

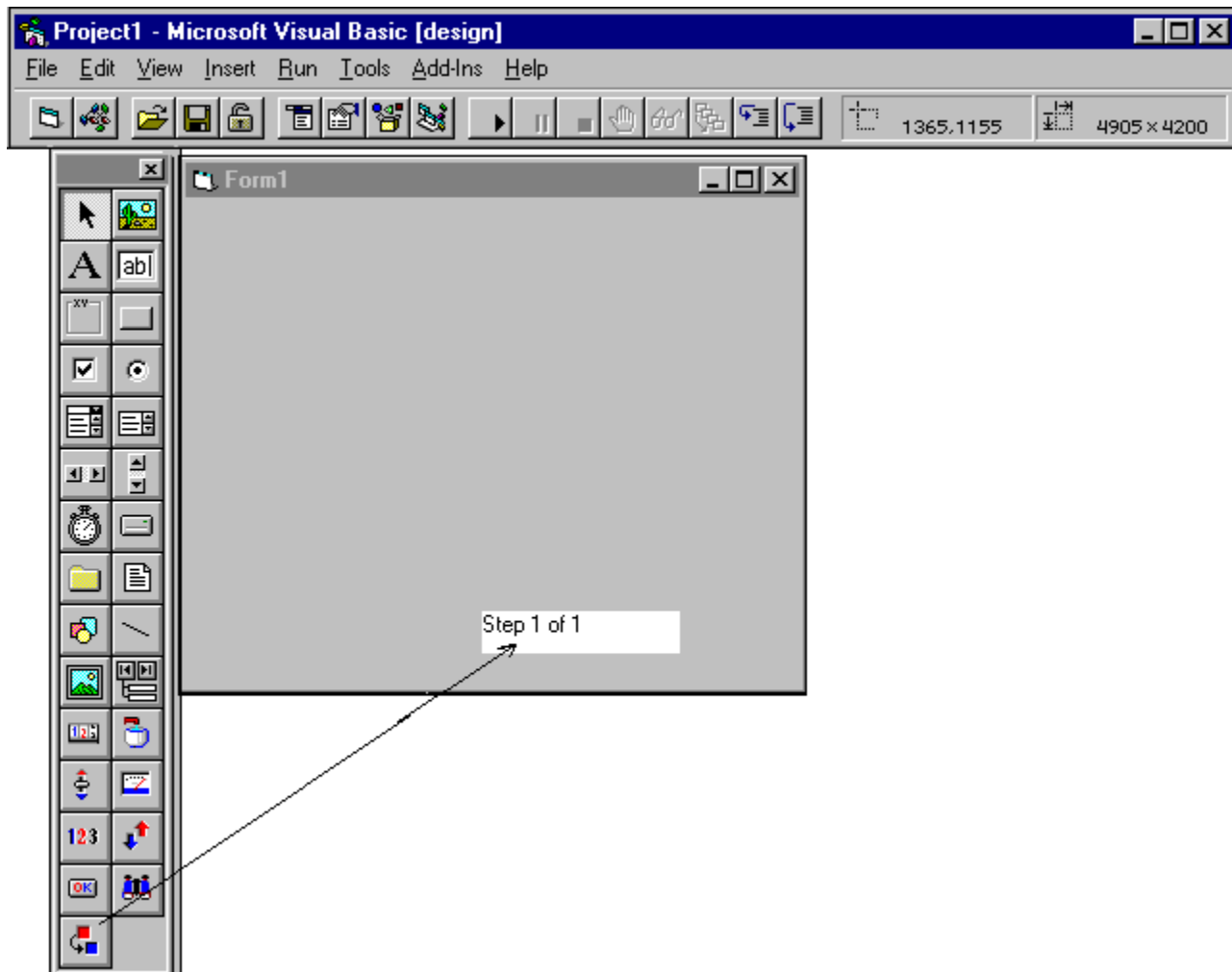
## Introduction of the RAnimator

[Properties](#)

[Events](#)

[Methods](#)

The RAnimator control is a behind the scenes control that stores OCX control's property settings in a step by step manner to create powerful animations. An animation is basically just 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 a feeling of motion. The RAnimator control does exactly this. With the RAnimator you choose the number of steps (frames) you want in your animation. Then you change some physical attribute of an object for each step in the animation. Then you choose how the animation will be played back. This is all done in a user friendly manner and the resulting animation files are much smaller than similar animation files created by other animation creating software available today.



**File Name** RSPWRTL.OCX

**Object Type** RAnimator

**Remarks** The RAnimator control has many custom properties and other features to allow you immense flexibility when creating a unique animation for your particular application

**Note** *When you create and distribute applications that use the RAnimator control, you should install the RAnimator.OCX file in the WINDOWS \ SYSTEM subdirectory. The Setup Wizard included with Visual Basic provides tools to help you write setup programs that install your applications correctly.*

## Description

- The RSAnimator is a Visual Basic Active X Control (OCX).
- The RSAnimator control is a behind the scenes control that stores numerous OCX control property settings in a step by step manner to create powerful animations. The resulting animation file is much smaller than similar animation files produced by other currently available animation creating software.
- The RSAnimator QkMover custom property tab is a quick and easy way for you to configure and save property settings that are used to plot paths for OCX controls to travel during an animation.
- The RSAnimator QkSizer custom property tab is a quick and easy way for you to configure and save property settings that are used to change the size of OCX controls during an animation.
- The RSAnimator QkColor custom property tab is a quick and easy way to configure and save property settings that are used to change color properties for OCX controls during an animation. The QkColor tab also includes a special feature which quickly creates color gradients for a property.
- The RSAnimator QkCaption custom property tab is a quick and easy way for you to configure and save property settings that are used to change the Captions of OCX controls during an animation. The QkCaption tab also includes a special feature which makes the Caption of a control appear in a scrolling manner.
- The RSAnimator Snapshot feature allows the user to position, change the size or set any property of a control and then take a picture to save the property settings for a particular step. This feature also includes an easy way for you to step through the animations to see what animated effects have all ready been created.
- The RSAnimator control is designed to be used with any Dynamic Data Exchange Server. Using the RSAnimator control with RSJunctionBox (purchased separately) will allow the use of Rockwell Software's AdvanceDDE protocol and also the XL\_Table protocol for communication with the RSAnimator control.

## **Installing the RSAnimator Control**

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

## System Requirements

Before you install RSAnimator, make sure that your computer meets the minimum system requirement. 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

5 MB free hard disk drive space

A 3.5" floppy drive

VGA Graphics Card

Microsoft Visual Basic 4.0

Windows NT 3.51 or Windows 95

### Recommended Requirements

IBM compatible Pentium

16 MB of RAM

15 MB free hard disk drive space

Color (800x 600) or (1024 x 768) display

Microsoft Mouse compatible pointing device (mouse, trackball, touchscreen, etc.)

Microsoft Visual Basic 4.0, Professional Edition

Microsoft NT 3.51 or Microsoft Windows 95

## **RSANIMATOR.WRI File**

The RSANIMATOR.WRI file lists any last minute changes to the RSAnimator documentation, Help file and to the RSAnimator control. This file will be placed in the **RSWKSHOP\STOOLBX** directory during setup. To read the file, open the Windows Write application and open the **RSANIMATOR.WRI** file or double-click on the file after locating it using Windows Explorer.

## **Running Setup**

When you run the setup program, you will have to set the path for the RSAnimator control.

