in the setup area.

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

Data Type Integer

StartValue Property

Applies To

RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel

Description

Specifies the starting value for the control.

Custom

Adjusted in the “Start Value” text box in the Value section of the custom properties page.

Visual Basic

*object***.StartValue[**=*setting* **%]**

Remarks

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.

Data Type

Double

Symbol Property

Applies To

RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel

Description

Serves as an alias name for the server, topic and item to be used in a DDE link.

Custom

LinkInfo Section.

Visual Basic

object.**Symbol[=***setting* **%]**

Remarks

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.

See also: *LinkTip* property.

Data Type

String

TabIndex Property

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

TabStop Property

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

Description Adds or removes the control from the form's Tab order at run time.

Custom No access via custom property page.

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

Remarks Setting to True adds the control to the form's tab order. False removes the control from the tab order.

Data Type Integer

Tag Property

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

Description Sets or returns an expression that stores any extra data needed in the application.

Custom No access via custom property page.

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

Remarks This is a user defined property that does not affect other VB properties.

Data Type String

Top Property

Applies To

RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel

Description

Determines the distance between the top edge of a control and the top edge of its container.

Custom

No access via custom property page.

Visual Basic

object**.Top[=**_setting_ **%]**

Data Type

Single

TopBorder Property

Applies To

RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel

Description

Determines the distance between the top external border of the control and the top of the control.

Custom

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.

Visual Basic

*object***.TopBorder[=***setting* **%]**

Data Type

Integer

TrailingZeros Property

Applies To

RSCompare, RSData, RSGauge, RSSlider, RSVessel

Description

Determines if the value is displayed with trailing insignificant zeros.

Custom

Check box in General section.

Visual Basic

object **.TrailingZeros[=** *setting* **%]**

Remarks

True displays the value with trailing zeros. False displays the value without trailing zeros.

Data Type

Integer

UseInPoke Property

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

Description
A flag to signify that the control should be used in a LinkPoke or DoPoke.

Custom
No access via custom property page.

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

Remarks
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.

Data Type
Integer

UseInRequest Property

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

Description A flag to signify that the control should be used in a LinkRequest or DoRequest.

Custom No access via custom property page.

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

Remarks 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.

Data Type Integer

UseStartEndValue Property

Applies To

RSCompare, RSData, RSWheel

Description

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.

Custom

Adjusted in the General section of the custom properties page with a checkbox in the Options window.

Visual Basic

object.**UseStartEndValue[**=*setting %***]**

Remarks

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.

Data Type

Integer

Value Property

Applies To

RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel

Description

Contains the value of the control at runtime.

Custom

Value section.

Visual Basic

object.**Value[=***setting* **%]**

Remarks

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.

See also: *Clip* property.

Data Type

String

ValueBackColor Property

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

ValueColor Property

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

ValuePadDownLoad Property

Applies To

RSCompare, RSData, RSVessel

Description

Enables / disables the number entry pad from being activated when the control is clicked on at run time.

Custom

Check box in the General section.

Visual Basic

object.**ValuePadDownLoad[=**_setting %_**]**

Data Type

Integer

ValueShadow Property

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

ValueShadowColor Property

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

ValueTransparent Property

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

Description Determines if the value background is transparent or opaque.

Custom Check box in General section.

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

Remarks True sets it to transparent. False sets it to opaque.

Data Type Integer

ValueX Property

Applies To

RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel

Description

Sets the horizontal position of the Value display within the control.

Custom

General section. Edit the picture directly.

Visual Basic

object.**ValueX[=**_setting_ **%]**

Remarks

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.

Data Type

Integer

ValueY Property

Applies To

RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel

Description

Sets the vertical position of the Value display within the control.

Custom

General section. Edit the picture directly.

Visual Basic

*object***.ValueY[=***setting* **%]**

Remarks

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.

Data Type

Integer

Visible Property

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

Description Determines if the control is visible at run-time.

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

Remarks True sets the control to be visible and False hides it.

Data Type Integer

WhatsThisHelpID Property

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

Description Sets or returns the associated WhatsThisHelp context number.

Visual Basic *object***.WhatsThisHelpID[=***number***]**

Remarks These context numbers are associated with Windows help files.

Data Type Long

Width Property

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

Description Sets or returns the width of the control.

Custom General section. Edit the picture directly.

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

Data Type Single

WriteStyle Property

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

WriteValue Property

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

Description Determines if the value will be downloaded to a database or LinkItem.

Custom The WriteValue is used when a person programmatically changes a value.

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

Data Type 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

Applies To

RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel

Description

Indicates that the contents of a control's Value property have changed.

Visual Basic

Private Sub *object*_**Change([***index* **As Integer])**

Remarks

The Change event procedure can synchronize or coordinate data display among controls.

