

## Macromedia Control Xtra Help

To learn more about the Control Xtra for ActiveX, choose the product you are using:

{button ,ji(``,`activex\_contents\_director`)} [Director](#)

{button ,ji(``,`activex\_contents\_authorware`)} [Authorware](#)

Copyright © 1997 Macromedia, Inc. All rights reserved. The information in this help system may not be copied, photocopied, reproduced, translated, or converted to any printed, electronic or machine-readable form in whole or in part without prior written approval of Macromedia, Inc.

## Macromedia Control Xtra Help (Director)

{button ,ji('`,`activex\_the\_big\_picture\_director')}} [Overview](#)

{button ,ji('`,`activex\_step\_by\_step\_procedures\_director')}} [How to](#)

{button ,ji('`,`activex\_activex\_xtra\_scripting\_functions\_director')}} [Scripting functions](#)

## Macromedia Control Xtra Help (Authorware)

{button ,ji(`,`activex\_the\_big\_picture\_authorware')} [Overview](#)

{button ,ji(`,`activex\_step\_by\_step\_procedures\_authorware')} [How to](#)

{button ,ji(`,`activex\_activex\_xtra\_scripting\_functions\_authorware')} [Scripting functions](#)

{button ,ji(``,`activex\_contents\_director`)} **CONTENTS**

## **Overview**

The Control Xtra lets you use most ActiveX controls within a Director movie. Before you get started, you'll need to know more about the Control Xtra's features:

{button ,ji(``,`activex\_purpose\_director`)} **What does the Control Xtra for ActiveX do?**

{button ,ji(``,`activex\_components\_director`)} **Control Xtra components**

{button ,ji(``,`activex\_control\_properties\_director`)} **ActiveX control properties**

{button ,ji(``,`activex\_control\_methods\_director`)} **ActiveX control methods**

{button ,ji(``,`activex\_control\_events\_director`)} **ActiveX control events**

{button ,ji(``,`activex\_contents\_authorware')}} **CONTENTS**

## **Overview**

The Control Xtra lets you use most ActiveX controls within an Authorware piece. Before you get started, you'll need to know more about the Control Xtra for ActiveX's features:

{button ,ji(``,`activex\_purpose\_authorware')}} **What does the Control Xtra for ActiveX do?**

{button ,ji(``,`activex\_components\_authorware')}} **Control Xtra for ActiveX components**

{button ,ji(``,`activex\_control\_properties\_authorware')}} **ActiveX control properties**

{button ,ji(``,`activex\_control\_methods\_authorware')}} **ActiveX control methods**

{button ,ji(``,`activex\_control\_events\_authorware')}} **ActiveX control events**

{button ,ji('',`activex\_contents\_director'')} **CONTENTS**

## **How to**

The following topics provide instructions for using the Control Xtra for ActiveX:

{button ,ji('',`activex\_inserting\_director'')} **Inserting an ActiveX control**

{button ,ji('',`activex\_editing\_properties\_director'')} **Editing property values**

{button ,ji('',`activex\_examining\_methods\_director'')} **Examining methods and events**

{button ,ji('',`activex\_downloading\_director'')} **Downloading ActiveX controls within a Director movie**

{button ,ji('',`activex\_contents\_authorware')}} **CONTENTS**

## **How to**

The following topics provide instructions for using the Control Xtra for ActiveX:

{button ,ji('',`activex\_inserting\_authorware')}} **Inserting an ActiveX Control**

{button ,ji('',`activex\_editing\_properties\_authorware')}} **Editing property values**

{button ,ji('',`activex\_examining\_methods\_authorware')}} **Examining methods and events**

{button ,ji('',`activex\_downloading\_authorware')}} **Downloading ActiveX controls within an Authorware piece**

{button ,ji('`,`activex\_contents\_director')}} **CONTENTS**    {button ,ji('`,`activex\_the\_big\_picture\_director')}} **OVERVIEW**

### **What does the Control Xtra for ActiveX do?**

Use the Control Xtra to employ ActiveX controls within a Director movie. The range of uses for the Control Xtra for ActiveX is as limitless as the variety of ActiveX controls available. For example, you can:

- Browse the Internet from within a multimedia production using the Microsoft Web Browser Control (installed with Microsoft Internet Explorer 3.0 or 4.0.)
- Create and access spreadsheets using the FarPoint Spreadsheet Control
- Explore virtual worlds using the InterVista VRML control
- Build and simulate complete windows applications within Director or Authorware using MicroHelp's extensive library of Window widget controls

The Control Xtra for ActiveX functions as a Sprite Xtra within Director and manages application resources for the hosted ActiveX control. This resource management includes property, event, and window management, along with the filing of properties and resources used by the ActiveX control within the Director movie itself.



{button ,ji('',`activex\_contents\_authorware' )} [CONTENTS](#)

{button ,ji('',`activex\_the\_big\_picture\_authorware' )} [OVERVIEW](#)

### **What does the Control Xtra for ActiveX do?**

Use the Control Xtra to employ ActiveX controls within an Authorware piece. The range of uses for the Control Xtra for ActiveX is as limitless as the variety of ActiveX controls available. For example, you can:

- Browse the Internet from within a multimedia production using the Microsoft Web Browser Control (installed with Microsoft Internet Explorer3.0 or 4.0.)
- Create and access spreadsheets using the FarPoint Spreadsheet Control
- Explore virtual worlds using the InterVista VRML control
- Build and simulate complete windows applications within Director or Authorware using MicroHelp's extensive library of Window widget controls

The Control Xtra for ActiveX functions as a Sprite Xtra within Authorware and manages application resources for the hosted ActiveX control. This resource management includes property, event, and window management, along with the filing of properties and resources used by the ActiveX control within the Authorware piece itself.

[{button ,ji\('`,`activex\\_contents\\_director'\)}``} CONTENTS](#)

[{button ,ji\('`,`activex\\_the\\_big\\_picture\\_director'\)}``} OVERVIEW](#)

## **Control Xtra for ActiveX components**

To ensure that the runtime component is as small as possible for redistribution, the Control Xtra for ActiveX consists of two MOA Xtras:

**ActiveX.X32** The runtime component

**ActXPriv.X32** The user interface component

When using Director to create an interactive title, both components should be installed in the Xtras directory local to the application.

