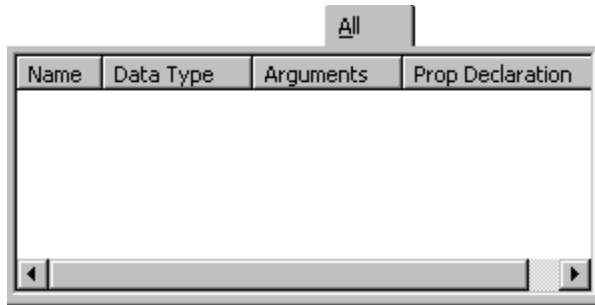




## All Tab (Class Builder Utility)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbdlgAllTabClassBuilderC;vbproBooksOnlineJumpTopic"}



Displays all the members (properties, methods, and events) of the currently selected class or collection, as well as their data type, arguments, and property declaration. The list can be sorted by clicking on the column header of the column to sort by. Properties, methods, and events can be added, renamed, cut, copied, and deleted. The All Tab combines the information from the Properties tab, the Methods tab, and the Events tab into a single display.

## Attributes Tab (Class Module Builder Dialog Box)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbdlgAttributesClassModuleBuilderDialogC;vbproBooksOnlineJumpTopic"}

The screenshot shows a dialog box titled 'Attributes'. It contains three main sections: a 'Description:' label above a large text area with a vertical scrollbar; a 'Project Help File:' label above a single-line text input field; and a 'Help Context ID:' label above a single-line text input field. The dialog box has a standard Windows-style border and a tabbed interface with the 'Attributes' tab selected.

Displays and allows editing of the attributes of the class being added or edited. None of these attributes is mandatory.

### Tab Options

**Description** A description of the class module. This description is embedded in the type library (typelib) of the class module, and will appear in the object browser when the class module is being browsed.

**Project Help File** The path to the help file for the project. This path is displayed here for information only; it can be set in the General tab in the **Project Properties** dialog box.

**Help Context ID** The help context ID in the Project Help File to jump to when help is invoked on this class module.

## Attributes Tab (Event Builder Dialog Box)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbdlgAttributesEventBuilderDialogC;vbproBooksOnlineJumpTopic"}

The screenshot shows a dialog box titled 'Attributes' with a tab labeled 'Attributes'. Inside the dialog, there are three main sections: a 'Description:' label followed by a large text area with a vertical scrollbar; a 'Project Help File:' label followed by a single-line text input field; and a 'Help Context ID:' label followed by a single-line text input field. The dialog has a standard Windows-style border and a close button in the top right corner.

Displays and allows editing of the attributes of the event being added or edited. None of these attributes is mandatory.

### Tab Options

**Description** A description of the event. This description is embedded in the type library (typelib) of the class module, and will appear in the object browser when the event is being browsed.

**Project Help File** The path to the help file for the project. This path is displayed here for information only; it can be set in the General tab in the **Project Properties** dialog box.

**Help Context ID** The help context ID in the Project Help File to jump to when help is invoked on this event.

## Attributes Tab (Method Builder Dialog Box)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbdlgAttributesMethodBuilderDialogC;vbproBooksOnlineJumpTopic"}

The screenshot shows the 'Attributes' tab of a dialog box. It has a title bar with the text 'Attributes'. Below the title bar, there is a label 'Description:' followed by a large text area with a vertical scrollbar. Below the text area, there is a label 'Project Help File:' followed by a single-line text box. Below the text box, there is a label 'Help Context ID:' followed by a single-line text box.

Displays and allows editing of the attributes of the method being added or edited. None of these attributes is mandatory.

### Tab Options

**Description** A description of the method. This description is embedded in the type library (typelib) of the class module, and will appear in the object browser when the method is being browsed.

**Project Help File** The path to the help file for the project. This path is displayed here for information only; it can be set in the General tab in the **Project Properties** dialog box.