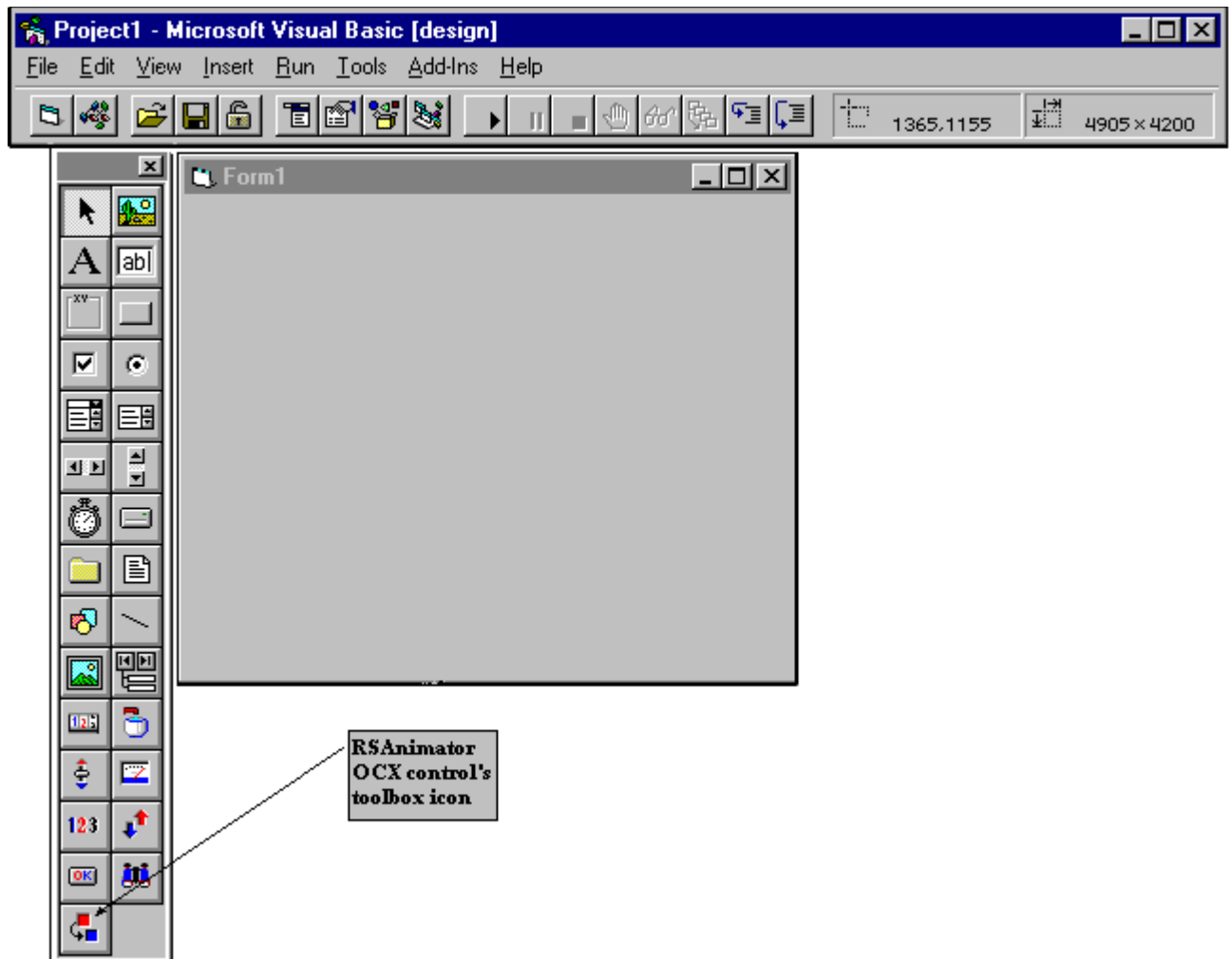
### **To Start Setup:**

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.

## Loading the RSAnimator Control to the VB Toolbox

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

1. Start Visual Basic.
2. From Visual Basic, select the Tools/Custom Controls menu item.
3. Scroll down the list box until “Rockwell Software RSPowerTools” is visible. Select it and choose OK.  
(You may have to click the browse button and manually search for the RSPWRTL.S.OCX, which is located in the C:\WINDOWS\SYSTEM folder). The control is added to the toolbox for the project.



The RSAnimator control's icon is added to the Visual Basic Toolbox.

## **Autoloading the RSAnimator Control to the VB Toolbox**

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

To configure Visual Basic to automatically load RSAnimator:

1. Start Visual Basic and choose File\Open Project from the menu.
2. Open the AUTO32LD.VBP project. (Found usually in the Microsoft Visual Basic Folder)
3. Choose Custom Controls from the Tools menu. (The “Custom Controls” Dialog appears.)
4. Scroll down the list box until “Rockwell Software RSAnimator” is visible. Select it and choose OK. (You may have to click the browse button and manually search for the RSGge32.OCX and RSGGEX32.dll, which are located in the C:\WINDOWS\SYSTEM folder). The control is added to the VB toolbox for the project.
5. Choose Save Project from the File menu.



## Placing the RAnimator Control on a Form

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

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

**or**

3. Double click on the RAnimator toolbox icon and a new RAnimator control will be placed in the center of the form.

## Using Help

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

### **To access the help contents page:**

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

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

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

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

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

## **Distributing RAnimator 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:

<b>File</b>	<b>Description</b>
RSPWRTLS.OCX	Code for the RSAnimator Control
RSTOOL32.DLL	Common Code for the Control
RSCALC32.DLL	Calculation Engine for Read / Write Expressions (Available only when RSJunctionBox is installed.)

If your application is using the RSJunctionBox module at run time, you will also need to include:

RSJBOX32.DLL	RSJunctionBox Module
RSJBP32.DLL	RSJunctionBox Protection DLL

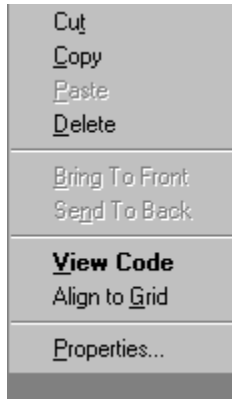
## Sample Applications

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

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

## Visual Basic Floating Menu

The Visual Basic Floating menu, shown below, will be displayed when you click the right mouse button while the pointer is positioned over the RSAnimator control. See the table below for a description of the menu items. A major function of this menu is to display the Custom Property Tabs (shown on the following page). This occurs when you click on the Properties... menu item.



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

## Using Custom Property Tabs

The RSAnimator control's Custom Property Tabs, shown below, are a new simplified way for you to adjust most of the RSAnimator control's properties and also save OCX control's property settings that are used during an animation.

Each Tab page provides a different set of options for rendering an animation. As you click each tab, the controls on the page change to allow you to configure a different set of options. Options that are not available for the current RSAnimator settings or situation are greyed.

**Rockwell Animator Control Properties** [X]

General | LinkInfo | QkMover | QkSizer | QkColor | QkCaption

Step: 1 Start: 0 # ☒ Auto Value  
Of 1 End: 100 # ☐ Auto Animate  
Value: 0 # Speed: 250

[+]  
[-] [Target Icon]

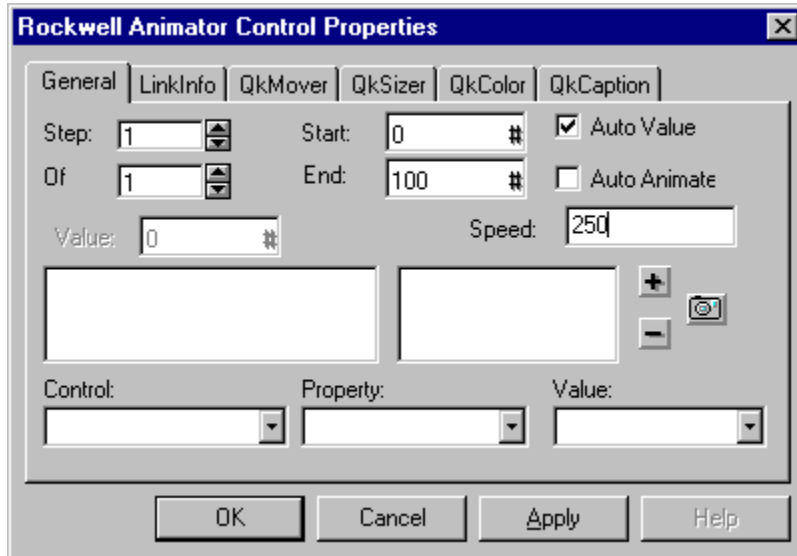
Control: Property: Value:

[OK] [Cancel] [Apply] [Help]

**Note:** Switching between tabs will force an automatic apply

## General Tab for RSAnimator

This tab page is where you set the RSAnimator properties that are used to control an animation. This tab page also gives the user access to individual properties of a specific control that can be manipulated during an animation.