After the interactive title has been packaged, you should redistribute only the runtime component, ActiveX.X32, with the title.

**Important:** Under no circumstances should the user interface component, ActXPriv.X32, be redistributed with a packaged Director title.

The files listed below are also provided with the Control Xtra for ActiveX. These are self-installing executables provided by Microsoft that will upgrade a Windows 95 or Windows NT 4.0 system to run ActiveX components. These files can be called by the installation program for a multimedia title that uses the Control Xtra for ActiveX.

**APRXDIST.EXE** ActiveX sub-installer for Win 95/NT 4.0.

**AXDIST.EXE** ActiveX sub-installer for Win 95/NT 4.0.

**WINDIST.EXE** ActiveX sub-installer for Win 95/NT 4.0.

[{button ,ji\('',`activex\\_contents\\_authorware' \)} CONTENTS](#)

[{button ,ji\('',`activex\\_the\\_big\\_picture\\_authorware' \)} OVERVIEW](#)

## **Control Xtra for ActiveX components**

To ensure that the runtime component is as small as possible for redistribution, the Control Xtra for ActiveX consists of two MOA Xtras:

**ActiveX.X32**      The runtime component

**ActXPriv.X32**    The user interface component

When using Authorware to create an interactive title, both components should be installed in the Xtras directory local to the application.

After the interactive title has been packaged, you should redistribute only the runtime component, ActiveX.X32, with the title.

**Important:** Under no circumstances should the user interface component, ActXPriv.X32, be redistributed with a packaged Authorware title.

The files listed below are also provided with the Control Xtra for ActiveX. These are self-installing executables provided by Microsoft that will upgrade a Windows 95 or Windows NT 4.0 system to run ActiveX components. These files can be called by the installation program for a multimedia title that uses the Control Xtra for ActiveX.

**APRXDIST.EXE**    ActiveX sub-installer for Win 95/NT 4.0.

**AXDIST.EXE**      ActiveX sub-installer for Win 95/NT 4.0.

**WINDIST.EXE**     ActiveX sub-installer for Win 95/NT 4.0.

[{button ,ji\('`,`activex\\_contents\\_director'\)}``}](#) **CONTENTS**    [{button ,ji\('`,`activex\\_the\\_big\\_picture\\_director'\)}``}](#) **OVERVIEW**

## **ActiveX control properties**

An ActiveX control describes its information through properties, which are named characteristics or values such as color, text, font, and so on. Properties can include not only visual characteristics, but also behavioral ones. For example, a button can have a property that indicates whether the button is a push button (it can be pushed on or off).

An ActiveX control's properties define its state; some or all of these properties can be persistent. A control can change its own properties; the container holding the control can also change a property in response to which the control would change its state, user interface, and so on.

When an ActiveX control is inserted into a Director movie or an Authorware piece, you can view and edit the properties exposed by the control by selecting the **Properties tab** of the Control Properties dialog for the Control Xtra for ActiveX.

The Control Xtra for ActiveX lists each property exported by the ActiveX control along with the current value of the property. You can **edit a property value** by clicking the existing value with the mouse. For most properties such as numeric or string values, you can enter the new value directly using the keyboard.

All methods exported by an ActiveX control are exported automatically by the Control Xtra for ActiveX as properties of the corresponding Sprite Xtra.

The generic Lingo syntax for setting an ActiveX control property is:

```
set the PropertyName of sprite X to Value
```

The generic Lingo syntax for getting an ActiveX control property is:

```
put the PropertyName of sprite X into Value
```

For example, if the Microsoft Access Calendar control is inserted into a Director movie as the second sprite on the score, the following Lingo code sets the Year property of the Calendar control to a specific year:

```
set the Year of sprite 2 to 1995
```

To get the Year property from the same Calendar Control and place it into a Lingo variable named CalendarYear, the following Lingo code would be used:

```
put the Year of sprite 2 into CalendarYear
```

Some ActiveX control properties are read-only. In this case, an attempt to set a property for such a control causes an error within Director.

## ActiveX control properties

An ActiveX control describes its information through properties, which are named characteristics or values such as color, text, font, and so on. Properties can include not only visual characteristics, but also behavioral ones. For example, a button can have a property that indicates whether the button is a push button (it can be pushed on or off).

An ActiveX control's properties define its state; some or all of these properties can be persistent. A control can change its own properties; the container holding the control can also change a property in response to which the control would change its state, user interface, and so on.

When an ActiveX control is inserted into a Director movie or an Authorware piece, you can view and edit the properties exposed by the control by selecting the [Properties tab](#) of the Control Properties dialog box for the Control Xtra for ActiveX.

The Control Xtra for ActiveX lists each property exported by the ActiveX control along with the current value of the property. You can [edit a property value](#) by clicking the existing value with the mouse. For most properties such as numeric or string values, you can enter the new value directly using the keyboard.

For Authorware, all methods exported by an ActiveX control are exported automatically by the Control Xtra for ActiveX as properties of the corresponding Sprite Xtra.

The generic Authorware calc language syntax for setting an ActiveX control property is:

```
SetSpriteProperty (@ "Icon Title", #PropName, Value)
```

The generic Authorware calc language syntax for getting an ActiveX control property is:

```
Value =GetSpriteProperty ( @ "Icon Title", #PropName)
```

For example, if the Microsoft Access Calendar control is inserted as a Sprite Xtra into an Authorware piece with the title "Calendar", the following Authorware calc language statement would set the Year property of the Calendar control to specific year:

```
SetSpriteProperty (@ "Calendar", #Year, 1996 )
```

To get the Year property from the same Calendar Control and place it into an Authorware variable named CalendarYear, you can use the following calc language statement:

```
CalendarYear =GetSpriteProperty ( @ "Calendar", #Year )
```

Some ActiveX control properties are read-only. In this case, an attempt to set a read-only property is not allowed.

