Contents

Chapter 1

Introducing RSTools ................................................................. 1
  Features and benefits ...................................................... 1
  RSTools controls ................................................................. 2
  System requirements .......................................................... 3
  Installing RSTools ................................................................. 3
  Activating RSTools ................................................................. 4
  Troubleshooting the installation ........................................... 4

Chapter 2

Getting started using RSTools .................................................. 5
  Quick start steps for RSView32 ............................................ 5
  Quick start steps for Visual Basic ........................................ 6
  Adding RSTools controls to your VB project ....................... 6
  Creating a template for your RSTools project ..................... 7
  Placing the RSTools control on a VB form ......................... 8
  Quick start steps for FrontPage ......................................... 9

Chapter 3

Working with RSTools .............................................................. 11
  RSData .............................................................................. 11
  Logging data to a database (one piece of data) .................. 11
  Poking and requesting data (one piece of data) ................. 13
RSWheel ................................................................................................................... 14
Viewing different wheel types ................................................................................ 14

RSButton .................................................................................................................. 16
Creating an array link with RSButton controls ....................................................... 16

RSSlider .................................................................................................................... 17
Binding RSSlider to a Visual Basic Data control ................................................... 17
RSSlider code ............................................................................................................. 18

RSGauge ................................................................................................................... 22
Displaying a warning area ....................................................................................... 22

RSVessel .................................................................................................................. 24
Using RSVessel with the Microsoft Grid control ................................................... 24

RSCompare ............................................................................................................. 26
Logging a value to a data source ............................................................................. 26

RSAnimator ............................................................................................................. 28
Using the Snapshot feature to create animation ................................................... 28

RSAAlarm ................................................................................................................ 31
Creating an application using RSAAlarm ............................................................... 31

RSEventMaster .................................................................................................... 33
Integrating RSAAlarm and RSEventMaster ......................................................... 33

Chapter 4

Getting the information you need ........................................................................ 35

Technical support ................................................................................................. 37

Solving problems ................................................................................................ 38

Index ..................................................................................................................... 41

ii · Getting Results with RSTools
Introducing RSTools

RSTools is a unique line of graphical ActiveX custom controls that let you turn any data into a dynamic picture. Display alarms, slide switches, gauges, and wheels using RSTools controls. In addition, link most RSTools controls to a database or server for up-to-the-minute data updates.

You can easily add a professional “custom” look to all interface and data management projects. RSTools controls are the fast and easy-to-use programming tools for developing applications for manufacturing and process control. They give you the power to build graphical database interfaces that tell the story in a flash.

Use the RSTools controls within a variety of containers such as Microsoft Visual Basic, Microsoft Access, Microsoft FrontPage, and Rockwell Software RSView32.

Features and benefits

- Configure RSTools to connect to and display DDE data by setting RSTools properties without the need for writing a single line of code. (Does not apply to RSAlarm, RSAnimator, and RSEventMaster.)

- RSTools is designed to be used with any Dynamic Data Exchange (DDE) server and provides enhanced performance when used with the AdvanceDDE protocol.

- The RSTools controls have three possible modes of DDE communication: AdvanceDDE, XLTABLE, or CF_TEXT mode. If the control cannot detect an AdvanceDDE server, then it will try XLTABLE. If it fails at finding XLTABLE, it will default to TXT mode, which is Microsoft Windows's default DDE protocol.

- Use the RSTools controls as data bound controls. They automatically add and display data from the Visual Basic Data control and from the Remote Data control, which supports ODBC, Microsoft Access, dBASE Foxpro, Paradox, Btrieve and several other database types. (Does not apply to RSAlarm, RSAnimator, and RSEventMaster.)

- The RSTools controls also support the Paste Link method from RSLinx, Excel or other servers for transferring DDE link information via the clipboard. (Does not apply to RSAlarm.)
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 or handled via code in one of the events.

**RSTools controls**

The controls that comprise RSTools are:

- RSAnimator
- RSAAlarm
- RSBButton
- RSCompare
- RSData
- RSEventMaster
- RSGauge
- RSSlider
- RSVessel
- RSWheel

**IMPORTANT** You must have RSJunctionBox installed to use the AdvanceDDE or XLTABLE DDE modes offered by some RSTools controls. RSJunctionBox is a communication module offered by Rockwell Software that enables AdvanceDDE and XLTABLE DDE protocols for RSJunctionBox.
System requirements

Before you install RSTools, make sure that your computer meets the minimum system requirements. The system requirements include:

- Personal computer with a 486 or higher processor (Intel Pentium recommended)
- 8 MB of RAM (16 MB recommended)
- 10 MB free hard disk space (30 MB recommended)
- CD-ROM drive
- VGA graphics card (Color 800 x 600 or 1024 x 768 graphics recommended)
- Microsoft Visual Basic 5.0 or later (Professional or Enterprise Edition recommended), Microsoft Access, Rockwell Software’s RSView32, or other ActiveX control container
- Microsoft Windows NT 4.0 or Microsoft Windows 95

Installing RSTools

Install RSTools using Rockwell Software’s SETUP.EXE. The setup program installs all RSTools files, the Help system, sample applications, and other product components from the RSTools CD-ROM to your hard disk.