**Help Context ID** The help context ID in the Project Help File to jump to when help is invoked on this method.

## Attributes Tab (Property Builder Dialog Box)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbdlgAttributesPropertyBuilderDialogC;vbproBooksOnlineJumpTopic"}

The screenshot shows the 'Attributes' tab of a dialog box. It has a title bar with the text 'Attributes'. Below the title bar, there is a section labeled 'Description:' followed by a large text area with a vertical scrollbar. Below this, there is a section labeled 'Project Help File:' followed by a single-line text input field. At the bottom, there is a section labeled 'Help Context ID:' followed by a single-line text input field.

Displays and allows editing of the attributes of the property being added or edited. None of these attributes is mandatory.

### Tab Options

**Description** A description of the property. This description is embedded in the type library (typelib) of the class module, and will appear in the object browser when the property is being browsed.

**Project Help File** The path to the help file for the project. This path is displayed here for information only; it can be set in the General tab in the **Project Properties** dialog box.

**Help Context ID** The help context ID in the Project Help File to jump to when help is invoked on this property.

## Attributes Tab (Collection Builder Dialog Box)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbdlgAttributesCollectionBuilderDialogC;vbproBooksOnlineJumpTopic"}

The screenshot shows a dialog box titled "Attributes". It contains three main input areas: a large text box for "Description:" with a vertical scrollbar, a text box for "Project Help File:", and a text box for "Help Context ID:". The dialog box has a standard Windows-style border and a title bar.

Displays and allows editing of the attributes of the collection being added or edited. None of these attributes is mandatory.

### Tab Options

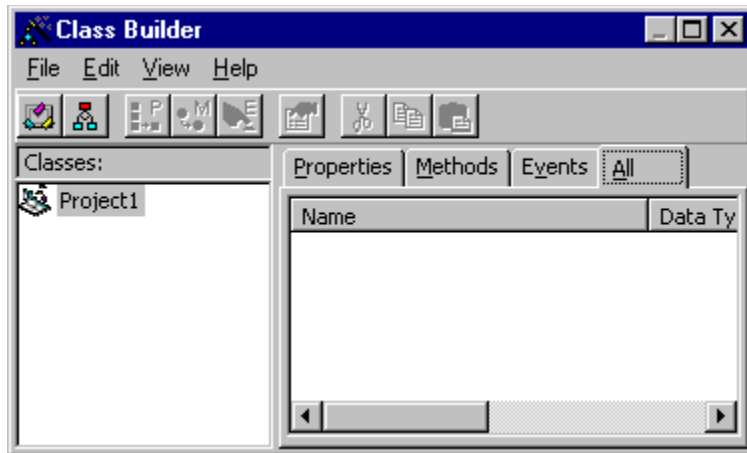
**Description** A description of the collection. This description is embedded in the type library (typelib) of the class module, and will appear in the object browser when the collection is being browsed.

**Project Help File** The path to the help file for the project. This path is displayed here for information only; it can be set in the General tab in the **Project Properties** dialog box.

**Help Context ID** The help context ID in the Project Help File to jump to when help is invoked on this collection.

# Class Builder Utility

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbdlgClassBuilderC;vbproBooksOnlineJumpTopic"}



The Class Builder utility is meant to help you build your class and collection hierarchy for your Visual Basic Project. The Class Builder utility will keep track of the hierarchy of your classes and collections, and generate the framework code necessary to implement the classes and collections, including the properties, methods, and events.

The Object Model Pane shows a visual representation of the hierarchy of all the classes and collections in your project. Clicking on a class or collection will select it for editing. Once a class or collection is selected, the properties, methods, and/or events will appear in the Properties, Methods, and Events Pane. Clicking on one of these members will select it for editing.

Classes and collections that existed in the project before the Class Builder was invoked, even classes and collections that were created by previous invocations of the Class Builder, cannot be edited or deleted by the Class Builder utility. Only classes and collections created in this invocation of the Class Builder utility may be edited or deleted.

If the Class Builder utility is invoked for the first time in a project that already has existing classes and collections, all these classes and collections will initially appear at the root of the hierarchy; the Class Builder does not have information about the hierarchy. You can arrange the existing classes and collections using drag-and-drop, and then the Class Builder will remember this arrangement. Subsequent invocations of the Class Builder in the project will preserve the arrangement of classes and collections.