[{button ,ji\('`,`activex\\_contents\\_director'\)}`` CONTENTS](#)    [{button ,ji\('`,`activex\\_the\\_big\\_picture\\_director'\)}`` OVERVIEW](#)

## ActiveX control methods

An ActiveX control describes its functionality through methods. Methods are functions implemented in the control that can be called from Director to perform some action. For example, an edit or other text-oriented control supports methods to allow Director to retrieve or modify the current text, perhaps performing such operations as copy and paste with that control.

When an ActiveX control is inserted into a Director movie, you can [view the methods](#) exposed by the control by selecting the [Methods tab](#) of the Control Properties dialog box for the Control Xtra for ActiveX. The Control Xtra for ActiveX lists each method exported by the ActiveX control along with a description of the parameters for each method.

All methods exported by an ActiveX control are exported automatically by the Control Xtra for ActiveX as functions for the corresponding Sprite Xtra.

The generic Lingo syntax for calling an ActiveX control method is:

```
put MethodName (sprite N, param1, param2, ... ) into RetValue
```

For example, if the Microsoft Access Calendar control is inserted into a Director movie as the second sprite on the score, the following Lingo code would increment the year displayed within the Calendar control:

```
NextYear ( sprite 2 )
```

For the same Calendar control the following Lingo code would decrement the year displayed by the Calendar control:

```
PrevYear ( sprite 2 )
```

Parameters passed to the ActiveX control are converted automatically from their Director data types to equivalent ActiveX data types by the Control Xtra for ActiveX. The return value is converted automatically from an ActiveX data type to an equivalent Director data type.

{button ,ji('`,`activex\_contents\_authorware')}} **CONTENTS**    {button ,ji('`,`activex\_the\_big\_picture\_authorware')}} **OVERVIEW**

## **ActiveX control methods**

An ActiveX control describes its functionality through methods. Methods are functions implemented in the control that can be called from Authorware to perform some action. For example, an edit or other text-oriented control supports methods to allow Authorware to retrieve or modify the current text, perhaps performing such operations as copy and paste with that control.

When an ActiveX control is inserted into an Authorware piece, you can [view the methods](#) exposed by the control by selecting the **Methods tab** of the Control Properties dialog box for the Control Xtra for ActiveX. The Control Xtra for ActiveX lists each method exported by the ActiveX control along with a description of the parameters for each method.

All methods exported by an ActiveX control are exported automatically by the Control Xtra for ActiveX as calc functions for the corresponding Sprite Xtra.

The generic Authorware calc language syntax for calling an ActiveX control method is:

```
RetVal = CallSprite( @"Icon Name", #MethodName, param1,
    param2, ... )
```

For example, if the Microsoft Access Calendar control is inserted as a Sprite Xtra into an Authorware piece with the title "Calendar", the following Authorware calc function call on the Sprite Xtra would increment the year displayed within the Calendar control:

```
CallSprite (@ "Calendar", #NextYear )
```

For the same Calendar control the following Authorware calc function call on the Sprite Xtra would decrement the year displayed by the control:

```
CallSprite (@ "Calendar", #PrevYear )
```

Parameters passed to the ActiveX control are converted from their Authorware data types to an equivalent ActiveX data type by the Control Xtra for ActiveX. The return value is converted automatically from an ActiveX data type to an equivalent Authorware data type.

{button ,ji('`,`activex\_contents\_director')}} [CONTENTS](#)    {button ,ji('`,`activex\_the\_big\_picture\_director')}} [OVERVIEW](#)

## ActiveX control events

Each ActiveX control typically generates a variety of events. For example, a button ActiveX control might generate a click event when the button is pressed and a calendar ActiveX control might generate a dateChanged event when the date within the calendar is changed. The Control Xtra for ActiveX converts automatically any event generated by the ActiveX control to a Sprite Xtra event that Director can handle.

When an ActiveX control is inserted into a Director movie, you can [view the events](#) exposed by the control by selecting the [Events tab](#) of the Control Properties dialog box for the Control Xtra for ActiveX. The Control Xtra for ActiveX lists each method exported by the ActiveX control along with a description of the parameters for each method.

For Director to respond to an event generated by the ActiveX control, you must create a Lingo event handler to capture the event. These event handlers can be placed into movie, sprite, cast or frame scripts. However, normally the handler will be placed into the script for the sprite that represents the ActiveX control in the score.

For example, if the Microsoft Access Calendar control is inserted into a Director movie as a sprite in the score, the following Lingo code would capture the click event from the Calendar control:

```
on click
    -- Do something interesting here.
    beep 2
end
```



{button ,ji('`,`activex\_contents\_authorware')}} **CONTENTS**    {button ,ji('`,`activex\_the\_big\_picture\_authorware')}} **OVERVIEW**

## **ActiveX control events**

Each ActiveX control typically generates a variety of events. For example, a button ActiveX control might generate a click event when the button is pressed and a calendar ActiveX control might generate a dateChanged event when the date within the calendar is changed. The Control Xtra for ActiveX converts automatically any event generated by the ActiveX control to a Sprite Xtra event that Authorware can handle.

When an ActiveX control is inserted into an Authorware piece, you can [view the events](#) exposed by the control by selecting the [Events tab](#) of the Control Properties dialog box for the Control Xtra for ActiveX. The Control Xtra for ActiveX lists each method exported by the ActiveX control along with a description of the parameters for each method.

For Authorware, to respond to an event generated by the ActiveX control, you must create an event response type symbol within Authorware.