1. Close all open windows and programs.
2. Insert the RSTools CD-ROM in the CD-ROM drive.
3. From the Start menu, choose Run.
4. Enter E:\SETUP where E is the drive containing your CD-ROM.
5. Follow the setup instructions on the screen.
Activating RSTools

RSTools includes a master disk that you must use to activate the software. At the end of the installation, the setup program asks you to insert the activation disk. Follow the steps below to activate RSTools. RSTools must be activated on each computer where it is installed.

1. Insert the master disk in the diskette drive.

2. Select the location of the master disk from the Available Drive(s) combo box, and then click OK.

3. If RSTools is installed on drive C, click OK. Otherwise, select the drive where RSTools is installed.

4. Click the Move button.

5. Close the EVMove window by clicking OK.

---

**IMPORTANT**

If you have trouble activating RSTools, view the help file called COPYPROT.HLP, located on the RSTools installation CD-ROM and in the directory where RSTools is installed. This file provides detailed information on how activation works and how to solve activation problems.

To view the help file, open Explorer and double-click COPYPROT.HLP.

---

Troubleshooting the installation

If RSTools did not install properly:

- Does your computer have enough memory? RSTools requires at least eight megabytes of RAM.

- Did you use the correct master disk? Most Rockwell Software products include an activation disk. You must use the one included with RSTools.

- Does your computer have a virus? Boot sector viruses can damage the activation disk. The Rockwell Software support team recommends using a commercial virus protection program.
This chapter describes how to start using RSTools in Rockwell Software's RSView32, Microsoft's Visual Basic, and Microsoft's FrontPage. You can use RSTools in any ActiveX container.

**Quick start steps for RSView32**

RSView32 can send and receive information using the RSTools controls. Information is passed between the RSTools control and RSView32 using RSView32 tags.

For example, you can embed an RSTools control in an RSView32 graphic display and attach an RSView32 tag to the control's Value property. As the tag's value changes, the RSTools control's Value property changes to reflect the tag's value.

1. In RSView32, open a large new display.
2. Click the Ctrl icon and draw a box. The Insert ActiveX Control window appears.
3. Select the name of the control you want to use from the list in the window. The RSTools control appears in the box you drew.
   
   Note that RSData is called Rockwell Numeric in the list and RSButton is called Rockwell Pushbutton.
4. Right-click on the RSTools control. A menu appears.
5. Select **Animation > ActiveX Control > Properties**. The ActiveX Control Properties window appears.
6. Select the name of the property to you want to assign to a tag from the list of properties. Typically, you will want to select the Value property, but you can select any property.
7. Enter the name of the tag you want to assign to the property or click the ellipses button to display a browser where you can select a tag.

8. Click OK.

<table>
<thead>
<tr>
<th>IMPORTANT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Perform the procedure for Adding RSTools to RSView32 when you want to use RSView32 tags to send and receive data. If you want to connect directly to a DDE server, select <strong>Rockwell Control Object &gt; Properties</strong> from the floating menu instead of performing step 5 in the procedure. Use the LinkInfo tab that appears to set up DDE communications instead of following the remainder of the steps in the procedure. Note that you should never use tags and a DDE server for communications at the same time.</td>
</tr>
</tbody>
</table>

Quick start steps for Visual Basic

**Adding RSTools controls to your VB project**

To use RSTools in Visual Basic, add the RSTools control to the Visual Basic toolbox.


2. Select Project > Components.

3. Select each control from the list by clicking the check box to the left of the control’s name.

You cannot just click on the name of the control to select it. You must click in the check box. Each RSTools control begins with the word Rockwell.
4. Click OK. The controls are added to the toolbox.

4. Click OK. The controls are added to the toolbox.

IMPORTANT Visual Basic adds the RSTools controls only to this project. If you want the RSTools controls to automatically appear in all of your projects, you must create a template. Follow the steps on the next page to create a template.

Creating a template for your RSTools project

When you add the RSTools controls to your project by selecting Components from the Project menu, Visual Basic adds the controls only to the open project. To cause the RSTools controls to appear in the toolbox of all new projects, create a Visual Basic template.

1. Make sure the RSTools controls appear in the toolbox for the open project. Also make sure the project is new (you haven't added any controls or code to the form).

2. Save the form and project in the following directory:
   Program Files\DevStudio\VB\Template\Projects

3. When starting a new project where you want to use the RSTools controls, choose the name of the project that you just saved instead of Standard EXE in the VB New Project window.
Placing the RSTools control on a VB form

To place an RSTools control on a form:

1. Select the RSTools control you want to use in the Visual Basic toolbox.
   Refer to “Adding the RSTools controls to your VB project” on page 6 if the
   RSTools control you want to use is not displayed in the Visual Basic toolbox.