## Dialog Box Options

**File Menu** Brings up menu items for manipulating the file being built by the Class Builder.

**Edit Menu** Brings up menu items for editing the currently selected item.

**View Menu** Brings up menu items for changing the toolbar, or viewing and changing various options of the Class Builder.

**Help Menu** Brings up menu items for accessing information about the Class Builder utility.

**Toolbar** Contains buttons representing shortcuts for various operations of the Class Builder utility.

**Object Model Pane** Contains a visual representation of the classes and collections in the current project. This view shows the hierarchy of the classes and collections. Clicking on a class or collection will select it.

**Properties, Methods, and Events Pane** Displays the properties, methods and events of the currently selected class or collection. Clicking on one of the column headings will sort the list using that column as the key. Clicking on one of the items will select that item.












- Properties Tab — Displays the properties of the currently selected class or collection.
- Methods Tab — Displays the methods of the currently selected class or collection.
- Events Tab — Displays the events of the currently selected class or collection.
- All Tab — Displays all the members (properties, methods, and events) of the currently selected class or collection.

## Class Builder Utility Toolbar

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbtbrClassBuilderC"}

The Toolbar has various buttons that act as shortcuts for various operations of the Class Builder utility. Pausing the cursor over a button will bring up a description of that button. Pushing a button will cause the associated operation to occur.

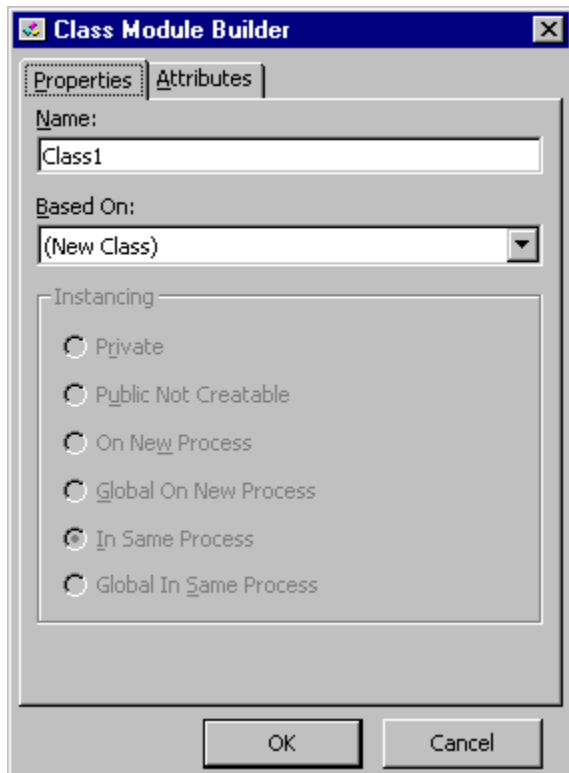
The Class Builder utility toolbar buttons are:

Button	Description
	Adds a new class to the class hierarchy
	Adds a new collection to the class hierarchy
	Adds a new property to the currently selected class or collection
	Adds a new method to the currently selected class or collection
	Adds a new event to the currently selected class or collection
	Edits the properties of the item currently selected in the <b>Class Builder</b> Dialog Box
	Cuts the currently selected item to the clipboard
	Copies the currently selected item to the clipboard
	Pastes the item currently on the clipboard to the currently selected class or collection

## Class Module Builder Dialog Box (Class Builder Utility)

{ewc HLP95EN.DLL,DYNALINK,"See

Also": "vbdlgClassModuleBuilderDialogBoxClassBuilderAddinC;vbproBooksOnlineJumpTopic"}



Allows adding of a new class module or editing of an existing class module. With this dialog box, the various properties and attributes can be specified for a new class module or edited for an existing class module.