### **To set up an event response type symbol:**

- 1 Set up an interaction icon if you haven't already done so.**  
Refer to the Authorware documentation regarding how to set up an interaction icon.
- 2 Drag an icon to the right of the interaction icon.**  
The icon determines what happens when a user clicks the button you're setting up.  
If you haven't planned in detail what you want to happen, use a map icon, because they're the easiest to modify. The Response Type dialog box appears automatically as soon as you release the icon if it's the first one you've added to the interaction. If it's not the first, Authorware doesn't display the dialog box because it assumes that you want to use the same response type symbol you last chose. If the Response Type dialog box doesn't appear automatically, Control-double-click or Command-double-click the response type symbol to display it.
- 3 Select Event, then click OK. The response type symbol on the interaction flowline changes to the event symbol.**
- 4 Give the result icon a name.**

### **To select the event(s) you want the event type symbol to recognize as its target response:**

- 1 Double-click the event response type symbol.**  
The Event response properties dialog box appears. Make sure the event card is currently active. The names of all the Sprite Xtras that exist in the Authorware piece appear in the Sender list.
- 2 Double-click the Sprite Xtra icon to whose event(s) you want to respond.**  
An x appears to the left of the Sprite Xtra name to indicate that it has been selected. You can select multiple Sprite Xtras. All the events generated by the selected Sprite Xtras appear in the Event Name list.
- 3 In the Event Name list, double-click the event names(s) that you want to respond to.**  
An x appears to the left of the event name to indicate that it has been selected. You can select multiple event names. Instructions on the suggested usage of the event that is currently selected may appear in the Description field below the Event Name list.
- 4 Make your desired settings to the response card, then click OK.**

{button ,ji('`,`activex\_contents\_director')}} [CONTENTS](#)    {button ,ji('`,`activex\_step\_by\_step\_procedures\_director')}} [HOW TO](#)

## Inserting an ActiveX control

With the Control Xtra for ActiveX (a type of Sprite Xtra), you can place ActiveX controls (formerly known as OLE/OCX controls) within a Director movie. Once inserted, the ActiveX control functions as a Sprite Xtra.

### To insert an ActiveX control onto the stage:

- 1 Make sure that the ActiveX controls you want to use within Director are installed on your system.**  
Most controls have their own installation utilities provided by the controls' manufacturers. You can also download and install an extensive variety of sample controls from the Microsoft ActiveX Gallery at <http://www.microsoft.com/activex/gallery> using Internet Explorer 3.0 or 4.0.  
{button ,EF('start',`http://www.microsoft.com/activex/gallery`,1)} [Click here to go to the Microsoft ActiveX Gallery now.](#)
- 2 Choose Control from the Insert menu. The Control submenu appears.**
- 3 Choose ActiveX from the Control submenu.**  
The [Select ActiveX Control dialog box](#) appears.
- 4 Select the desired ActiveX control then click OK.**  
The [ActiveX Control Properties dialog box](#) appears.  
If the ActiveX control does not appear on the list, the system may not have installed the control properly. You can verify this by viewing the list of ActiveX controls in another application such as Visual Basic.
- 5 Set the values for each necessary property in the ActiveX Control, then click OK.**  
The ActiveX Control now appears in the cast. In addition to editing each property for the ActiveX control within the ActiveX Control Properties dialog box, you can view information regarding each method the control supports and each event the control can generate. See the following topics for more information:  
{button ,ji('activex.hlp',`activex\_editing\_properties\_director')}} [Editing property values](#)  
{button ,ji('activex.hlp',`activex\_examining\_methods\_director')}} [Examining methods and events](#)
- 6 Drag the ActiveX control from the cast to the stage.**  
After the ActiveX control appears on the stage, it can be repositioned and resized just like any other Sprite Xtra. While the movie is stopped, the ActiveX control is kept in authoring mode and will not react to mouse or keyboard events. When the movie is played, the control becomes responsive to user input.

{button ,ji('`,`activex\_contents\_authorware')}} [CONTENTS](#)    {button ,ji('`,`activex\_step\_by\_step\_procedures\_authorware')}} [HOW TO](#)

## Inserting an ActiveX control

With the Control Xtra for ActiveX (a type of Sprite Xtra), you can place ActiveX controls (formerly known as OLE/OCX controls) within an Authorware piece. Once inserted, the ActiveX control functions as a Sprite Xtra.

### To insert an ActiveX control onto the flowline:

- 1 Make sure that the ActiveX controls you want to use within Authorware are installed on your system.**  
Most controls have their own installation utilities provided by the controls' manufacturers. You can also download and install an extensive variety of sample controls from the Microsoft ActiveX Gallery at <http://www.microsoft.com/activex/gallery> using Internet Explorer 3.0 or 4.0.  
{button ,EF('start','http://www.microsoft.com/activex/gallery',1)} [Click here to go to the Microsoft ActiveX Gallery now.](#)
- 2 From within Authorware, place the flowline pointer where you want to insert the Sprite Xtra.**
- 3 Choose Control from the Insert menu.**  
The Control submenu appears.
- 4 Choose ActiveX from the Control submenu.**  
The [Select ActiveX Control dialog box](#) appears.
- 5 Select the desired ActiveX control then click OK.**  
The [ActiveX Control Properties dialog box](#) appears.  
If the ActiveX control does not appear on the list, the system may not have installed the control properly. You can attempt to verify this by viewing the list of ActiveX controls in another application such as Visual Basic.
- 6 Set the values for each property in the ActiveX Control, then click OK.**  
The selected ActiveX control is contained within the Sprite Xtra icon that is inserted on the flowline. The icon is automatically named 'ActiveX.' In addition to editing each property for the ActiveX control within the ActiveX Control Properties dialog box, you can also view information regarding each method the control supports and each event the control can generate. See the following topics for more information:  
{button ,ji('activex.hlp','activex\_editing\_properties\_authorware')}} [Editing property values](#)  
{button ,ji('activex.hlp','activex\_examining\_methods\_authorware')}} [Examining methods and events](#)
- 7 Place additional icons on the flowline; rearrange them as you want. Continue to develop your Authorware piece as usual.**  
An Event response type symbol can be set up to respond to events created generated by the control.


[{button ,ji\('`,`activex\\_contents\\_director'\)}} CONTENTS](#)    [{button ,ji\('`,`activex\\_step\\_by\\_step\\_procedures\\_director'\)}} HOW TO](#)

## **Editing property values**

You can edit control property values using the ActiveX properties. For more information on properties, see:

[{button ,ji\('activex.hlp`,`activex\\_control\\_properties\\_director'\)}}    ActiveX Control Properties.](#)

### **To edit control property values:**

- 1 Choose the control to edit in the Select ActiveX Control dialog box, then click OK.**  
The [ActiveX Control Properties dialog box](#) appears.
- 2 Click on the property whose value you want to change.**  
The value appears at the top of the list.
- 3 Modify the property value using one of the following methods:**
  - Enter a new value in the edit box.
  - Choose a value from the dropdown list.
  - Click the ellipsis button, , to display and modify Custom settings.
- 4 Choose the next property to modify or click OK to save your changes.**  
Your changes are saved when you click a new property.