- Step** The current step in the animation. Use the spin buttons to increase or decrease the current step value or type a step value into the box.. (See also [CurrentStep](#) property.)
- Of** The total number of steps in the animation. Use the spin buttons to increase or decrease the total number of steps value or type a step value into the box. (See also [NumberOfSteps](#) property.)
- Value** The value of the RSAnimator that is used for communication between DDE links. The value could be considered the current step value in most instances.
- Start** The starting value for the range of values for steps in the animation.
- End** The ending value for the range of values for steps in the animation.
- AutoValue** When this box is marked the difference between the start and end value will be divided by the number of steps in the animation to produce a step value.
- AutoAnimate** When this box is marked it will cause the RSAnimator to cycle continuously through its steps automatically, at the speed designated in the Speed box described below. (See also [AutoAnimate](#) property.)
- Speed** When the AutoAnimate check box is marked the RSAnimator will cycle through each step in the animation at this speed designated in milliseconds. (See also [AutoAnimateSpeed](#) property.)
- Animation Properties List boxes** These boxes show the actual properties and their current step values for the specific controls used in the animation. When using any of the Qk tabs, properties are automatically added to these boxes which will be used to change a control's property settings during an animation. To set up to monitor other control properties choose a control and property using the combo boxes at the bottom of the General Tab. Set the value you want for the property and then press the + button to add the property to the Animated properties list box. To delete a property from the Animated property list box, highlight the appropriate property and press the - button.
- Control, Property and Value combo boxes** These boxes are used to add or change control settings that will be used during the animation. The settings are added to the Animation Properties list boxes described above. To add a control's property to be used in the animation choose one of the available controls from the drop down list labeled Control. Then choose the property to be used in the animation from the

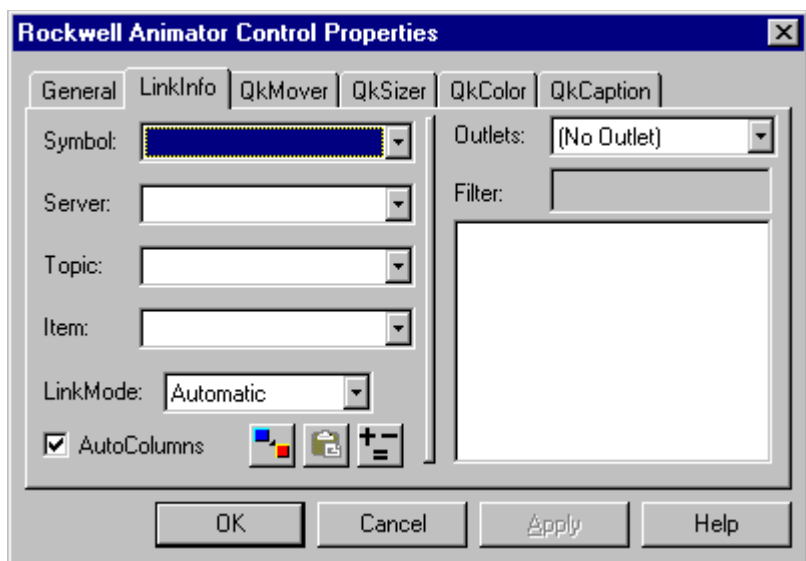


drop down list labeled Property. Then choose a Value for the property and add it to the list box. Finally press the + button to add the new settings to the Animation Properties List boxes discussed above. When the new settings are added they will be added to every step of the animation. To change a value for a property in the Animation Properties list box, go to the appropriate step using the Step box described above, highlight the property in the Animation Properties List box, change the current specified value using the combo box labeled Value and then press the + button.

**Snapshot button** This feature allows you to position, change the size or set any property of a control and then take a picture to save the property settings for a particular step in the animation. This feature also includes an easy way for the user to step through the animations to see what animated effects have all ready been created. To enable this button, at least one control's property must be added to the Animation Properties list box. When a snapshot is taken only the properties shown in the Animation Properties list box will be saved.

## LinkInfo Tab for RSAnimator

This tab page is where you can link the RSAnimator control to another application. If you have not purchased and installed the RSJunctionBox software, then the right side of the page will be blank.



**Symbol.** This combo box is where you can choose the name of a Symbol, that has been previously defined for a specific Server, Topic and Item, in a DDE link. To define Symbols for different DDE links see the section on Manage Symbols to follow. If the Symbol has been previously defined select it from the combo box list or enter the name of the Symbol in the combo box directly. When a defined Symbol is chosen then the Server, Topic and Item combo boxes on the LinkInfo page will automatically be filled in with the appropriate data. (See also [Symbol](#) property.)

**Server.** This combo box determines the application or Server name that the RSAnimator control is linked to. If the Server name has been previously used it can be chosen from the combo box list or enter the name of a new Server into the combo box directly. (See also [LinkServer](#) property.)

**Topic.** This combo box determines the Topic portion of the data link string, which the RSAnimator control uses for addressing in a DDE link. If the topic has been previously used it can be chosen from the combo box list or enter the name of a new Topic into the combo box directly. (See also [LinkTopic](#) property.)

**Item.** This combo box determines the Item portion of the data link string, which the RSAnimator control uses for addressing in a DDE link. If the Item has been previously used it can be chosen from the combo box list or enter the name of a new Item into the combo box directly. (See also [LinkItem](#) property.)

**AutoColumns.** This check box is where you can enable or disable the AutoColumns property for the control. Some DDE servers can provide data in block format. For example, a 5 by 5 block of data can be read from an Excel spread sheet with the following syntax for the ITEM:

R1C1:R5C5

The AutoColumns property will automatically split the blocked data into columns to fit the width of the control. (See the RSAnimator Property Reference Chapter to locate more information on the AutoColumns property.)

**LinkMode.** Sets the type of link to be used for a DDE conversation and activates the connection. Available options for the LinkMode property are:

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

**1 = Automatic** - A "HOT link". The **Server** automatically updates the control when data changes.

**2 = Manual** - A "COLD link". The client has to issue a DOREQUEST method to get data from the **Server**.

**3 = Notify** - Causes Visual Basic to Fire the LINKNOTIFY event which the user can respond to.



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

To define a new Symbol for a specific Server, Topic, and Item in a DDE Link, place the name for the new Symbol in the text box labeled Symbol. Then place the DDE Link Server name in the text box labeled Server, the DDE Link Topic name in the text box labeled Topic, and the DDE Link Item name in the text box labeled Item.

Press the **Add/Change** button to add the new Symbol name to the Symbol list box.

Press **Delete** to delete an existing 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.

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



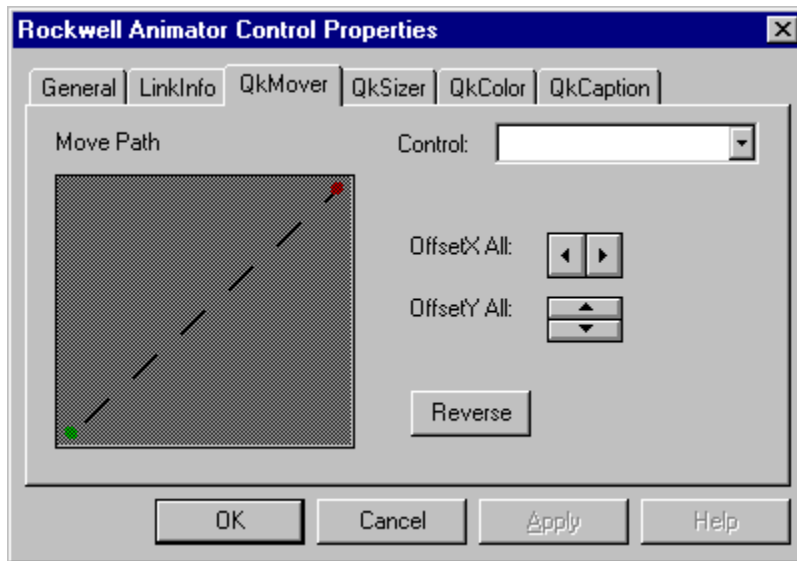
**Paste Link.** This button is enabled when the user has copied specific DDE Link information from another application (example: Excel) to the clipboard. Pressing this button will then fill in the appropriate DDE Link information for the Server, Topic and Item.



**Expression.** (This feature is available when the RSJunctionBox is installed on the computer.)

## QkMover Tab

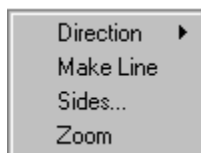
This tab page is a quick and easy way for you to set up and save property settings that are used to plot paths for OCX controls to travel during an animation..



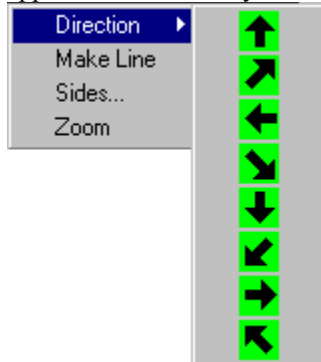
**Control:** This combo box is where you choose a control to be configured so it will be moved during an animation. The available controls on the form that the RSAnimator can move is shown in the drop down list. When a control has been chosen the Move Path box will be enabled.