For additional information, refer to the description of the **Change** event in the Microsoft Visual Basic Language Reference Manual.

Click Event

Applies To

RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel

Description

Occurs when the user presses and then releases a mouse button over an object.

Visual Basic

Private Sub *object*_**Click([***index* **As Integer])**

Remarks

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.

For additional information, refer to the description of the **Click** event in the Microsoft Visual Basic Language Reference Manual.

Clicks Event

Applies To

RSButton, RSCompare, RSVessel

Description

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.

Visual Basic

Sub RSButton1_Clicks ([ByVal dValue As Double, ByVal iButtonIndex As Integer])

Remarks

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

Applies To	RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel
Description	Occurs when the user presses and releases a mouse button and then presses and releases it again over an object.
Visual Basic	**Private Sub** *object*_**DblClick (***index* **As Integer)**
Remarks	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. For additional information, refer to the description of the DblClick event in the Microsoft Visual Basic Language Reference Manual.

DragDrop Event

Applies To

RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel

Description

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).

Visual Basic

Private Sub *object*_**DragDrop([***index* **As Integer,]***source* **As Control,** *x* **As Single,** *y* **As Single)**

Remarks

Use a DragDrop event procedure to control what happens after a drag operation is completed.

For additional information, refer to the description of the DragDrop event in the Microsoft Visual Basic Language Reference Manual.

DragOver Event

Applies To

RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel

Description

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.

Visual Basic

Private Sub *object*__**DragOver([**__*index* **As Integer,]**_source_ **As Control,** *x* **As Single,** *y* **As Single,** *state* **As Integer)**

Remarks

Use a DragOver event procedure to determine what happens after dragging is initiated and before a control drops onto a target.

For additional information, refer to the description of the DragOver event in the Microsoft Visual Basic Language Reference Manual.

EndMove Event

Applies To

RSGauge, RSSlider

Description

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.

Visual Basic

Private Sub *object*_**EndMove ([byVal** *Value* **As Double, ByVal** *Index* **As Integer])**

Remarks

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.

Note: This event is unavailable when using a RSGauge type with fill boxes unless you apply a needle to it.

GotFocus Event

Applies To

RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel

Description

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.

Visual Basic

Private Sub *object***_GotFocus([***index* **As Integer])**

Remarks

Typically, you use a GotFocus event procedure to specify the actions that occur when a control or form first receives the focus.

For additional information, refer to the description of the GotFocus event in the Microsoft Visual Basic Language Reference Manual.

KeyDown, KeyUp Events XE " KeyUp Event"

Applies To

RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel

Description

Occur when the user presses (KeyDown) or releases (KeyUp) a key while an object has the focus.

Visual Basic

Private Sub *object*_**KeyDown([***index* **As Integer,]***keycode* **As Integer,** *shift* **As Integer)**

Private Sub *object*_**KeyUp([***index* **As Integer,]***keycode* **As Integer,** *shift* **As Integer)**

Remarks

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

Applies To

RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel

Description

Occurs when the user presses and releases an ANSI key.

Visual Basic

Private Sub *object*_**KeyPress([***index* **As Integer,]***keyascii* **As Integer)**

Remarks

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

Applies To

RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel

Description

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.

Visual Basic

Private Sub *object*_**LinkError(ByVal** *iRet* **As Integer, ByVal** *ErrorString* **As String)**

Remarks

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

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

Description Occurs when the control’s LinkItem has an incorrect format.

Visual Basic **Private Sub** *object***_LinkItemNotSupported ( )**

LinkItemSupported Event

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

Description Occurs when the control's LinkItem has a correct format.

Visual Basic **Private Sub** *object*_**LinkItemSupported ( )**

LinkNotify Event

Applies To

RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel

Description

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).

Visual Basic

Private Sub *object*_**LinkNotify([ByVal** *index* **As Integer])**

Remarks

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

Applies To

RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel

Description

Occurs when the client (control) has exhausted its memory resources.

Visual Basic

Private Sub *object*_**DblClick ([ByVal** *Index* **As Integer])**

LinkServerDisconnected Event

Applies To

RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel

Description

Occurs when the control is connected to a server and that server becomes unavailable..

Visual Basic

Private Sub *object*_**LinkServerDisconnected ( )**

LinkUnableToConnectToServer Event

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

Description Occurs when the control attempts to connect to a server that is unavailable.

Visual Basic **Private Sub** *object*_**LinkUnableToConnectToServer** ( )

LostFocus Event

Applies To

RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel

Description

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.

Visual Basic

Private Sub *object*_**LostFocus([***index* **As Integer])**

Remarks

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

Applies To

RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel

Description

Occur when the user presses (MouseDown) or releases (MouseUp) a mouse button.