{button ,ji('',`activex\_contents\_authorware')}} [CONTENTS](#)    {button ,ji('',`activex\_step\_by\_step\_procedures\_authorware')}} [HOW TO](#)

## Editing property values

You can edit control property values using the ActiveX properties. For more information on properties, see:

{button ,ji('activex.hlp',`activex\_control\_properties\_authorware')}} [ActiveX Control Properties](#).

### To edit control property values:

- 1 Choose the control to edit in the Select ActiveX Control dialog box, then click OK.**  
The [ActiveX Control Properties dialog box](#) appears.
- 2 Click on the property whose value you want to change.**  
The value appears at the top of the list.
- 3 Modify the property value using one of the following methods:**
  - Enter a new value in the edit box.
  - Choose a value from the dropdown list.
  - Click the ellipsis button, , to display and modify Custom settings.
- 4 Choose the next property to modify or click OK to save your changes.**  
Your changes are saved when you click a new property.

{button ,ji('`,`activex\_contents\_director')}} **CONTENTS**

{button ,ji('`,`activex\_step\_by\_step\_procedures\_director')}} **HOW TO**

## **Examining methods and events**

You can examine the methods and events associated with ActiveX controls. This is useful when you're using a control that does not have documentation.

**See also:**

[ActiveX Control methods](#)

[ActiveX Control events](#)

### **To examine control methods or events:**

- 1 Choose the control to edit in the Select ActiveX Control dialog box, then click OK.**  
The [ActiveX Control Properties dialog box](#) appears.
- 2 Click the Methods or Events tab.**  
Either the [ActiveX Control Methods dialog box](#) or the [ActiveX Control Events dialog box](#) appears depending on your choice.
- 3 Click a method or event in the list.**  
A description appears at the top of the list.
- 4 Click OK when you're finished examining the methods or events.**

`{button ,ji('',`activex_contents_authorware')}`` [CONTENTS](#) `{button ,ji('',`activex_step_by_step_procedures_authorware')}`` [HOW](#)  
[TO](#)

## Examining methods and events

You can examine the methods and events associated with ActiveX controls. This is useful when you're using a control that does not have documentation.

See also:

[ActiveX Control Methods](#)

[ActiveX Control Events](#)

### To examine control methods or events:

- 1 **Choose the control to edit in the Select ActiveX Control dialog box, then click OK.**  
The [ActiveX Control Properties dialog box](#) appears.
- 2 **Click the Methods or Events tab.**  
Either the [ActiveX Control Methods dialog box](#) or the [ActiveX Control Events dialog box](#) appears depending on your choice.
- 3 **Click a method or event in the list.**  
A description appears at the top of the list.
- 4 **Click OK when you're finished examining the methods or events.**

{button ,ji('`,`activex\_contents\_director')}} **CONTENTS**      {button ,ji('`,`activex\_step\_by\_step\_procedures\_director')}} **HOW TO**

## **Downloading ActiveX controls within a Director movie**

You can download an ActiveX control dynamically from within a Director movie as it plays. When the end user encounters the portion of the movie or piece that specifies a download, the control is downloaded.

### **To download controls dynamically:**

- 1 Choose the control to download in the Select ActiveX Control dialog, then click OK.**  
The **ActiveX Control Properties dialog box** appears.
- 2 Click the URL button.**  
The **ActiveX Control URL dialog box** appears.
- 3 Enter the full URL of the control to download.**  
The specified URL must match the URL for the ActiveX control exactly. If it does not, the download fails.
- 4 To use the latest version of the control, keep the default -1, -1, -1, -1 for the version number.**  
If you want to download an older version of the control, enter the version number. The version number must match a version exactly. If it does not, the download fails.
- 5 Click OK.**  
When the end user encounters the portion of the movie or piece that specifies a download, the application tries to retrieve the control. If the URL is incorrect or the control's version number is not available, the download fails. If the URL and version number are correct, the application downloads the control. The system confirms that the user wants to install the control before installing it.



{button ,ji('`,`activex\_contents\_authorware')}} **CONTENTS**    {button ,ji('`,`activex\_step\_by\_step\_procedures\_authorware')}} **HOW TO**

## **Downloading ActiveX controls within an Authorware piece**

You can download an ActiveX control dynamically from within an Authorware piece as it plays. When the end user encounters the portion of the movie or piece that specifies a download, the control is downloaded.

### **To download controls dynamically:**

- 1 Choose the control to download in the Select ActiveX Control dialog box, then click OK.**  
The **ActiveX Control Properties dialog box** appears.
- 2 Click the URL button.**  
The **ActiveX Control URL dialog box** appears.
- 3 Enter the full URL of the control to download.**  
The specified URL must match the URL for the ActiveX control exactly. If it does not, the download fails.
- 4 To use the latest version of the control, keep the default -1, -1, -1, -1 for the version number.**  
If you want to download an older version of the control, enter the version number. The version number must match a version exactly. If it does not, the download fails.
- 5 Click OK.**  
When the end user encounters the portion of the movie or piece that specifies a download, the application tries to retrieve the control. If the URL is incorrect or the control's version number is not available, the download fails. If the URL and version number are correct, the application downloads the control. The system confirms that the user wants to install the control before installing it.

### Select ActiveX Control dialog box

The Select ActiveX Control dialog box lets you select and insert a control.

- |               |   |
|---------------|---|
| <b>List</b>   | Displays the ActiveX controls installed on your system.   |
| <b>Search</b> | Lets you scroll to an installed control. Enter the name of the control, then press Enter. The first occurrence of the name is highlighted. Pressing Enter again displays the next occurrence of the name. |
| <b>OK</b>     | Inserts the selected control and closes the dialog box.   |
| <b>Cancel</b> | Closes the dialog box without inserting the selected control.   |
| <b>Help</b>   | Displays Help for the dialog box.   |

## ActiveX Control Properties dialog box

The ActiveX Control Properties dialog box lets you examine and modify control property values.