**Move Path** This box is where you plot the path that the chosen control will travel during the animation. The green dot represents the starting position and the red dot represents the ending position for the path that the control will travel during the animation. The black dots represent the rest of the steps that the chosen control will travel during the animation. The number of black dots is dependant on the total number of steps in the animation which is determined on the General custom property tab.

**Move Path Menu** The menu shown below is available when you press the right mouse button while the mouse pointer is positioned over the Move Path box.



**Direction >** When the mouse pointer is positioned over the Direction > label an additional menu, shown below, appears which allows you to quickly assign a move path to a chosen control.



If you select one of the directions shown on the menu the chosen control will move from one side of its container to the other side in the direction specified.

**Make Line** This menu item allows you to create a line between the starting and ending points after they have been positioned inside the Move Path box. To perform this operation position the green and red dots to the appropriate position in the Move Path box and click on the Make Line menu option. The rest of the steps in the animation will be placed in a uniformly spaced line between the green and red dots.

**Sides** This menu item allows you to quickly assign a move path to a control in the form of known geometric shapes. Sides describes the number of sides in the geometric shape. For example 3 sides designates a triangle, while four sides designates a square and so on. To use this feature the total number of steps in your animation must be at least the number of sides you require. If you have more total steps than the Sides value you require the extra steps will be distributed along the path of the shape chosen.

**Zoom** This menu item gives you a view of the actual container that the chosen control will be moving in. This container could be a form or another control. This view is the one you should use when plotting a move path manually for a control because the actual size and position of the container is shown.

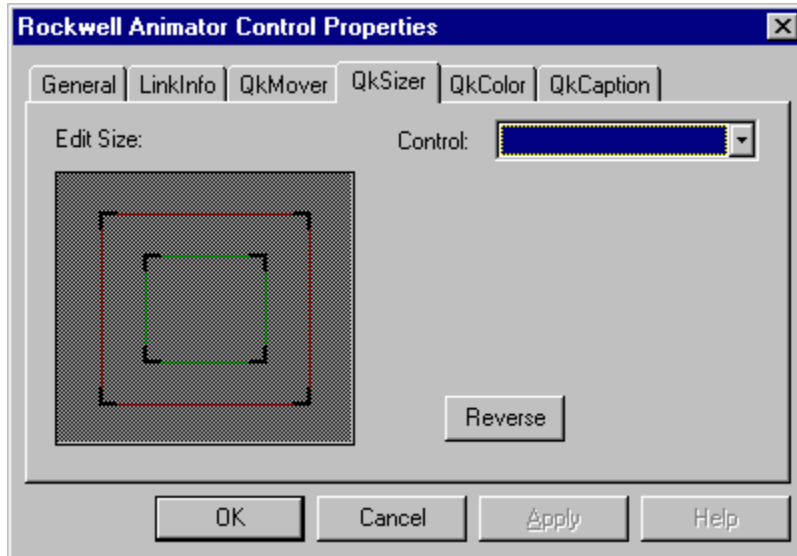
**OffsetX all** These spin buttons are used to move all the points in the Move path box by the same distance in either the left or right direction.

**OffsetY all** These spin buttons are used to move all the points in the Move path box by the same distance in either the up or down direction.

**Reverse** This button is used to reverse the starting position (green dot) and the ending position (red dot) for the move path of the chosen control.

## QkSizer Tab

This tab page is a quick and easy way for you to set up and save property settings that are used to change the size of OCX controls during an animation.



**Control:** This combo box is where you choose a control that will be configured so its size will be changed during an animation. The available controls on the form that the RSAnimator can size is shown in the drop down list. When a control has been chosen the Edit Size box will be enabled.

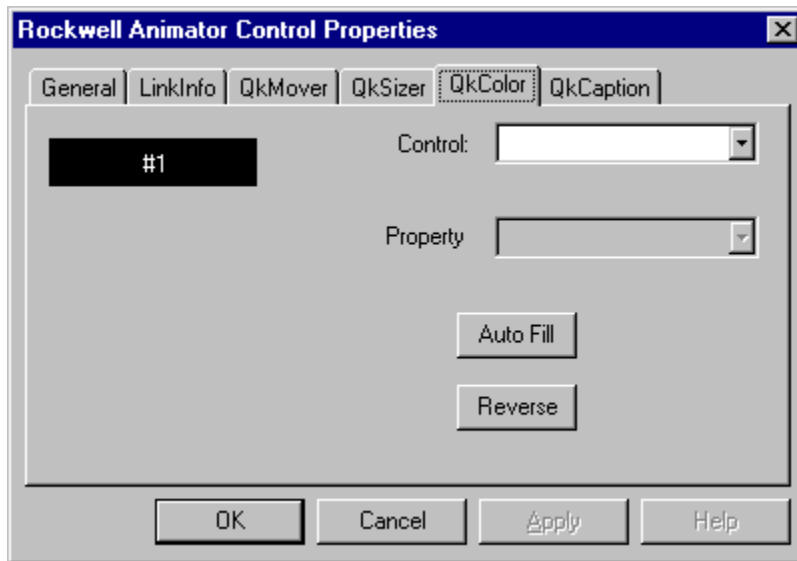
**Edit Size** This box is where you set up the starting and ending sizes for the chosen control that will be used during the animation. The green box represents the starting size and the red box represents the ending size for the control. The size of the chosen control will change in size incrementally, depending on the total number of steps in the animation, between the start and end sizes.

**Zoom** This menu item gives you a view of the actual container that the chosen control will be changing its size in. This container could be a form or another control. This view is the one you should use when setting up the starting and ending sizes because the actual size and position of the container is shown.

**Reverse** This button is used to reverse the starting size (green box) and the ending size (red box) of the chosen control.

## QkColor Tab

This tab page is a quick and easy way for you to set up and save property settings that are used to change color properties for OCX controls during an animation



**Control:** This combo box is where you choose a control that will be configured so that one of its color properties will be changed during an animation. The available controls on the form that the RSAnimator can manipulate are shown in the drop down list. When a control has been chosen the Property combo box will be enabled.

**Property** This combo box is where you select a color property for the chosen control to have changed during the animation.

**Color Boxes** The box in the upper left with the number #1 is where the color is assigned to the selected color property of the chosen control. There is only one box because only one step was assigned for the total number of steps in the animation (Assigned on General Tab). The more steps there are in the animation the more color boxes there will be. When click on any of the boxes a color palette will appear allowing you to assign a color to the selected property for that step in the animation.

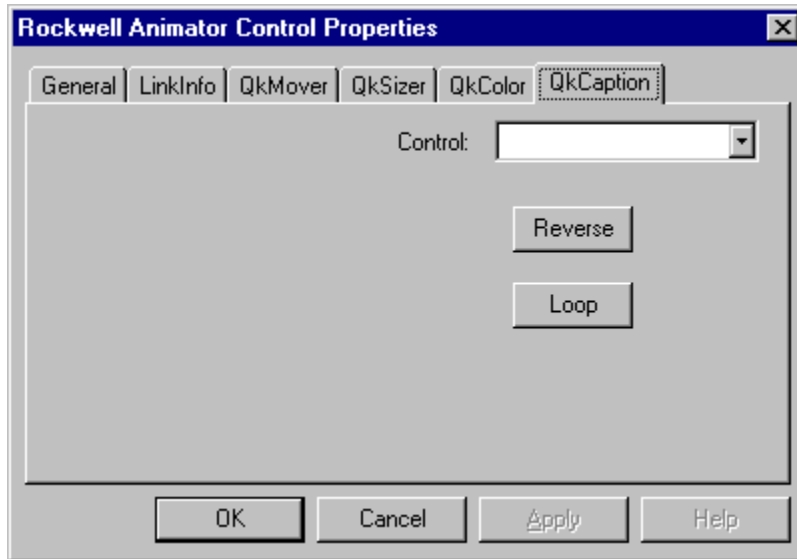
**Auto Fill** This button is used to set up a color gradient between the start and end color that has been assigned to a selected color property of a chosen control.

**Reverse** This button is used to reverse the order of the steps.



## QkCaption Tab

This tab page is a quick and easy way for you to set up and save property settings that are used to change the captions of OCX controls during an animation.



**Control:** This combo box is where you choose a control whose caption will be changed during an animation. The available controls on the form that the RSAnimator can manipulate are shown in the drop down list. When a control has been chosen the Property combo box will be enabled.

**Caption text boxes** These boxes will appear when a control has been chosen whose caption will be changed during the animation. These text boxes are where you type the text you want to be displayed for each step in the animation. Each text box represents a different step in the animation.