2. Click and drag to draw the outline of the RSTools control on the form.
   If you double-click the RSTools control’s icon in the Toolbox, Visual Basic places the
   control in the center of the form.

3. To open the Custom Properties Page, position the cursor on the control and
   click the right mouse button. Select Properties from the floating menu that
   appears.
Quick start steps for FrontPage

You can add RSTools to your web page using FrontPage 98 and have it available for use by anyone who accesses your web page.

1. Start FrontPage.

2. Select FrontPage Editor from the Tools menu.

3. Open an existing HTML document or create a new one.

4. Position the cursor where you want to add a control.

5. Select Insert > Advanced > ActiveX Control.
6. Select the name of the control from the Pick a Control box.

All ActiveX controls currently registered on your computer should appear in this box. Note that Rockwell RSNumeric in this window is the same as RSData and Rockwell PushButton is the same as RSBUTTON.

7. In the Name box, enter a name for the control.

The name should be unique among all objects on your web page and will be used when accessing the control via script.

8. Specify the size of your control by filling in the Height and Width boxes. If you do not specify a size, default values will be chosen for you.

9. Click OK.

A fully functional RSTools control is now embedded in your HTML document. To customize the control, click the Properties button on the window and make changes.
Chapter 3  Working with RSTools

RSTools includes ten ActiveX controls that can be used to display data graphically and display behind-the-scenes control of events and animation. This chapter describes each control and provides an example of how you may want to use it.

**RSData**

Use RSData’s properties and events with AdvanceDDE to allow your application to continually read, write, and display a large amount of data. For example, use the Change event to create bar graphs that change as the data being received changes. The Clip property allows you to transfer large amounts of data quickly to other controls such as a gauge or graph. Place a Grid control on your form and then use the Clip property to quickly display data in the grid.

RSData greatly increases DDE data transfers by communicating in Rockwell Software’s AdvanceDDE protocol. AdvanceDDE compresses a large amount of data and transfers it to and from your application in a block array of data.

**Logging data to a database (one piece of data)**

Two of the most important features of RSData are its ability to log data to a database and request data from a DDE server. The example below describes how to log data.

The example described here was installed on your system during the RSTools installation. You may use any part of the sample in your own applications. The sample project is located in the \Program Files\Rockwell Software\RSTools\Samples subdirectory on the drive you where you installed RSTools.

The DATALOGDEMO.VBP project logs one piece of data at a time to a Microsoft Access database using RSData. The project consists of three forms: **ConfigLink** allows you to set up the DDE link by setting the server, topic and item. ConfigLink requires you to use RSLinx as your DDE server. This form uses the LinkServer, LinkTopic, and LinkItem properties to set up the DDE link.
**DatalogDemo** provides three methods for logging data to the DEMO.MDB database:

- log data on demand by taking a snapshot of the data whenever you want
- log data periodically by setting an interval when RSData should log the data
- log data whenever a change occurs

This form uses the DataUpdate property to insert the data in the database at the specified time. It also uses the Change event to detect a change in the data being read and cause the database to be updated.

**Splash** supplies the banner page providing the name of the application.
Poking and requesting data (one piece of data)

The POKENREQUEST.VBP project (located in the \Program Files\Rockwell Software\RSTools\Samples subdirectory) allows you to manually poke one piece of data to a processor or request data from a processor. If you are poking data, you will manually enter the data to poke and then click a button to complete the poke. The project uses the DoPoke method to perform the poke operation.

If you are requesting data from a DDE server, the project uses the DoRequest method to read the data from the processor. Use the LinkRequest method instead to request data from a single control.

The POKENREQUEST.VBP project includes three forms:

- **PokenRequest** allows you to set up the DDE Link where you will receive or poke data by setting the server, topic and item. The PokenRequest form requires you to use RSLinx as your DDE server. This form uses the LinkServer, LinkTopic, and LinkItem properties to set up the DDE Link.

  In this sample, you must specify the number of values that will be poked using the NumberofDataValues property.

  **IMPORTANT** If your application does not set the NumberofDataValues property, you will receive an “Invalid index for data access” error message when running the application.

- **PokeData** allows you to enter the data that RSData will poke to the DDE server.

- **Splash** supplies the banner page providing the name of the application.
RSWheel

RSWheel is a panel thumbwheel emulator with multi-column spin control. Use RSWheel to enter data or display data from a database in a graphical wheel.

The types of wheel available for display are single spin, classical multi-wheel, add-a-spin, electronic advance, slot machine (random number generator) and read only. Decimal, octal, binary and hexadecimal data types are supported.

**Viewing different wheel types**

The following example enables you to view each of the available wheel types.