**Methods tab** Displays the [ActiveX Control Methods dialog box](#).

**Events tab** Displays the [ActiveX Control Events dialog box](#).

**Edit box** Lets you edit the highlighted property value. Your changes are saved after you click a different property.

You can modify the property value using one of the following methods:

Enter a new value in the edit box.

- Choose a value from the dropdown list.
- Click the ellipsis button, , to display and modify Custom settings.

**List** Displays the control properties.

**Select** Lets you select a different control.

**OK** Saves your changes and closes the dialog box.

**Cancel** Closes the dialog box without saving your changes.

**URL** Displays the [ActiveX Control URL dialog box](#).

**Custom** Displays a dialog box showing Custom properties for the control. Some controls may not provide this feature.

**About** Displays information about the control. Some controls may not provide this feature.

**Help** Displays Help for the dialog box.

### ActiveX Control Methods dialog box

The ActiveX Control Methods dialog box lets you examine control methods.

|                        |  |
|------------------------|--|
| <b>Properties tab</b>  | Displays the <a href="#"><u>ActiveX Control Properties dialog box</u></a> .                                  |
| <b>Events tab</b>      | Displays the <a href="#"><u>ActiveX Control Events dialog box</u></a> .                                      |
| <b>Description box</b> | Displays information about the method.   |
| <b>List</b>            | Displays the control methods.  |
| <b>Select</b>          | Lets you select a different control.   |
| <b>OK</b>              | Saves your changes and closes the dialog box.  |
| <b>Cancel</b>          | Closes the dialog box without saving your changes.   |
| <b>URL</b>             | Displays the <a href="#"><u>ActiveX Control URL dialog box</u></a> .   |
| <b>Custom</b>          | Displays a dialog box showing Custom properties for the control. Some controls may not provide this feature. |
| <b>About</b>           | Displays information about the control. Some controls may not provide this feature.                          |
| <b>Help</b>            | Displays Help for the dialog box.  |

### ActiveX Control Events dialog box

The ActiveX Control Events dialog box lets you examine control events.

**Properties tab** Displays the [ActiveX Control Properties dialog box](#).

**Methods tab** Displays the [ActiveX Control Methods dialog box](#).

**Description box** Displays information about the event.

**List** Displays the control events.

**Select** Lets you select a different control.

**OK** Saves your changes and closes the dialog box.

**Cancel** Closes the dialog box without saving your changes.

**URL** Displays the [ActiveX Control URL dialog box](#).

**Custom** Displays a dialog box showing Custom properties for the control. Some controls may not provide this feature.

**About** Displays information about the control. Some controls may not provide this feature.

**Help** Displays Help for the dialog box.

### ActiveX Control URL dialog box

The ActiveX Control URL dialog box lets you specify that a download of an ActiveX control will occur dynamically from within a Director movie or Authorware piece as it plays. When the end user encounters the portion of the movie or piece that specifies a download, the control is downloaded.

**Download from URL** Enter the full URL of the control to download. The specified URL must match the URL for the ActiveX control exactly. If it does not, the download fails.

**Version to Download** Enter the version number of the control to download. To use the latest version of the control, keep the default -1, -1, -1, -1 for the version number. If you want to download an older version of the control, enter the version number. The version number must match a version exactly. If it does not, the download fails.

**CLASSID** Displays the class ID of the ActiveX control. The value displayed here can be copied and pasted into the [Control Xtra for ActiveX Scripting Functions](#).

**OK** Saves the specification and closes the dialog box.

When the end user encounters the portion of the movie or piece that specifies a download, the application tries to retrieve the control. If the URL is incorrect or the control's version number is not available, the download fails. If the URL and version number are correct, the application downloads the control. The system confirms that the user wants to install the control before installing it.

**Cancel** Closes the dialog box without saving the download specification.

**Help** Displays Help for the dialog box.

{button ,ji('`,`activex\_contents\_director')}} **CONTENTS**

## **Control Xtra for ActiveX Scripting functions**

The Control Xtra for ActiveX provides information about its status and control over features (such as security and downloading) through a set of scripting functions. These functions can be called from Lingo in Director. They are implemented as global functions in a Scripting Xtra within the Control Xtra for ActiveX.

The global functions provided by the Control Xtra for ActiveX are: ActiveXInstalled, ActiveXSecuritySetting, ActiveXDownloadSetting, ActiveXSecurityDialog, ActiveXControlQuery, ActiveXControlDownload, ActiveXControlRegister, and ActiveXControlUnregister.

### **ActiveXInstalled**

**Syntax:** ActiveXInstalled

Returns -1 if ActiveX support is installed, otherwise 0.

Determines whether ActiveX support is currently installed on the system. If ActiveX support isn't installed, the multimedia title shouldn't use the Control Xtra for ActiveX.

### **ActiveXSecuritySetting**

**Syntax:** ActiveXSecuritySetting

Returns ActiveX security setting: 'High', 'Medium', or 'None'.

This setting is used by the system when ActiveX controls are downloaded and installed from a URL.

**High** The ActiveX control referenced by the URL must be a digitally signed with a valid signature before the control is installed into the system. Otherwise the download and installation for the control fails and a warning is displayed to the user.

**Medium** The ActiveX control referenced by the URL does not have to be digitally signed with a valid signature before the control is installed, but the user will be warned if the signature is missing or invalid. If the user elects not to install the control, the download and installation will fail.

**None** There is no security enabled and the user may not be warned about dangerous content when an ActiveX control is downloaded and installed on the system.

### **ActiveXDownloadSetting**

**Syntax:** ActiveXDownloadSetting

Returns ActiveX download setting: 'Enabled' or 'Disabled'. This setting allows the user to specify whether the download of ActiveX content is allowed, regardless of the ActiveX security level.

**Enabled** Download of ActiveX controls from a URL is enabled.

**Disabled** Download of ActiveX controls from a URL is disabled.

If a multimedia title attempts to download and install an ActiveX control when the download setting is disabled, an error message will be displayed to the user warning that the control could not be accessed.