**Loop** This button is used to automatically create a looping caption for the chosen control during the animation.

**Reverse** This button is used to reverse the order of the steps.

## Property List for RAnimator

<a href="#">(About)</a>	<a href="#">DragIcon</a>	<a href="#">LinkItem</a>	<a href="#">TabIndex</a>
<a href="#">(Custom)</a>	<a href="#">DragMode</a>	<a href="#">LinkMode</a>	<a href="#">TabStop</a>
<a href="#">AutoAnimate</a>	<a href="#">Enabled</a>	<a href="#">LinkServer</a>	<a href="#">Tag</a>
<a href="#">AutoAnimateSpeed</a>	<a href="#">EndValue</a>	<a href="#">LinkTopic</a>	<a href="#">Top</a>
<a href="#">AutoValue</a>	<a href="#">ExpressionForRead</a>	<a href="#">Name</a>	<a href="#">Value</a>
<a href="#">Container</a>	<a href="#">ExpressionForWrite</a>	<a href="#">NumberOfSteps</a>	<a href="#">Visible</a>
<a href="#">CurrentStep</a>	<a href="#">Height</a>	<a href="#">Object</a>	<a href="#">WhatsThisHelpID</a>
<a href="#">DataChanged</a>	<a href="#">HelpContextID</a>	<a href="#">Parent</a>	<a href="#">Width</a>
<a href="#">DataField</a>	<a href="#">Index</a>	<a href="#">StartValue</a>	
<a href="#">DataSource</a>	<a href="#">Left</a>	<a href="#">Symbol</a>	

## AutoAnimate Property

**Description** Determines if the RAnimator will cycle through its steps automatically at the speed designated by the AutoAnimateSpeed property.

**Custom** [General Tab](#) AutoAnimate check box

**Visual Basic** [Form.]RAnimator1.AutoAnimate[ = setting ]

**Remarks** Available options for the AutoAnimate property are:

Setting	Description
True	RAnimator control will continuously cycle through steps automatically.
False	RAnimator control will not cycle through steps automatically.

**Data Type** Boolean

## AutoAnimateSpeed Property

<b>Description</b>	Determines the speed that the RSAnimator control will cycle through each step in the animation when the AutoAnimate property is true. The speed is designated in milliseconds
<b>Custom</b>	<u><a href="#">General Tab</a></u> Auto Animate Speed check box
<b>Visual Basic</b>	[Form.]RSAnimator1.AutoAnimateSpeed[ =setting]
<b>Remarks</b>	The range for the AutoAnimateSpeed property is from 0 to 2,147,483,647.
<b>Data Type</b>	Long

**AutoValue Property**

**Description** When this box is marked the difference between the start and end value will be divided by the number of steps in the animation to produce a step value.

**Custom** [General Tab](#)

**Visual Basic** [Form.]RSAnimator1.AutoValue[ = setting ]

**Remarks** Available options for the AutoAnimate property are:

Setting	Description
True	RSAnimator control will continuously cycle through steps automatically.
False	RSAnimator control will not cycle through steps automatically.

**Data Type** Boolean

## Container Property

<b>Description</b>	Runtime property that returns or sets the container of an object.
<b>Custom</b>	Not available
<b>Visual Basic</b>	[Form.]RSAnimator1.Container [= container]
<b>Remarks</b>	A container is an object that can contain for other controls in it.
<b>Data Type</b>	Object

## CurrentStep Property

**Description** Returns or sets the current step in the animation

**Custom** [General Tab](#)

**Visual Basic** [Form.]RSAnimator1.CurrentStep[ = setting ]

**Remarks**

**Data Type** Integer

## DataChanged Property

<b>Description</b>	Runtime property that returns/sets a value indicating that data in a control has changed by some process other than by retrieving data from the current record.
<b>Custom</b>	Not available
<b>Visual Basic</b>	[Form.]RSAnimator1.DataChanged[ = setting ]
<b>Remarks</b>	The Visual Basic Data control will record changes made to a bound database as you move through its records. In order to avoid recording changes made to the database's records, set the DataChanged property to False in the Data control's Validate event. When the Data control is moved to the next record, the Validate event is fired and if DataChanged is True, changes made to the database are recorded.
<b>Data Type</b>	Boolean

## DataField Property

<b>Description</b>	Binds the RSAnimator to a particular field in a database. Used for reading from and writing to a database. The DataSource property must be set prior to this property to enable browsing.
<b>Custom</b>	Not available
<b>Visual Basic</b>	RSAnimator1. <b>DataField</b> [= <i>setting</i> ]
<b>Remarks</b>	The DataField, DataSource, and DataUpdate properties work together with the Visual Basic Data control to bind the RSTools control to a database.
<b>Data Type</b>	String



## DataSource Property

<b>Description</b>	Binds the RSAnimator to the particular Visual Basic data control which is bound directly to a database. The available data controls appear in a drop-down list next to the property name in the properties window
<b>Custom</b>	Not available
<b>Visual Basic</b>	RSAnimator1. <b>DataSource</b> [= <i>setting</i> ]
<b>Remarks</b>	DataSource is read/write at design time; not available at run time. The DataField, DataSource, and DataUpdate properties work together with the Visual Basic Data control to bind the RSTool control to a database.
<b>Data Type</b>	String

**DragIcon Property**

<b>Description</b>	Returns/sets the icon that will be displayed in a drag and drop operation.
<b>Custom</b>	Not available
<b>Visual Basic</b>	[Form.]RSAnimator1.DragIcon[ = setting ]
<b>Remarks</b>	
<b>Data Type</b>	Integer

## DragMode Property

**Description** Returns/sets a value that determines if manual or automatic drag mode is used.

**Custom** Not available

**Visual Basic** [Form.]RSAnimator1.DragMode[ = setting]

**Remarks** Available options for the DragMode property are:

Setting	Description
0	Manual drag mode
1	Automatic

**Data Type** Integer

## Enabled Property

**Description** Returns\sets a value (true or false) that determines if the RSAnimator can respond to user-generated events.

**Custom** Not available

**Visual Basic** [Form.]RSAnimator1.Enabled[ = setting ]

**Remarks** Available options for the Enabled property are:

Setting	Description
True	RSAnimator can respond to user-generated events.
False	RSAnimator can not respond to user-generated events.

**Data Type** Boolean

## EndValue Property

**Description**      Sets the maximum for the value range of the RSAnimator.

**Custom**            [General Tab](#)

**Visual Basic**      [Form.]RSAnimator1.EndValue[ = setting ]

**Remarks**

**Data Type**        Double

## ExpressionForRead Property

<b>Description</b>	Determines the mathematical expression that will be performed on the link item value when the RSAnimator reads that value.
<b>Custom</b>	<a href="#">LinkInfo Tab</a> . When the "Expression" button is pressed another form appears with two input boxes. Input the appropriate mathematical expression into the "Read Expression" text box. The expression must be in the form item[mathematical expression]. For example, "item+5" would add 5 to the LinkItem value.
<b>Visual Basic</b>	<i>RSAnimator1.ExpressionForRead[=<i>setting</i> ]</i>
<b>Remarks</b>	<p>This property allows you to perform a mathematical function on the link item as it is read by the control. The expression must be in the form item[mathematical expression]. For example, "item+5" would add 5 to the LinkItem value.</p> <p><i>Optional Calculation/Math module required, RSCALC32.DLL. Available with RSJunctionBox.</i></p>
<b>Data Type</b>	String

## ExpressionForWrite Property

<b>Description</b>	Determines the mathematical expression that will be performed on the link item when the RSAnimator writes that value.
<b>Custom</b>	<a href="#">LinkInfo Tab</a> . When the "Expression" button is pressed another form appears with two input boxes. Input the appropriate mathematical expression into the "Write Expression" text box. The expression must be in the form item[mathematical expression]. For example, "item+5" would add 5 to the LinkItem value.
<b>Visual Basic</b>	<i>RSAnimator1.ExpressionForWrite[=setting %]</i>
<b>Remarks</b>	<p>This property allows you to perform a mathematical function on the link item as a write is performed by the control. The expression must be in the form item[mathematical expression]. For example, "item+5" would add 5 to the LinkItem value.</p> <p><i>Optional Calculation/Math module required, RSCALC32.DLL. Available with RSJunctionBox.</i></p>
<b>Data Type</b>	String