The **Class Module Builder** dialog box has these tabs:

Properties

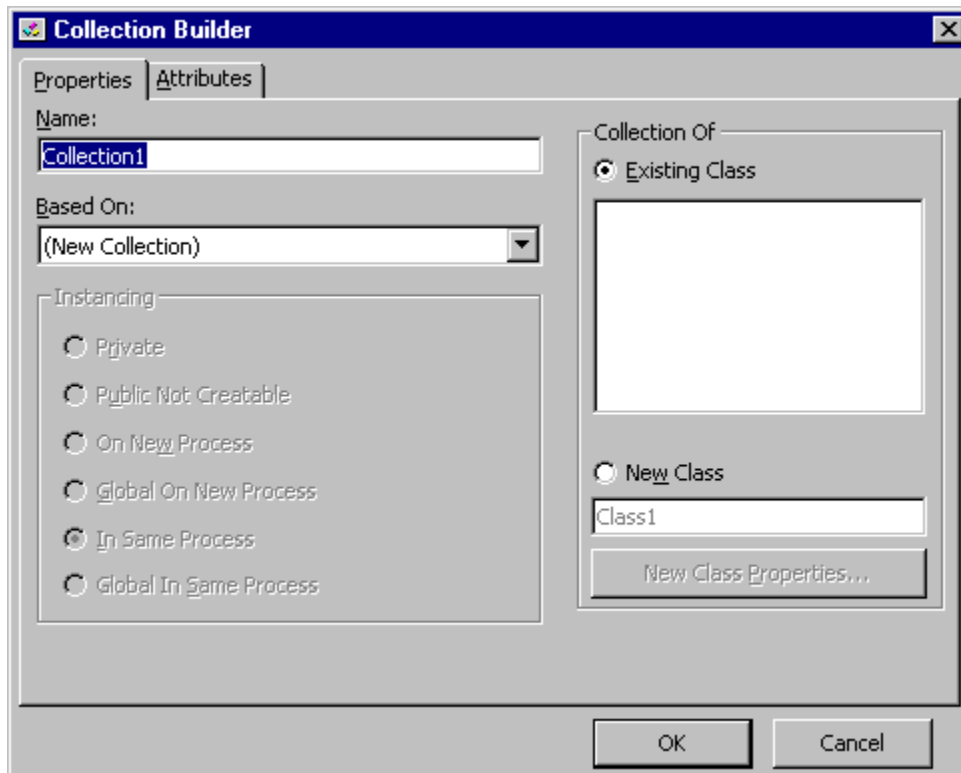
Attributes

Object Creation

## Properties Tab (Class Module Builder Dialog Box)

{ewc HLP95EN.DLL,DYNALINK,"See

Also": "vbdlgClassPropertiesTabClassModuleBuilderDialogC;vbproBooksOnlineJumpTopic"}



Displays and allows editing of the properties of the class module being added or edited.

### Tab Options

**Class Name** The name of the class module.

**Based On** The existing class this class is to be based on. All existing classes are listed in this list box. If the class is to be based on an existing class, the existing base class is selected with this text box. If this class is based on another class, all the properties, methods, and events shells are copied from the base class to this class; none of the code that has been added to the properties, events, and methods shells of the base class is copied over.

**Note** this copying of properties, methods, and events is done only when the class is created; there is no link maintained between the base class and this class, and if the base class is modified after this class is created, the modifications will not be reflected in this class.

**Instancing** Determines how an external application can create an instance of the class module. Other classes in the project containing this class module can always create an instance of this class.

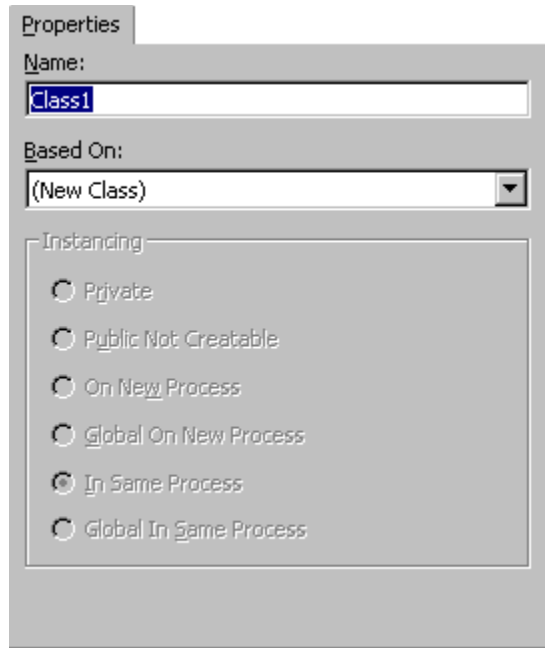
- **Private** – Other applications are not allowed access to the type library information about this class, and cannot create instances of it.
- **Public Not Creatable** – Other applications can use objects of this class only if the project this class is in creates the objects first. Other applications cannot use the **CreateObject** function or the **New** operator to create objects from this class.
- **Multi Use** – Allows other applications to create objects from this class. One instance of the component will provide the objects created with this option.