### **ActiveXSecurityDialog**

**Syntax:** ActiveXSecurityDialog

Displays a dialog box for the user to set ActiveX security settings. This dialog box is modeled after the security dialog boxes within Microsoft Internet Explorer 3.02.

Since the Control Xtra for ActiveX does not allow the system security settings to be changed from a script handler, displaying the security dialog box is the only way to dynamically change the settings in a multimedia title. Only allowing the user to modify security settings is done for obvious reasons.

Typically, a multimedia title that uses ActiveX controls will use the ActiveXSecuritySetting and ActiveXDownloadSetting handlers to determine the current system security settings. If the settings are inappropriate for the multimedia title, the user can be asked to adjust the security settings to the desired

level from the Control Xtra for ActiveX security dialog box.

### ActiveXControlQuery

**Syntax:** ActiveXControlQuery string CLASSID

Returns -1 if the ActiveX control is installed, otherwise 0. This handler can be used to determine if an ActiveX control is currently installed on the system.

The CLASSID value must be the unique 64-bit identifier for the control. For example the following values is the CLASSID for the Microsoft Calendar Control:

```
"{8E27C92B-1264-101C-8A2F-040224009C02}"
```

### ActiveXControlDownload

**Syntax:** ActiveXControlDownload string CLASSID, string URL, integer VER1, integer VER2, integer VER3, integer VER4

Downloads a particular version of the specified ActiveX control from a URL. Returns -1 if the download succeeded, otherwise 0.

The CLASSID value must be the unique 64-bit identifier for the control. For example the following values is the CLASSID for the Microsoft Calendar Control:

```
"{8E27C92B-1264-101C-8A2F-040224009C02}"
```

The URL value is the URL from which to download and install the ActiveX control from. It can be a local file URL as well. If the value is just the filename of the control installation file, it must be found on the application search path for Director or Authorware.

The version parameters specify the version of the control to be installed, or these parameters can all be set to -1 to use the latest version of the control.

### ActiveXControlRegister

**Syntax:** ActiveXControlRegister string FILENAME

Locates the specified control file on the application path and calls the DllRegisterServer entry point. Typically, this will install an ActiveX control that is self registering. Returns -1 if the install succeeded, otherwise 0.

Because this handler does not download an ActiveX control, it is not affected by the ActiveX security settings.

### ActiveXControlUnregister

**Syntax:** ActiveXControlUnregister string FILENAME

Locates the specified control file on the application path and calls the DllUnregisterServer entry point. Typically, this will uninstall an ActiveX control that is self registering. Returns -1 if the install succeeded, otherwise 0.

Because this handler does not download an ActiveX control, it is not affected by the ActiveX security settings.



{button ,ji('`,`activex\_contents\_authorware')}} **CONTENTS**

## **Control Xtra for ActiveX Scripting functions**

The Control Xtra for ActiveX provides information about its status and control over features (such as security and downloading) through a set of scripting functions. These functions can be called from Functions in Authorware. They are implemented as global functions in a Scripting Xtra within the Control Xtra for ActiveX.

The global functions provided by the Control Xtra for ActiveX are: ActiveXInstalled, ActiveXSecuritySetting, ActiveXDownloadSetting, ActiveXSecurityDialog, ActiveXControlQuery, ActiveXControlDownload, ActiveXControlRegister, and ActiveXControlUnregister.

### **ActiveXInstalled**

**Syntax:** ActiveXInstalled

Returns -1 if ActiveX support is installed, otherwise 0.

Determines whether ActiveX support is currently installed on the system. If ActiveX support isn't installed, the multimedia title shouldn't use the Control Xtra for ActiveX.

### **ActiveXSecuritySetting**

**Syntax:** ActiveXSecuritySetting

Returns ActiveX security setting: 'High', 'Medium', or 'None'.

This setting is used by the system when ActiveX controls are downloaded and installed from a URL.

**High** The ActiveX control referenced by the URL must be a digitally signed with a valid signature before the control is installed into the system. Otherwise the download and installation for the control fails and a warning is displayed to the user.

**Medium** The ActiveX control referenced by the URL does not have to be digitally signed with a valid signature before the control is installed, but the user will be warned if the signature is missing or invalid. If the user elects not to install the control, the download and installation will fail.

**None** There is no security enabled and the user may not be warned about dangerous content when an ActiveX control is downloaded and installed on the system.

### **ActiveXDownloadSetting**

**Syntax:** ActiveXDownloadSetting

Returns ActiveX download setting: 'Enabled' or 'Disabled'. This setting allows the user to specify whether the download of ActiveX content is allowed, regardless of the ActiveX security level.

**Enabled** Download of ActiveX controls from a URL is enabled.

**Disabled** Download of ActiveX controls from a URL is disabled.

If a multimedia title attempts to download and install an ActiveX control when the download setting is disabled, an error message will be displayed to the user warning that the control could not be accessed.

### **ActiveXSecurityDialog**

**Syntax:** ActiveXSecurityDialog

Displays a dialog box for the user to set ActiveX security settings. This dialog box is modeled after the security dialog boxes within Microsoft Internet Explorer 3.02.

Since the Control Xtra for ActiveX does not allow the system security settings to be changed from a script handler, displaying the security dialog box is the only way to dynamically change the settings in a multimedia title. Only allowing the user to modify security settings is done for obvious reasons.

Typically, a multimedia title that uses ActiveX controls will use the ActiveXSecuritySetting and ActiveXDownloadSetting handlers to determine the current system security settings. If the settings are inappropriate for the multimedia title, the user can be asked to adjust the security settings to the desired

level from the Control Xtra for ActiveX security dialog box.

### ActiveXControlQuery

ActiveXControlQuery string CLASSID

Returns -1 if the ActiveX control is installed, otherwise 0. This handler can be used to determine if an ActiveX control is currently installed on the system.

The CLASSID value must be the unique 64-bit identifier for the control. For example the following values is the CLASSID for the Microsoft Calendar Control:

```
"{8E27C92B-1264-101C-8A2F-040224009C02}"
```

### ActiveXControlDownload

**Syntax:** ActiveXControlDownload string CLASSID, string URL, integer VER1, integer VER2, integer VER3, integer VER4

Downloads a particular version of the specified ActiveX control from a URL. Returns -1 if the download succeeded, otherwise 0.