1. Open a new project and form in Visual Basic.
2. Place six RSWheel controls on the form.
3. Set the following properties:
   - SpinType0 = 1 – Side Button
   - SpinType1 = 2 – All Buttons
   - SpinType2 = 3 – Mechanical
   - SpinType3 = 4 – Spin Last
   - SpinType4 = 5 – Slot Machine
   - SpinType5 = 6 – No Spin
4. Set the Caption property of each control to describe each spin type.
   To adjust the placement of the caption on the control, use the General tab of the Custom Properties Page. The form should look like this:
5. If the value is visible above your control and you wish to remove it, right-click in the Setup Area of the General section of the Custom Properties Page. On the floating menu that appears, click Value to disable displaying the value.

6. Add a Visual Basic CommandButton control below the slot machine and change its caption to GO.

7. To set the slot machine wheel to automatically spin and generate new values, add the following code.

```vbnet
Private Sub Command1_Click()
    If RSWheel5.HandleDown = False Then
        RSWheel5.HandleDown = True
    Else
        RSWheel5.HandleDown = False
    End If
End Sub
```

8. Run your Visual Basic program and click the GO command button to see the random generator in action.
**RSButton**

The RSButton control allows you to place different button types on your form. Button types include momentary, toggle, rocker, checkbox, option, dual button, diode, and picture buttons.

Use RSButton properties to monitor and control data. Click a button to fire an event that reads data from a database or saves data to a database. Use RSButton events to notify the user when data changes or when data finishes its request or poke.

**Creating an array link with RSButton controls**

The following example links an RSButton control with a Boolean Excel database and creates an array of eight buttons in either the up or down state based on the values read from the database.

1. Place an RSButton control on a form in Visual Basic and set the following properties:
   - ButtonStyle: 6 (Diode)
   - LinkServer: Excel
   - LinkTopic: [book1]sheet1
   - LinkItem: R1C1:R8C1
   - LinkMode: 1(Automatic)

2. Open Excel and enter random values of 0 and 1 in cells A1 through A8.

3. Run your Visual Basic program. RSButton splits into eight different buttons. The red buttons are linked with a value of 0 and are in the up state. The green buttons are linked with a value of 1 and are in the down state.
RSSlider

The RSSlider control provides a graphical means to input data in your application. As you move the slider switch, RSSlider reads the new value and fires an event or saves the value to a database. You can set min/max points and scaling on the slider. RSSlider also supports Rockwell Software’s AdvanceDDE format and can be linked to servers via AdvanceDDE, XL_TABLE, and CF-TEXT protocols. In addition, RSSlider is a data bound control that can be bound to a database.

RSSlider offers complete control of the shape, size, and appearance of the slider through the Custom Properties Page.

Binding RSSlider to a Visual Basic Data control

RSSlider is a data bound control and can be bound to the Visual Basic Data control. The sample described below shows how RSSlider can be used to log data to a database and also update itself with the database values. The sample uses RSSlider to store RGB color values in a database when you click on the Save Mix button. Also, scrolling through the database causes the controls to update their values according to the data in the database.

During the installation, RSTools included the sample application described here. It is stored in Program Files\Rockwell Software\RSTools\Samples \SLIDDATA.VBP.

1. Open the SLIDDATA.VBP application in Visual Basic and display the SLIDDATA.FRM form.

2. The sample uses four RSSliders. Three of them are input controls and one is an output control that displays the tank level. For the three input RSSliders, verify the properties were set as follows:

- Bevel Style: 3-Beveled
- DataField: Red for RSSlider1, Green for RSSlider2, Blue for RSSlider3
- DataSource: Data1 (Name of the data control)
- DataUpdate: 2-DataSource
- DisplayCaptionVertically: True

The DataUpdate property sets the source priority for RSSlider. For example, if DataUpdate is set to DataSource then the data control gets priority over the DDE link source.
The red buttons on the RSSliders are regular Visual Basic Shape controls configured as circles with the fill color set to red. RSSlider can contain other controls inside it, just like a frame or a panel. These button-like shapes illustrate the use of StartMove and EndMove events. The colors of these shapes change to indicate when the slider starts to move and stops moving.

3. The fourth RSSlider acts as a level indicator for the vessel. The slider is configured not to accept any user input but can be controlled via code. Verify the properties for the fourth RSSlider are set as follows:

- CenterOnKnob: False
- DisplayCaption: False
- DrawDisabledShadow: False
- Enabled: False
- Scale1NumbersVisible: False
- Scale1Visible: False
- Shadow: False
- SliderType: 1-Vertical

4. Review the code for the RSSlider application included at the end of this procedure.

5. Run this sample and scroll through the database to see the changes take effect on the form as the data values change. To add a new record, move the RSSlider knobs to the desired values and click the Save Mix button.

**RSSlider code**

Option Explicit ' Declare all variables before using

'The following subroutine updates the colors of the various objects with the new RSSlider values 'and is declared Public so that it can be called from anywhere.