Visual Basic

Private Sub *object* **_MouseDown([***index* **As Integer,]***button* **As Integer,** *shift* **As Integer,** *x* **As Single,** *y* **As Single)**

Private Sub *object* **_MouseUp([***index* **As Integer,]***button* **As Integer,** *shift* **As Integer,** *x* **As Single,** *y* **As Single)**

Remarks

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

Applies To

RSButton, RSCompare, RSData, RSGauge, RSSlider, RSVessel, RSWheel

Description

Occurs when the user moves the mouse.

Visual Basic

Private Sub *object* **_MouseMove([***index* **As Integer,]***button* **As Integer,** *shift* **As Integer,** *x* **As Single,** *y* **As Single)**

Remarks

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

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

Description The PokeCompleted event occurs when a poke action has completed

Visual Basic **Private Sub** *object*_**PokeCompleted ([ByVal** *iRet* **As Integer])**

Remarks The argument iRet returns error numbers

RequestCompleted Event

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

Description The RequestCompleted event occurs when a request action has completed.

Visual Basic **Private Sub** *object*_**RequestCompleted ([ByVal** *iRet* **as Integer])**

Remarks The argument iRet returns error numbers

StartMove Event

Applies To

RSGauge, RSSlider

Description

Occurs when the RSSlider’s knob or RSGauges Needle begins to move.

Visual Basic

Private Sub *object*_**StarMove([ByVal** *Value* **As Double, ByVal** *Index* **as Integer])**

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

Applies To

RSButton, RSCompare, RSData, EventMaster, RSGauge, RSSlider, RSWheel

Description

Returns or sets the container of a control.

Visual Basic

object*.Control.***Container***[=setting]*

Remarks

Not available at design time.   A control's parent (container) may be changed at run time with the Container method.

DoPoke Method

Applies To

RSButton, RSCompare, RSData, RSGauge, RSSlider, RSWheel, RSVessel

Description

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.

Visual Basic

object.*Control.***DoPoke**

Remarks

The control’s LinkMode property must be set to 2 - Manual.

See also: *PokeStartIndex* and *PokeLength* properties, *LinkPoke* method.

DoRequest Method

Applies To

RSButton, RSCompare, RSData, RSGauge, RSSlider, RSWheel, RSVessel

Description

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.

Visual Basic

object.*Control*.**DoRequest**

Remarks

The control’s LinkMode property must be set to 2 - Manual.

See also: *RequestStartIndex* and *RequestLength* properties, *LinkRequest* Method.

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:

Constant	Value	Description
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

Applies To

RSButton, RSCompare, RSData, RSGauge, RSSlider, RSWheel, RSVessel

Description

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.

Visual Basic

object.**LinkPoke**

Remarks

Typically, information in a DDE conversation flows from source to destination. However, LinkPoke allows a destination object to supply RSData to the source.

See also: *PokeStartIndex* and *PokeLength* properties, *DoPoke* method.

LinkRequest Method

Applies To

RSButton, RSCompare, RSData, RSGauge, RSSlider, RSWheel, RSVessel

Description

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.

Visual Basic

object.**LinkRequest**

Remarks

LinkRequest causes the source application to send the most current RSData to object.

See also: *RequestStartIndex* and *RequestLength* properties, *DoRequest* method.

Move Method

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

Description Moves an MDIForm, Form, or control.

Visual Basic *object*.**Move** left, top, width, height

Remarks 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.

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

Object Method

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

Description Returns an object in a control.

Visual Basic *object*.*Control*.**Object**[.property|.method][=value]

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

Parent Method

Applies To

RSButton, RSCompare, RSData, RSGauge, RSSlider, RSWheel, RSVessel

Description

Returns the container on which an object is located.

Visual Basic

object.*Control*.**Parent**

Remarks

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

SetFocus Method

Applies To

RSButton, RSCompare, RSData, RSGauge, RSSlider, RSWheel, RSVessel

Description

Moves the focus to the specified control or form.

Visual Basic

*object.***SetFocus**

Remarks

The object (control) that can receive the focus. After invoking the SetFocus method, any user input is directed to the specified control.

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

ShowWhatsThis Method

Applies To

RSButton, RSCompare, RSData, RSGauge, RSSlider, RSWheel, RSVessel

Description

Displays a selected topic in a Help file using the What's This popup provided by Windows 95 Help.

Visual Basic

object.**ShowWhatsThis**

Remarks

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

Applies To

RSButton, RSCompare, RSData, RSGauge, RSSlider, RSWheel, RSVessel

Description

Places a specified MDIForm, Form, or control at the front or back of the z-order within its graphical level.

Visual Basic

object.**ZOrder** position

Remarks

The ZOrder method syntax has these parts:

Part	Description
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.