The CLASSID value must be the unique 64-bit identifier for the control. For example the following values is the CLASSID for the Microsoft Calendar Control:

```
"{8E27C92B-1264-101C-8A2F-040224009C02}"
```

The URL value is the URL from which to download and install the ActiveX control from. It can be a local file URL as well. If the value is just the filename of the control installation file, it must be found on the application search path for Director or Authorware.

The version parameters specify the version of the control to be installed, or these parameters can all be set to -1 to use the latest version of the control.

### ActiveXControlRegister

**Syntax:** ActiveXControlRegister string FILENAME

Locates the specified control file on the application path and calls the DllRegisterServer entry point. Typically, this will install an ActiveX control that is self registering. Returns -1 if the install succeeded, otherwise 0.

Because this handler does not download an ActiveX control, it is not affected by the ActiveX security settings.

### ActiveXControlUnregister

**Syntax:** ActiveXControlUnregister string FILENAME

Locates the specified control file on the application path and calls the DllUnregisterServer entry point. Typically, this will uninstall an ActiveX control that is self registering. Returns -1 if the install succeeded, otherwise 0.

Because this handler does not download an ActiveX control, it is not affected by the ActiveX security settings.

{button ,ji('`,`activex\_contents\_authoware')}} **CONTENTS**

## **Scripting functions**

The Control Xtra for ActiveX provides information about its status and control over features (such as security and downloading) through a set of scripting functions. These functions can be called from either Lingo in Director or Functions in Authorware. They are implemented as global functions in a Scripting Xtra within the Control Xtra for ActiveX.

The global functions provided by the Control Xtra for ActiveX are: ActiveXInstalled, ActiveXSecuritySetting, ActiveXDownloadSetting, ActiveXSecurityDialog, ActiveXControlQuery, ActiveXControlDownload, ActiveXControlRegister, and ActiveXControlUnregister.

### **ActiveXInstalled**

ActiveXInstalled

Returns -1 if ActiveX support is installed, otherwise 0.

Determines whether ActiveX support is currently installed on the system. If ActiveX support isn't installed, the multimedia title shouldn't use the Control Xtra for ActiveX.

### **ActiveXSecuritySetting**

**Syntax:** ActiveXSecuritySetting

Returns ActiveX security setting: 'High', 'Medium', or 'None'.

This setting is used by the system when ActiveX controls are downloaded and installed from a URL.

**High** The ActiveX control referenced by the URL must be a digitally signed with a valid signature before the control is installed into the system. Otherwise the download and installation for the control fails and a warning is displayed to the user.

**Medium** The ActiveX control referenced by the URL does not have to be digitally signed with a valid signature before the control is installed, but the user will be warned if the signature is missing or invalid. If the user elects not to install the control, the download and installation will fail.

**None** There is no security enabled and the user may not be warned about dangerous content when an ActiveX control is downloaded and installed on the system.

### **ActiveXDownloadSetting**

ActiveXDownloadSetting

Returns ActiveX download setting: 'Enabled' or 'Disabled'. This setting allows the user to specify whether the download of ActiveX content is allowed, regardless of the ActiveX security level.

**Enabled** Download of ActiveX controls from a URL is enabled.

**Disabled** Download of ActiveX controls from a URL is disabled.

If a multimedia title attempts to download and install an ActiveX control when the download setting is disabled, an error message will be displayed to the user warning that the control could not be accessed.

### **ActiveXSecurityDialog**

**Syntax:** ActiveXSecurityDialog

Displays a dialog box for the user to set ActiveX security settings. This dialog box is modeled after the security dialog boxes within Microsoft Internet Explorer 3.02.

Since the Control Xtra for ActiveX does not allow the system security settings to be changed from a script handler, displaying the security dialog box is the only way to dynamically change the settings in a multimedia title. Only allowing the user to modify security settings is done for obvious reasons.

Typically, a multimedia title that uses ActiveX controls will use the ActiveXSecuritySetting and ActiveXDownloadSetting handlers to determine the current system security settings. If the settings are

inappropriate for the multimedia title, the user can be asked to adjust the security settings to the desired level from the Control Xtra for ActiveX security dialog box.

### ActiveXControlQuery

**Syntax:** `ActiveXControlQuery string CLASSID`

Returns -1 if the ActiveX control is installed, otherwise 0. This handler can be used to determine if an ActiveX control is currently installed on the system.

The `CLASSID` value must be the unique 64-bit identifier for the control. For example the following values is the `CLASSID` for the Microsoft Calendar Control:

```
"{8E27C92B-1264-101C-8A2F-040224009C02}"
```

### ActiveXControlDownload

**Syntax:** `ActiveXControlDownload string CLASSID, string URL, integer VER1, integer VER2, integer VER3, integer VER4`

Downloads a particular version of the specified ActiveX control from a URL. Returns -1 if the download succeeded, otherwise 0.

The `CLASSID` value must be the unique 64-bit identifier for the control. For example the following values is the `CLASSID` for the Microsoft Calendar Control:

```
"{8E27C92B-1264-101C-8A2F-040224009C02}"
```

The URL value is the URL from which to download and install the ActiveX control from. It can be a local file URL as well. If the value is just the filename of the control installation file, it must be found on the application search path for Director or Authorware.

The version parameters specify the version of the control to be installed, or these parameters can all be set to -1 to use the latest version of the control.

### ActiveXControlRegister

**Syntax:** `ActiveXControlRegister string FILENAME`

Locates the specified control file on the application path and calls the `DllRegisterServer` entry point. Typically, this will install an ActiveX control that is self registering. Returns -1 if the install succeeded, otherwise 0.

Because this handler does not download an ActiveX control, it is not affected by the ActiveX security settings.

### ActiveXControlUnregister

**Syntax:** `ActiveXControlUnregister string FILENAME`

Locates the specified control file on the application path and calls the `DllUnregisterServer` entry point. Typically, this will uninstall an ActiveX control that is self registering. Returns -1 if the install succeeded, otherwise 0.

Because this handler does not download an ActiveX control, it is not affected by the ActiveX security settings.