Public Sub UpdateColor()
    'Routine to update colors on the controls
    Dim iTemp As Integer
    Dim iRed As Integer
    Dim iGreen As Integer
    Dim iBlue As Integer
iTemp = 0
iTemp = Val(RSSlider1(0).Value) +
    _Val(RSSlider1(1).Value) +
    _Val(RSSlider1(2).Value)
'Calculate color values
iRed = RSSlider1(0).Value * 2.55
iGreen = RSSlider1(1).Value * 2.55
iBlue = RSSlider1(2).Value * 2.55
'Set RSSlider2 value that's a tank level marker
RSSlider2.Value = iTemp
'Set the colors for Shape and RSSlider2 objects
Shape3.FillColor = RGB(iRed, iGreen, iBlue)
RSSlider2.FaceColor = RGB(iRed, iGreen, iBlue)
End Sub

Private Sub Command1_Click()
    Dim sTemp0 As String
    Dim sTemp1 As String
    Dim sTemp2 As String
    'Save the control value before calling the AddNew method of the data control that resets the control values to zeros
    sTemp0 = RSSlider1(0).Value
    sTemp1 = RSSlider1(1).Value
    sTemp2 = RSSlider1(2).Value
    Data1.Recordset.AddNew
    'Set RSSlider's value to the saved value
    RSSlider1(0).Value = sTemp0
    RSSlider1(1).Value = sTemp1
    RSSlider1(2).Value = sTemp2
After updating the values, the record pointer returns to the last record that it was pointing to before the AddNew and Update methods were called. To reflect the last updated value on your controls, manually move the record pointer to the last record through code.

Private Sub Data1_Reposition()
    ' Refresh the colors when user scrolls through the database
    Call UpdateColor
End Sub

Private Sub Form_Load()
    Dim sDBName As String
    sDBName = App.Path & "\color.mdb"
    Data1.DatabaseName = sDBName
    ' Set the Data Control Properties
    Data1.DatabaseName = _
    "C:\RSWKSHOP\RSTOOLBX\DEMO\Color.mdb"
    ' You may change the path to point to the proper location of the database
    Data1.Connect = "Access"
    Data1.RecordSource = "Table1"
    Data1.Caption = Data1.DatabaseName
End Sub

Private Sub RSSlider1_Change(Index As Integer, ByVal _
    Value As Double, ByVal SliderIndex As Integer)
    ' Refresh control with new values when user moves the slider knob
    Call UpdateColor
End Sub
Private Sub RSSlider1_EndMove(Index As Integer, _
    ByVal Value As Double, ByVal _
    SliderIndex As Integer)
    'Turn the on indicator to red when RSSlider
    'stops moving
    Shape2(Index).FillColor = vbRed
End Sub
Private Sub RSSlider1_StartMove(Index As Integer, _
    ByVal Value As Double, _
    ByVal SliderIndex As _Integer)
    'Turn the on indicator to green when RSSlider
    'starts to move
    Shape2(Index).FillColor = vbGreen
End Sub
**RSGauge**

The RSGauge control gives your application a way to graphically display numeric data to make it easily understood by almost any user.

RSGauge displays a gauge that can graphically show danger and warning zones. The needle on the gauge moves as the data changes. RSGauge includes over 200 properties to allow immense flexibility when designing a gauge for a specific application. Properties allow you to select from numerous gauge styles, needle types, and scale options. The control also handles moving pictures, which give a unique animated visual display in response to changes in the gauge value.

RSGauge greatly increases DDE data transfers by communicating in Rockwell Software’s AdvanceDDE protocol. AdvanceDDE compresses a large amount of data and transfers it to and from your application in a block array of data.

**Displaying a warning area**

The RSGauge control can display a warning area that is similar to what is used on machine gauges. The procedure below creates a warning area.

1. Place an RSGauge control, an RSSlider control, a label and two RSButton controls on a Visual Basic form.

2. Set the following RSGauge properties:
   - WarningArea = True
   - CaptionY = 70
   - ZoneOffset = 75

3. Set the following RSButton properties on both RSButton controls:
   - ButtonStyle = 6 - Diode
   - Visible = False

4. Set the RSButton1’s LightOffColor property to &H0000FFFF& (yellow).

5. Set the following RSSlider properties:
   - AllowChangeEvent = True
   - WriteStyle = 1 – Continuous
6. Enter the following code:

Private Sub RSSlider1_Change(ByVal Value As Double, ByVal SliderIndex As Integer)
    RSGauge1.Value = RSSlider1.Value
    ' The RSSlider control's value will now position the RSGauge control's needle
    Label1.Caption = RSGauge1.InWhichZone(0)
End Sub

Private Sub RSGauge1_EnteringDangerZone(ByVal EnteringDanger As Double, ByVal GaugeIndex as Integer)
    RSButton2.Visible = True
End Sub

Private Sub RSGauge1_EnteringCautionZone(ByVal EnteringCaution As Double, ByVal GaugeIndex as Integer)
    RSButton1.Visible = True
End Sub