## Height Property

**Description** Returns\sets the height of the RSAnimator.

**Custom** Not available.

**Visual Basic** [Form.]RSAnimator1.Height[ = setting ]

**Remarks**

**Data Type** Integer



## HelpContextID Property

**Description** Specifies the default Help file context ID for the RSAnimator.

**Custom** Not available

**Visual Basic** [Form.]RSAnimator1.HelpContextID[ = setting ]

**Remarks**

**Data Type** Integer

## Index Property

**Description** Returns\sets the number identifying an RAnimator in an RAnimator array.

**Custom** Not available

**Visual Basic** [Form.]RAnimator1.Index[ = setting ]

**Remarks**

**Data Type** Integer

## Left Property

<b>Description</b>	Returns\sets the distance between the internal left edge of the and the left edge of its container.
<b>Custom</b>	Not available
<b>Visual Basic</b>	[Form.]RSAnimator1.Left[ = setting ]
<b>Remarks</b>	The container could be the form or anther control.
<b>Data Type</b>	Integer

## LinkItem Property

<b>Description</b>	Returns\sets the item portion of the data link string to which the RSAnimator is linked.
<b>Custom</b>	<a href="#">LinkInfo tab</a>
<b>Visual Basic</b>	[Form.]RSAnimator1.LinkItem[ = setting ]
<b>Remarks</b>	Depending upon which type of DDE link is being established, the LinkItem can have many different formats, for example: "N7:1" is an integer address from a PLC datatable; "T4:0.ACC,L10" is a DDE block array, with a length of 10 items; "B3/0" is a binary address from a PLC datatable; "T4:11.ACC" is a timer address from a PLC datatable; and "r1c1" or "R2C2" are row and column addresses from a Microsoft Excel table.
<b>Data Type</b>	String

## LinkMode Property

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

**Custom** [LinkInfo Tab](#)

**Visual Basic** [Form.]RSAnimator1.LinkMode[ = setting ]

**Remarks** Available options for the LinkMode property are:

Setting	Description
0	None
1	Automatic
2	Manual
3	Notify

**Data Type** Integer

## LinkServer Property

<b>Description</b>	Returns\Sets the application or server name that the RSAnimator is linked to.
<b>Custom</b>	<a href="#">LinkInfo Tab</a>
<b>Visual Basic</b>	[Form.]RSAnimator1.LinkServer[ = setting ]
<b>Remarks</b>	Depending upon which type of DDE link is being established, the LinkServer can have different formats, for example: "ICOMWDRV" is the DDE server name for WINtelligent Linx and "EXCEL" is the DDE server name for Microsoft Excel.
<b>Data Type</b>	String

## LinkTopic Property

<b>Description</b>	Returns\Sets the topic portion of a data link string to which the RSAnimator is linked.
<b>Custom</b>	<a href="#">LinkInfo Tab</a>
<b>Visual Basic</b>	[Form.]RSAnimator1.LinkTopic[ = setting ]
<b>Remarks</b>	Depending upon which type of DDE link is being established, the LinkTopic can have different. Formats, for example: "testsol" would be a DDE topic created for the WIntelligent Linx driver and "sheet1.xls" would be a DDE topic name for Microsoft Excel.
<b>Data Type</b>	String

**Name Property**

**Description** Returns\sets the name used in code to identify the RSAnimator.

**Custom** Not available

**Visual Basic** [Form.]RSAnimator1.[ = setting ]

**Remarks**

**Data Type** String



## NumberOfSteps Property

**Description** Returns\sets the total number of steps in the animation.

**Custom** [General Tab](#)

**Visual Basic** [Form.]RSAnimator1.NumberOfSteps[ = setting ]

**Remarks** Maximum of 10,000 total steps.

**Data Type** Integer

## Object Property

<b>Description</b>	Returns an object that is contained inside the RSAnimator control.
<b>Custom</b>	Not available
<b>Visual Basic</b>	[Form.]RSAnimator1.Object[ = setting ]
<b>Remarks</b>	Runtime only
<b>Data Type</b>	Object

## Parent Property

**Description** Returns the form on which the RSAnimator is located.

**Custom** Not available

**Visual Basic** [Form.]RSAnimator1.Parent[ = setting ]

**Remarks** Runtime only

**Data Type** Form

## StartValue Property

**Description** Returns\Sets the starting value for the RSAnimator.

**Custom** [General Tab](#)

**Visual Basic** [Form.]RSAnimator1.StartValue[ = setting ]

**Remarks**

**Data Type** Double

## Symbol Property

<b>Description</b>	Serves as an alias name for the server, topic and item to be used in a DDE link.
<b>Custom</b>	<a href="#">LinkInfo Tab</a>
<b>Visual Basic</b>	[Form.]RSAnimator1.Symbol[ = setting ]
<b>Remarks</b>	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.
<b>Data Type</b>	String

## **TabIndex Property**

**Description** Returns\Sets the tab order of the RSAnimator within its parent form.

**Custom** Not available

**Visual Basic** [Form.]RSAnimator1.TabIndex[ = setting ]

**Remarks**

**Data Type** Integer

## TabStop Property

**Description** Returns\Sets a value (true or false) indicating if a user can use the TAB key to give the focus to the RAnimator.

**Custom** Not available

**Visual Basic** [Form.]RAnimator1.TabStop[ = setting ]

**Remarks** Available options for the Enabled property are:

Setting	Description
True	User can use the TAB key to give the focus to the RAnimator.
False	User can not use the TAB key to give the focus to the RAnimator.

**Data Type** Boolean

## Tag Property

<b>Description</b>	Stores any data needed for your program.
<b>Custom</b>	Not available
<b>Visual Basic</b>	[Form.]RSAnimator1.Tag[ = setting ]
<b>Remarks</b>	This is a user defined property that does not affect other VB properties
<b>Data Type</b>	String



## Top Property

<b>Description</b>	Returns\Sets the distance between the internal top edge of the RSAnimator and the top edge of its container.
<b>Custom</b>	Not available
<b>Visual Basic</b>	[Form.]RSAnimator1.Top[ = setting ]
<b>Remarks</b>	The container could be a form or another control.
<b>Data Type</b>	Single

## Value Property

**Description** Returns\Sets the value for the RSAnimator.

**Custom** Not available

**Visual Basic** [Form.]RSAnimator1.Value[ = setting ]

**Remarks**

**Data Type** Double

**Visible Property**

**Description**      Determines if the RSAnimator is visible at runtime

**Custom**            Not available

**Visual Basic**      [Form.]RSAnimator1.Visible[ = setting ]

**Remarks**        Available options for the Visible property are:

<b>Setting</b>	<b>Description</b>
True	RSAnimator will be visible at runtime.
False	RSAnimator will not be visible at runtime.

**Data Type**        Boolean

## WhatsThisHelpID Property

<b>Description</b>	Returns\Sets the associated WhatsThisHelp context number for the RAnimator.
<b>Custom</b>	Not available
<b>Visual Basic</b>	[Form.]RAnimator1.WatsThisHelpID[ = setting ]
<b>Remarks</b>	These context numbers are associated with Windows help files.
<b>Data Type</b>	Long

## Width Property

**Description** Returns\Sets the width of the RSAnimator.

**Custom** Not available

**Visual Basic** [Form.]RSAnimator1.Width[ = setting ]

**Remarks**

**Data Type** Single

## Event List for RSAnimator

<a href="#"><u>Change</u></a>	<a href="#"><u>KeyUp</u></a>	<a href="#"><u>LinkWriteCompleted</u></a>
<a href="#"><u>Click</u></a>	<a href="#"><u>LinkError</u></a>	<a href="#"><u>LinkWriteFailed</u></a>
<a href="#"><u>DblClick</u></a>	<a href="#"><u>LinkItemNotSupported</u></a>	<a href="#"><u>LostFocus</u></a>
<a href="#"><u>DragDrop</u></a>	<a href="#"><u>LinkItemSupported</u></a>	<a href="#"><u>MouseDown</u></a>
<a href="#"><u>DragOver</u></a>	<a href="#"><u>LinkNotify</u></a>	<a href="#"><u>MouseMove</u></a>
<a href="#"><u>GotFocus</u></a>	<a href="#"><u>LinkOutOfMemory</u></a>	<a href="#"><u>MouseUp</u></a>
<a href="#"><u>KeyDown</u></a>	<a href="#"><u>LinkServerDisconnected</u></a>	<a href="#"><u>PokeCompleted</u></a>
<a href="#"><u>KeyPress</u></a>	<a href="#"><u>LinkUnableToConnectToServer</u></a>	<a href="#"><u>RequestCompleted</u></a>

## Change Event

<b>Description</b>	Occurs when the RSGauge control's value has changed. The value can be changed through a link to the gauge through the LinkItem, through a bound field or through code.
<b>Visual Basic</b>	<b>Private Sub RSGauge1_Change</b> (Index As Integer, ByVal Value As Double)
<b>Remarks</b>	Arguments passed in this event are:  <i>Index</i> : The index value for the RSGauge control when it is part of an array of controls. (Shown only if true)  <i>Value</i> : The current value of the RSGauge.

