

CreateControl, CreateReportControl Methods

- The **CreateControl** method creates a [control](#) on a specified open form.
- The **CreateReportControl** method creates a control on a specified open report.

For example, suppose you are building a custom wizard that allows users to easily construct a particular form. You can use the **CreateControl** method in your wizard to add the appropriate controls to the form.

Syntax

CreateControl(*formname*, *controltype*[, *section*[, *parent*[, *columnname*[, *left*[, *top*[, *width*[, *height*]]]]]])

CreateReportControl(*reportname*, *controltype*[, *section*[, *parent*[, *columnname*[, *left*[, *top*[, *width*[, *height*]]]]]])

The **CreateControl** and **CreateReportControl** methods have the following arguments.

Argument	Description																																								
<i>formname</i> , <i>reportname</i>	A string expression identifying the name of the open form or report on which you want to create the control.																																								
<i>controltype</i>	One of the following intrinsic constants identifying the type of control you want to create. To view these constants and paste them into your code from the Object Browser, click Object Browser  on the Visual Basic toolbar , then click Access in the Project/Library box, and click AcControlType in the Classes box.																																								
	<table border="1"> <thead> <tr> <th>Constant</th> <th>Control</th> </tr> </thead> <tbody> <tr> <td>acLabel</td> <td>Label</td> </tr> <tr> <td>acRectangle</td> <td>Rectangle</td> </tr> <tr> <td>acLine</td> <td>Line</td> </tr> <tr> <td>acImage</td> <td>Image</td> </tr> <tr> <td>acCommandButton</td> <td>Command button</td> </tr> <tr> <td>acOptionButton</td> <td>Option button</td> </tr> <tr> <td>acCheckBox</td> <td>Check box</td> </tr> <tr> <td>acOptionGroup</td> <td>Option group</td> </tr> <tr> <td>acBoundObjectFrame</td> <td>Bound object frame</td> </tr> <tr> <td>acTextBox</td> <td>Text box</td> </tr> <tr> <td>acListBox</td> <td>List box</td> </tr> <tr> <td>acComboBox</td> <td>Combo box</td> </tr> <tr> <td>acSubform</td> <td>Subform</td> </tr> <tr> <td>acObjectFrame</td> <td>Unbound object frame</td> </tr> <tr> <td>acPage</td> <td>Page</td> </tr> <tr> <td>acPageBreak</td> <td>Page break</td> </tr> <tr> <td>acCustomControl</td> <td>ActiveX control</td> </tr> <tr> <td>acToggleButton</td> <td>Toggle button</td> </tr> <tr> <td>acTabCtl</td> <td>Tab control</td> </tr> </tbody> </table>	Constant	Control	acLabel	Label	acRectangle	Rectangle	acLine	Line	acImage	Image	acCommandButton	Command button	acOptionButton	Option button	acCheckBox	Check box	acOptionGroup	Option group	acBoundObjectFrame	Bound object frame	acTextBox	Text box	acListBox	List box	acComboBox	Combo box	acSubform	Subform	acObjectFrame	Unbound object frame	acPage	Page	acPageBreak	Page break	acCustomControl	ActiveX control	acToggleButton	Toggle button	acTabCtl	Tab control
Constant	Control																																								
acLabel	Label																																								
acRectangle	Rectangle																																								
acLine	Line																																								
acImage	Image																																								
acCommandButton	Command button																																								
acOptionButton	Option button																																								
acCheckBox	Check box																																								
acOptionGroup	Option group																																								
acBoundObjectFrame	Bound object frame																																								
acTextBox	Text box																																								
acListBox	List box																																								
acComboBox	Combo box																																								
acSubform	Subform																																								
acObjectFrame	Unbound object frame																																								
acPage	Page																																								
acPageBreak	Page break																																								
acCustomControl	ActiveX control																																								
acToggleButton	Toggle button																																								
acTabCtl	Tab control																																								
<i>section</i>	One of the following intrinsic constants identifying the section that will contain the new control. To view these constants and paste them into your code from the Object Browser, click Object Browser  on the Visual Basic toolbar , then click Access in the Project/Library box, and click AcSection in the Classes box.																																								

Constant	Section
acDetail	(Default) Detail section
acHeader	Form or report header
acFooter	Form or report footer
acPageHeader	Page header
acPageFooter	Page footer
acGroupLevel1Header	Group-level 1 header (reports only)
acGroupLevel1Footer	Group-level 1 footer (reports only)
acGroupLevel2Header	Group-level 2 header (reports only)
acGroupLevel2Footer	Group-level 2 footer (reports only)
	If a report has additional group levels, the header/footer pairs are numbered consecutively, beginning with 9.
<i>parent</i>	A string expression identifying the name of the parent control of an attached control. For controls that have no parent control, use a zero-length string for this argument, or omit it.
<i>columnname</i>	The name of the field to which the control will be bound, if it is to be a data-bound control. If you are creating a control that won't be bound to a field, use a zero-length string for this argument.
<i>left, top</i>	Numeric expressions indicating the coordinates for the upper-left corner of the control in twips .
<i>width, height</i>	Numeric expressions indicating the width and height of the control in twips.

Remarks

You can use the **CreateControl** and **CreateReportControl** methods in a custom wizard to create controls on a form or report. Both methods return a [Control](#) object.

You can use the **CreateControl** and **CreateReportControl** methods only in [form Design view](#) or [report Design view](#), respectively.

You use the *parent* argument to identify the relationship between a main control and a subordinate control. For example, if a text box has an attached label, the text box is the main (or parent) control and the label is the subordinate (or child) control. When you create the label control, set its *parent* argument to a string identifying the name of the parent control. When you create the text box, set its *parent* argument to a zero-length string.

You also set the *parent* argument when you create check boxes, option buttons, or toggle buttons. An option group is the parent control of any check boxes, option buttons, or toggle buttons that it contains. The only controls that can have a parent control are a label, check box, option button, or toggle button. All of these controls can also be created independently, without a parent control.

Set the *columnname* argument according to the type of control you are creating and whether or not it will be [bound](#) to a field in a table. The controls that may be bound to a field include the text box, list box, combo box, option group, and bound object frame. Additionally, the toggle button, option button, and check box controls may be bound to a field if they are not contained in an option group.

If you specify the name of a field for the *columnname* argument, you create a control that is bound to that field. All of the control's properties are then automatically set to the settings of any corresponding field properties. For example, the value of the control's [ValidationRule](#) property will be the same as the value of that property for the field.

Note If your wizard creates controls on a new or existing form or report, it must first open the form or report in [Design view](#).

To remove a control from a form or report, use the [DeleteControl](#) and [DeleteReportControl](#) statements.