Private Sub RSGauge1_LeavingDangerZone(ByVal LeavingDanger As Double, ByVal GaugeIndex as Integer)
    RSButton2.Visible = False
End Sub

Private Sub RSGauge1_LeavingCautionZone(ByVal LeavingCaution As Double, ByVal GaugeIndex as Integer)
    RSButton1.Visible = False
End Sub

7. Run the program.

When the program runs, moving the RSSlider control changes the value of the RSGauge control. Whenever the gauge value enters either the caution zone or danger zone, the appropriate RSButton LED lights. The Label control's caption shows the result of the InWhichZone method which is 0 for the OK Area, 1 for the Caution Zone, and 2 for the Danger Zone.
RSVessel

The RSVessel control is a free-form dynamic-fill shape control. Use RSVessel to display data from a database or server. The RSVessel control also can be configured to act as a button control.

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

The free-form shape editor allows you to add and remove nodes/sides, 360 degree rotation, and bezier curve segments. The shape editor 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.

Using RSVessel with the Microsoft 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. The example below uses the block DDE method.

1. Place an RSVessel control on a form in Visual Basic.

2. Set the following properties:
   - AllowChangeEvent = True
   - DisplayValue = False
   - FillEndX = 75
   - FillEndY = 50
   - FillStartX = 25
   - FillStartY = 50
   - LinkServer = Excel
   - LinkTopic = [Book1]Sheet1
   - LinkItem = R1C1:R8C1
   - LinkMode = 1 - Automatic

3. Place a Grid control on the form to the right of the RSVessel control.
4. Set the following properties for the Grid control:
   Rows = 2
   Cols = 4
   FixedRows = 0
   FixedCols = 0

5. Enter 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

6. Start Microsoft Excel and run the Visual Basic program. As you enter values into cells A1 - A8 in Excel Sheet 1, the values display in the RSVessel control and in the Grid control. Because the RSVessel LinkItem (R1C1:R8C1) is an array of 8 data values, the RSVessel control splits itself into eight separate controls, each displaying data from a different Excel cell.
RSCompare

RSCompare is a data comparison multi-state control. Set the shape of the control and use it to display, log, and receive data that is to be compared. Its multiple state shape editor allows addition and removal of nodes and sides, 360 degree rotation, and bezier curve assignments. Its three-state capability allows you to view data trends in either a last point/this point mode or in an absolute mode. Absolute mode allows you to view the status of data with respect to a check point value. In addition to three-state graphic shapes, you may also use RSCompare’s three-state picture switching by assigning a bitmap to each of the three states (<= >).

Logging a value to a data source

One of the biggest advantages of the RSCompare control is its ability to log the value displayed to the data source specified by the 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 containing at least one entry or record. In the database, you must specify a field name for the data point you want to log. You may have several field names, depending upon how many data points you have.

Follow the steps below to set up the RSCompare control for data logging. You may want to use the DBFLOG.MDB database that came with RSTools, which is located in C:\Program Files\Rockwell Software\RSTools\Samples.

1. Place a Visual Basic Data control on a new form.
2. Set the following properties for the Data control:
   - Connect = Access
   - DatabaseName = C:\Program Files\Rockwell Software\RSTools\Samples\Dbflog.mdb
   - RecordSource = LogTable
3. Place an RSCompare control on the form.
4. Set the following properties for the RSCompare control:
   - AllowChangeEvent = True
   - DataSource = Data1
   - DataField = FIELD_1
   - ValuePadDownload = True
   - CheckValue = 250
   - EndValue = 500

5. Enter the following code:

   Private Sub RSCompare1_Change(ByVal Value As Double, ByVal CompareIndex As Integer)
   Dim sTemp As String
   sTemp = RSCompare1.Value
   Data1.Recordset.AddNew
   RSCompare1.Value = sTemp
   Data1.Recordset.Update
   Data1.Recordset.MoveLast
   End Sub

6. Run the program.

   Each time you click the RSCompare control, a number pad allows you to enter a new value for RSCompare. The RSCompare control value is updated, and at the same time the new value is written to the database. You can test this by entering a series of numbers and then scrolling through the values in the database using the Data control’s arrow buttons. Each new value that is entered logs a new record in the database.
RSAnimator

The RSAnimator control is a behind-the-scenes control that stores an ActiveX control’s property settings in a step-by-step manner to create powerful animations. An animation is a collection of frames (pictures) that are played back in order. In each frame, some physical attribute of an object is changed by a small amount so when the frames are played back, there is an illusion of motion.

With RSAnimator, choose the number of steps (frames) you want in your animation. Then change some physical attribute of an object for each step in the animation. After changing the physical attributes, choose how the animation will be played back.

RSAnimator creates animations that are much smaller than similar animations created by other currently available animation creating software. The Snapshot feature allows you to position, change the size, or set any property of an ActiveX control and then take a picture to save the position, size or property settings for a particular step.