## Click Event

<b>Description</b>	Occurs when the user presses and then releases a mouse button over the RAnimator.
<b>Visual Basic</b>	<b>Private Sub RAnimator1_Click</b> ( <i>Index</i> As Integer)
<b>Remarks</b>	Arguments passed in this event are:  <i>Index</i> : The index value for the RAnimator control when it is part of an array of controls. (Shown only if true)



## DblClick Event

**Description:** Occurs when the user presses and releases a mouse button and then presses and releases it again over the RAnimator.

**Visual Basic** **Private Sub RAnimator1\_DblClick**(*Index* As Integer

**Remarks** Arguments passed in this event are:

*Index*: The index value for the RAnimator control when it is part of an array of controls.  
(Shown only if true)

## DragDrop Event

<b>Description:</b>	Occurs when a drag-and-drop operation is completed as a result of dragging a control over a form or control and releasing the mouse button or using the Drag method with its action argument set to 2 (Drop). Use a DragDrop event procedure to control what happens after a drag operation is completed.
<b>Visual Basic</b>	<b>Private Sub RAnimator1_DragDrop</b> ( <i>Index</i> As Integer, <i>Source</i> As Control, <i>x</i> As Single, <i>y</i> As Single)
<b>Remarks</b>	<p>Arguments passed in this event are:</p> <p><i>Index</i>: The index value for the RAnimator control when it is part of an array of controls. (Shown only if true)</p> <p><i>Source</i>: This is the control being dragged. Properties and methods can be included with this argument, for example, RAnimator.Visible = false..</p> <p><i>x, y</i>: These numbers specify the current horizontal (x) and vertical (y) location of the mouse pointer within the form or control that the control is being dragged to. The x and y values are expressed in terms of the coordinate system of the form or control that the control is being dragged to.</p>

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

## DragOver Event

<b>Description:</b>	Occurs when a drag-and-drop operation is in progress. You can use this event to monitor the mouse pointer as it enters, leaves, or rests directly over a valid target. The mouse pointer position determines the destination object that receives this event
<b>Visual Basic</b>	<b>Private Sub RAnimator1_DragOver</b> ( <i>Index</i> As Integer, <i>Source</i> As Control, <i>x</i> As Single, <i>y</i> As Single, <i>State</i> As Integer)
<b>Remarks</b>	<p>Arguments passed in this event are:</p> <p><i>Index</i>: The index value for the RAnimator control when it is part of an array of controls. (Shown only if true)</p> <p><i>Source</i>: This is the control being dragged. Properties and methods can be included with this argument, for example, RAnimator.Visible = false..</p> <p><i>x, y</i>: These numbers specify the current horizontal (x) and vertical (y) location of the mouse pointer within the form or control that the control is being dragged to. The x and y values are expressed in terms of the coordinate system of the form or control that the control is being dragged to.</p> <p><i>State</i>: An integer that represents the transition state of the RAnimator being dragged with respect to the destination form or control:</p> <p>0 = Enter (RAnimator is being dragged within the range of the destination).</p> <p>1 = Leave (RAnimator is being dragged out of the range of the destination).</p> <p>2 = Over (RAnimator has moved from one position in the destination to another).</p>

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

## GotFocus Event

**Description:** Occurs when the RAnimator receives the focus, either by user action, such as tabbing to or clicking with the mouse pointer on the RAnimator, or by changing the focus in code using the SetFocus method.

**Visual Basic** **Private Sub RAnimator1\_GotFocus(*Index* As Integer)**

**Remarks** Arguments passed in this event are:

*Index*: The index value for the RAnimator control when it is part of an array of controls.  
(Shown only if true)

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

## KeyDown Event, KeyDown Event

**Description:** Occur when a keyboard key is pressed (KeyDown) or released (KeyUp) when the RAnimator has the focus

**Visual Basic** **Private Sub RAnimator1\_KeyDown**(*Index* As Integer, *KeyCode* As Integer, *Shift* As Integer)

**Remarks** Arguments passed in this event are:

*Index*: The index value for the RAnimator control when it is part of an array of controls.  
(Shown only if true)

*KeyCode*: A key code which is used to represent a keyboard key, such as vbKeyF1 (the F1 key). To assign key codes, use the constants in the Visual Basic (VB) object library in the Object Browser.

*Shift*: An integer that represents the state of the SHIFT, CTRL, and ALT keys at the time of the event occurs. The shift argument is a bit field with the least-significant bits corresponding to the SHIFT key (bit 0), the CTRL key (bit 1), and the ALT key (bit 2). These bits correspond to the values 1, 2, and 4, respectively. Some, all, or none of the bits can be set, indicating that some, all, or none of the keys are pressed. For example, if both CTRL and ALT are pressed, the value of shift is 6.

For additional information, refer to the description of the KeyDown, KeyUp events in the Microsoft Visual Basic Language Reference Manual.

## KeyPress Event

**Description:** Occurs when a ANSI keyboard key is pressed and released while the RAnimator has the focus.

**Visual Basic** **Private Sub RAnimator1\_KeyPress**(*Index* As Integer, *KeyAscii* As Integer)

**Remarks** Arguments passed in this event are:

*Index*: The index value for the RAnimator control when it is part of an array of controls.  
(Shown only if true)

*KeyAscii*: An integer that represents the standard numeric ANSI keycode for the key that was pressed..

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

## LinkError Event

**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 there isn't any Visual Basic code being executed. The error number is passed as an argument.

**Visual Basic** **Private Sub RAnimator1\_LinkError**(*Index* as Integer, ByVal *iRet* As Integer, ByVal *ErrorString* As String)

**Remarks** Arguments passed in this event are:

*Index*: The index value for the RAnimator control when it is part of an array of controls.  
(Shown only if true)

*iRet*: Error Number

*ErrorString*:

## LinkItemNotSupported Event

<b>Description</b>	Occurs when the RAnimator's LinkItem has an incorrect format.
<b>Visual Basic</b>	<b>Private Sub RAnimator1_LinkItemNotSupported</b> ( <i>Index</i> as Integer)
<b>Remarks</b>	Arguments passed in this event are:  <i>Index</i> : The index value for the RAnimator control when it is part of an array of controls. (Shown only if true)



## LinkItemSupported Event

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

**Visual Basic** **Private Sub RAnimator1\_LinkItemSupported**(*Index* as Integer

**Remarks** Arguments passed in this event are:

*Index*: The index value for the RAnimator control when it is part of an array of controls.  
(Shown only if true)

## LinkNotify Event

<b>Description</b>	Occurs when the source has changed the data defined by the DDE link if the LinkMode property of the RAnimator is set to 3 (Notify).
<b>Visual Basic</b>	<b>Private Sub RAnimator1_LinkNotify</b> ( <i>Index</i> as Integer)
<b>Remarks</b>	Arguments passed in this event are:  <i>Index</i> : The index value for the RAnimator control when it is part of an array of controls. (Shown only if true)

## LinkOutOfMemory Event

<b>Description</b>	Occurs when the client (RSAnimator) has exhausted its memory resources.
<b>Visual Basic</b>	<b>Private Sub RSAnimator1_LinkOutOfMemory</b> ( <i>Index</i> as Integer
<b>Remarks</b>	Arguments passed in this event are:  <i>Index</i> : The index value for the RSAnimator control when it is part of an array of controls. (Shown only if true)

## LinkServerDisconnected Event

**Description**      Occurs when the RSAnimator is connected to a server and that server becomes unavailable..

**Visual Basic**      **Private Sub RSAnimator1\_LinkServerDisconnected**(*Index* as Integer)

**Remarks**          Arguments passed in this event are:

*Index*: The index value for the RSAnimator control when it is part of an array of controls.  
(Shown only if true)

## LinkUnableToConnectToServer Event

<b>Description</b>	Occurs when the RAnimator attempts to connect to a server that is unavailable.
<b>Visual Basic</b>	<b>Private Sub RAnimator1_LinkUnableToConnectToServer</b> ( <i>Index</i> as Integer)
<b>Remarks</b>	Arguments passed in this event are:  <i>Index</i> : The index value for the RAnimator control when it is part of an array of controls. (Shown only if true)

## LinkWriteCompleted Event

<b>Description</b>	Occurs when the RSAimator has completed writing to the DDE LinkItem.
<b>Visual Basic</b>	<b>Private Sub RSAimator1_LinkWriteCompleted</b> ( <i>Index</i> as Integer)
<b>Remarks</b>	Arguments passed in this event are:  <i>Index</i> : The index value for the RSAimator control when it is part of an array of controls. (Shown only if true)