RSAnimator can be controlled to go through its steps by any of the following ways:

- Use the Auto Animate feature which causes RSAnimator to cycle through its steps automatically at any speed you designate.
- Use a DDE link where each step responds to a specific range of values.
  RSAnimator is designed to be used with any Dynamic Data Exchange (DDE) server and provides enhanced performance when used with the AdvanceDDE protocol.
- Write code.

Using the Snapshot feature to create animation

The procedure on the following page uses the Snapshot feature of RSAnimator to set up an actual animation. The Snapshot feature allows you to position, change the size, or set any property of an ActiveX control, and then take a picture to save the property settings for a particular step in the animation. You also can use the Snapshot feature to step through an animation to see what animated effects have already been created.

The procedure uses the RSGauge control, but you can use any ActiveX control that is registered as an ActiveX control in the Windows registry. This includes most third-party ActiveX controls, but does not include the standard Microsoft Visual Basic controls like text boxes, labels, and picture boxes.
1. Add an RSAnimator control and RSGauge control to a Visual Basic form.

2. Right-click when the cursor is positioned on the RSAnimator control to open the Custom Properties Page.

3. On the General tab, enter 10 in the Of field.

![RSAnimator1 Properties](image)

4. Click the Auto Animate check box. When the program is run later, RSAnimator cycles through the steps automatically at the speed designated in the Speed field (250 milliseconds in this example).

5. Choose RSGauge1 from the Control box.

Next, you will add properties to the Animation Properties listbox.

6. Choose BackColor from the Property box and then click the button to add the property to the Animation Properties box.

7. Add the CenterWidthHeight property and the CenterXY property to the Animation Properties box using the same procedure as in step 6.

8. To open the Animate Snap dialog box, click the button.
9. Make sure the # (Current Step) field is set to 1 and the Animate check box is not marked.

If the Animate check box is marked when a snapshot is taken, the step number automatically increments.

10. Move the position of the RSGauge control to another location on the form and resize the control.

11. Change the BackColor property of the RSGauge control using the Property list on the right side of the window.

12. Click the Snapshot button on the Animate Snap window.

13. Repeat steps 10 through 12 until the # (Current Step) field in the Animate Snap dialog equals 10.

14. Press the Run button to run the animation.

**IMPORTANT** When you close the Animate Snap window, RSAnimator saves all changes to the controls made by RSAnimator.
RSAlarm

RSAlarm is an ActiveX Custom Control (OCX) that allows you to monitor and control alarms. The RSAlarm control provides a graphical alarm management tool that helps you handle alarm conditions and log alarms to a database. The control gives you the ability to prioritize alarms. Using RSAlarm, you also can display alarms in different colors in a list box that provides an easy-to-use interface for the user. RSAlarm has custom events and methods that notify you when an alarm has occurred or has been acknowledged.

Creating an application using RSAlarm

During the RSTools installation, Rockwell Software included a sample application that allows you to view some of the features offered by RSAlarm. It provides examples of how to set RSAlarm properties, methods, and events to control alarms and log alarms to a database. The sample application is stored in \\Program Files\\Rockwell Software\\RSTools\\Samples\\RSAlmSmp.exe.
The sample application illustrates how to add alarms at runtime using the AddAlarm method. The AddAlarm method uses the AlarmAction parameter to specify from where you want the alarm text or index to be read. The AlarmText string is constructed based on the AlarmAction settings and displayed in the AlarmText edit box. The example provides a good representation of how your AlarmText string should be set for the corresponding AlarmAction.

This example also demonstrates how RSAlarm can automatically log alarm information to a database when bound to a Visual Basic Data control. The LogAlarms property specifies when the information will be logged.

To use the sample application included with RSAlarm:

1. Start the application by double-clicking on RSALMSMP.EXE in Windows Explorer. It is located in \Program Files\Rockwell Software\RSTools \Samples\RSAlmSmp.exe

2. To begin logging data, pick the appropriate choice from the list of choices in the Log To Database combo box. RSAlarm data properties are set at runtime to bind it to the database.

3. Click the buttons on the window to Add, Acknowledge, Remove, and Clear all alarms from the list. Clicking these buttons calls RSAlarm methods to perform an action.
RSEventMaster

RSEventMaster gives your application the ability to create complex event handling and notification without having to write extensive Visual Basic code. For example, instead of having to write a global keyboard event handler to trap certain keystroke combinations and perhaps and/or them with other standard VB and mouse events, the RSEventMaster can be easily configured for such purposes.

The RSEventMaster control is a master event handler that can be configured to trigger a master event procedure when all events are true or have occurred.

Integrating RSAlarm and RSEventMaster

RSEventMaster can integrate directly with Rockwell Software’s RSAlarm control to provide automatic alarming functionality to your application. When all defined subevents are set to true and the EventHappened event fires, RSEventMaster can send messages to RSAlarm or request messages from a DDE server and pass them on to RSAlarm.

The procedure below describes how to link RSEventMaster to RSAlarm.

1. Place an RSEventMaster and RSAlarm control on a form in Visual Basic.
2. Right-click the RSEventMaster control to open the Custom Properties Page.
3. On the General tab, click the Keyboard tab.
4. Select F from the Key box and then click Add to add it to the subevents list.
5. Click the Alarm tab.
6. Select the name of the RSAlarm control from the AlarmControl combo box.
7. Enter 0 (zero) in the AlarmPriority field.
8. Select 0 – Display String from the AlarmAction box.
9. Enter “You pressed the F key!” in the AlarmActionData field.
10. Click OK.
11. Run the application.
12. Press the F key on the keyboard.
Each RSTools control has its own help file. The online help provides information on all properties, events, and methods. It also describes how to use the RSTools controls. Common information is shared among the help files.

To access a table of contents for the online help, click the icon for the RSTools control in the Visual Basic toolbox (list of icons) or click on the RSTools control on the form. Then press F1.

Double-click any book to open it and display help topics. Double-click any help topic to view the help.
To access help for a property, select the RSTools control on the form. Click the property in the Properties window and press F1. A list of properties appears as shown below. Click the property you want to view help on from the list.

Click on the property name to display information about the property.
To open help for methods or events, select Properties, Events, and Methods from the Contents tab in the online help. Then select “Use methods” or “Use events” from the list of topics.

**Technical Support**

If you have questions about RSTools after consulting this guide and the RSTools online help, contact Rockwell Software Technical Support:

- Technical Support Telephone—440-646-7800
- Technical Support Fax—440-646-7801
- World Wide Web—www.software.rockwell.com
Solving problems

This chapter describes some common problems you may experience and tells you how to solve them.

When distributing an RSTools application, the following message appears during the installation:

“Unexpected error: Quitting”

Your computer has version 5.x of Olepro32.dll installed, but does not have version 2.20.xxx of Oleout32.dll installed.

Install an application that includes the new Oleout21.dll. You can get the dll from the Visual Basic 4.0 Setup Toolkit, Internet Explorer 3.0, Visual Basic 5.0 CC edition, Office 97, Windows NT 4.0, or Windows NT 3.51 Service Pack 3.

After installing the dll, you must recreate your installation by copying the following files to your C:\Program Files\Rockwell Software\RSTools from the \Visual Basic\System directory:

OLEPRO32.DLL
MFC40.DLL
MSVCRT40.DLL
The Custom Property Page does not display all the tabs

To solve this problem, you must manually register the following dlls:

- **Link/Display tab**: RSTOOLED.DLL
- **Font tab**: OLEPRO32.DLL
- **General and Value tabs**: RSBTN32.DLL, RSCMPX32.DLL, RSDATX32.DLL, RSGGEX32.DLL, RSSLDX32.DLL, RSVSLX32.DLL, RSWHLX32.DLL, RSPWRTLX.DLL

To manually register an ActiveX file, run the Regsvr32.exe program with the file path and name as the command line parameter. For example:

```
Regsvr32.exe c:\winnt\system32\rstooled32.dll
```

The RSTools activation is on the computer, but the tools generate copy protection errors when dropped into a form

Uninstall RSTools from the computer. Restart the computer, and then reinstall the correct version of RSTools. Do not just reinstall.

You receive the following message:

**“Attempt to establish a conversation has failed”**

One of the following has occurred:

- DDE/Data server is not running.
- LinkTopic does not exist or you mistyped it.
- LinkItem does not exist or you mistyped it.
- You are using the Lite version of RSLinx or Wintelligent LINX.
- You are using the OEM version of RSLinx without RSJunctionBox.
Index

Activating RSTools, 3, 4
Adding RSTools to a VB project, 6
Copyprot.hlp, 4
DATALOGDEMO.VBP, 11
Events, 37
Features, 1
FrontPage
   Using RSTools, 9
Help, 35
   Methods, 37
Installation, 3
Introduction, 1
Methods, 37
POKENREQUEST.VBP, 13
Properties, 36
RSAlarm, 2, 31, 33
   Creating an application, 31
RSAnimator, 2, 28
   Snapshot feature, 28
RSButton, 2, 16
RSCompare, 2, 26
   Logging a value to a data source, 26
RSData, 2, 11
   Logging data to a database, 11
   Poking and requesting data, 13
RSEventMaster, 2, 33
   Integrating RSAAlarm, 33
RSGauge, 2, 22
   Displaying a warning area, 22
RSJunctionBox, 2
RSSlider, 2, 17
   Bind RSSlider to a VB Data control, 17
RSVessel, 2, 24
   Grid control, 24
RSView32, 5
RSWheel, 2, 14
   Viewing different wheel types, 14
SETUP.EXE, 3
SLIDDATA.VBP, 17
Solving problems, 38
Support, 38
System requirements, 2
Template, 7
Troubleshooting, 4, 38
Visual Basic
   Adding the RSTools control, 6
   Using RSTools, 7

Spare Allen-Bradley Parts