## LinkWriteFailed Event

<b>Description</b>	Occurs when the RSAnimator has failed at trying to write to the DDE LinkItem.
<b>Visual Basic</b>	<b>Private Sub RSAnimator1_LinkWriteFailed</b> ( <i>Index</i> as Integer)
<b>Remarks</b>	Arguments passed in this event are:  <i>Index</i> : The index value for the RSAnimator control when it is part of an array of controls. (Shown only if true)

## LostFocus Event

**Description** Occurs when the RSAimator 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 RSAimator1\_LostFocus**(*Index* as Integer)

**Remarks** Arguments passed in this event are:

*Index*: The index value for the RSAimator control when it is part of an array of controls.  
(Shown only if true)

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



## MouseDown Event, MouseUp Event

<b>Description</b>	Occurs when a mouse button is pressed (MouseDown) or released (MouseUp) while the mouse pointer is positioned over the RAnimator.
<b>Visual Basic</b>	<b>Private Sub RAnimator1_MouseDown</b> ( <i>Index</i> as Integer, <i>Button</i> As Integer, <i>Shift</i> As Integer, <i>x</i> As Single, <i>y</i> As Single)
<b>Remarks</b>	<p>Arguments passed in this event are:</p> <p><i>Index</i>: The index value for the RAnimator control when it is part of an array of controls. (Shown only if true)</p> <p><i>Button</i>: Returns an integer that represents the mouse button that was pressed (MouseDown) or released (MouseUp). The button argument is a bit field with bits corresponding to the left button (bit 0), right button (bit 1), and middle button (bit 2). These bits correspond to the values 1, 2, and 4, respectively. Only one of the bits is set, indicating the button that caused the event.</p> <p><i>Shift</i>: Returns an integer that represents the state of the SHIFT, CTRL, and ALT keys when the mouse button specified in the button argument is pressed or released. A bit is set if the key is down. The shift argument is a bit field with the least-significant bits corresponding to the SHIFT key (bit 0), the CTRL key (bit 1), and the ALT key (bit 2). These bits correspond to the values 1, 2, and 4, respectively. The shift argument indicates the state of these keys. Some, all, or none of the bits can be set, indicating that some, all, or none of the keys are pressed. For example, if both CTRL and ALT were pressed, the value of shift would be 6.:</p> <p><i>x, y</i>: These numbers specify the current horizontal (x) and vertical (y) location of the mouse pointer when the mouse pointer is pressed.</p>

For additional information, refer to the description of the MouseDown or MouseUp events in the Microsoft Visual Basic Language Reference Manual.

## MouseMove Event

<b>Description</b>	Occurs when the mouse pointer is moved over the RAnimator.
<b>Visual Basic</b>	<b>Private Sub RAnimator1_MouseMove</b> ( <i>Index</i> as Integer, <i>Button</i> As Integer, <i>Shift</i> As Integer, <i>x</i> As Single, <i>y</i> As Single)
<b>Remarks</b>	<p>Arguments passed in this event are:</p> <p><i>Index</i>: The index value for the RAnimator control when it is part of an array of controls. (Shown only if true)</p> <p><i>Button</i>: Returns an integer that represents the mouse button that the current state of the mouse buttons. The button argument is a bit field with bits corresponding to the left button (bit 0), right button (bit 1), and middle button (bit 2). These bits correspond to the values 1, 2, and 4, respectively. Only one of the bits is set, indicating that a button pressed.</p> <p><i>Shift</i>: Returns an integer that represents the state of the SHIFT, CTRL, and ALT keys when the mouse button specified in the button argument is pressed or released. A bit is set if the key is down. The shift argument is a bit field with the least-significant bits corresponding to the SHIFT key (bit 0), the CTRL key (bit 1), and the ALT key (bit 2). These bits correspond to the values 1, 2, and 4, respectively. The shift argument indicates the state of these keys. Some, all, or none of the bits can be set, indicating that some, all, or none of the keys are pressed. For example, if both CTRL and ALT were pressed, the value of shift would be 6.:</p> <p><i>x, y</i>: These numbers specify the current horizontal (x) and vertical (y) location of the mouse pointer when the mouse pointer is pressed.</p>

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

## PokeCompleted Event

<b>Description</b>	Occurs when the RAnimator completes poking (writing) to the DDE LinkItem.
<b>Visual Basic</b>	<b>Private Sub RAnimator1_StartMove</b> ( <i>Index</i> as Integer, ByVal <i>iRet</i> As Integer)
<b>Remarks</b>	Arguments passed in this event are:  <i>Index</i> : The index value for the RAnimator control when it is part of an array of controls. (Shown only if true)  <i>iRet</i> : Error numbers.

## RequestCompleted Event

<b>Description</b>	Occurs when the RSAimator completes requesting (reading) from the DDE LinkItem.
<b>Visual Basic</b>	<b>Private Sub RSAimator1_RequestCompleted</b> ( <i>Index</i> as Integer, ByVal <i>iRet</i> As Integer)
<b>Remarks</b>	Arguments passed in this event are:  <i>Index</i> : The index value for the RSAimator control when it is part of an array of controls. (Shown only if true)  <i>iRet</i> : Error numbers.

Note : This event is useful for preventing the sending of additional data to a server while it is still processing previous data.("Outrunning the server")

## Method List for RSAnimator

[Drag](#)                      [SetFocus](#)                      [ZOrder](#)  
[GetPropValue](#)                      [SetPropValue](#)  
[Move](#)                      [ShowWhatsThis](#)

### Drag Method

**Description:** Begins, ends or cancels a drag operation of the RSAnimator.

**Visual Basic:** RSAnimator1.Drag *action*

**Remarks** Available options for *action*

Setting	Description
0	vbCancel - Cancels drag operation
1	vbBeginDrag - Begins dragging RSAnimator.
2	vbEndDrag - Ends dragging and drop of RSAnimator.

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

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

## GetPropValue Method

**Description:** Returns a property's value of a control that is configured to be manipulated by the RSAnimator during an animation.

**Visual Basic:** RSAnimator1.GetPropValue(*nStep* As Integer, *Control* As String, *Property* As String) As Variant

**Remarks** Arguments passed in this method are:

*nStep:* The step value in the animation that you want the property value for.

*Control:* The name of the control whose property value you require.

*Property:* The name of the property whose value you require.

## Move Method

**Description:** Moves an RAnimator control.

**Visual Basic:** RAnimator1.Move left; top; width; height(iIndex As Integer) As Double

**Remarks** Only the left argument is required. However, to specify any other arguments, you must specify all arguments that appear in the syntax.

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

## **SetFocus Method**

**Description:** Moves the focus to the RAnimator.

**Visual Basic:** RAnimator1.SetFocus

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



## SetPropValue Method

<b>Description:</b>	Sets a property value of a control that is configured to be manipulated by the RSAnimator during an animation.
<b>Visual Basic:</b>	RSAnimator1.GetPropValue( <i>nStep</i> As Integer, <i>Control</i> As String, <i>Property</i> As String, <i>Value</i> As String) As Variant
<b>Remarks</b>	<p>Arguments passed in this method are:</p> <p><i>nStep</i>: The step value in the animation that you want to set a property value for.</p> <p><i>Control</i>: The name of the control whose property value you want to set.</p> <p><i>Property</i>: The name of the property whose value you want to set.</p> <p><i>Value</i>: The value that you want to set the property to.</p>

## ShowWhatsThis

- Description:** Displays a selected topic in a Help file using the What's This popup provided by Windows 95 Help.
- Visual Basic:** RSAnimator1.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

**Description** Places the RSAnimator at the front or back of the z-order within its graphical level.

**Visual Basic** RSAnimator1.ZOrder position

**Remarks** Available options for position are;

<b>Setting</b>	<b>Description</b>
0	RSAnimator positioned at the front of the Zorder
1	RSAnimator positioned at the back of the ZOrder.

## INI File Used for RSAnimator

RSAnimator saves some of the information such as the default files names etc. in the RSTOOLS.INI file located in the \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 file for the RSAnimator control. The user templates are stored in this file.

```
[cfg]
```

```
cfg=c:\windows\mytempl.s.rwc
```

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

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

```
[sym]
```

```
symbolname=server|topic|item
```

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

```
[sym]
```

```
subtotal= Excel|[Book1]Sheet1|R6C2
```

```
total= Excel|[Book1]Sheet1|R8C2
```