- Global Multi Use – Like Multi Use, with one addition: properties and methods of this class can be invoked as if they were simply global functions. It's not necessary for the invoking application to explicitly create an instance of this class first; one will automatically be created.
- Single Use – Allows other applications to create objects from this class, but every object of this class that a client creates starts a new instance of this component. Not allowed in ActiveX DLL projects.
- Global Single Use – Like Single Use, with one addition: properties and methods of this class can be invoked as if they were simply global functions. It's not necessary for the invoking application to explicitly create an instance of this class first; one will automatically be created. Not allowed in ActiveX DLL projects.

**This class is a top level object**    Checking this option box makes this class module a top level (root) object.

## Collection Builder Dialog Box (Class Builder Utility)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbdlgCollectionBuilderDialogClassBuilderC;vbproBooksOnlineJumpTopic"}



Allows adding of a new collection or editing of an existing collection. With this dialog box, the various properties and attributes can be specified for a new collection or edited for an existing collection.

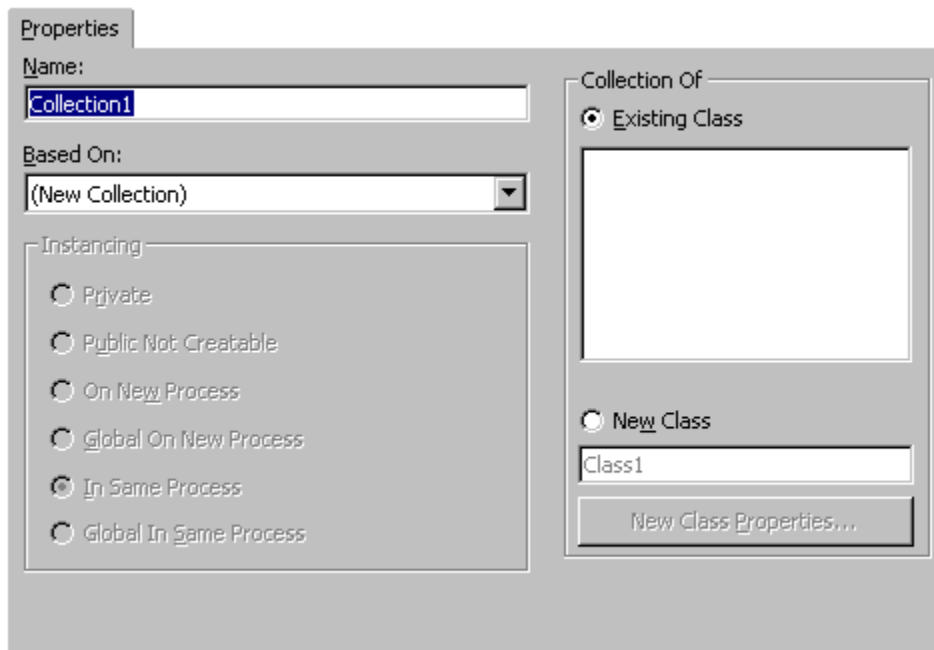
The **Collection Builder** dialog box has these tabs:

Properties

Attributes

## Properties Tab (Collection Builder Dialog Box)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbdlgCollectionPropertiesCollectionBuilderDialogC;vbproBooksOnlineJumpTopic"}



Displays and allows editing of the properties of the collection being added or edited.

### Dialog Box Options

**Collection Name** The name of the collection.

**Based On** The collection this collection is to be based on. All existing collections are listed in this list box. If the collection is to be based on an existing collection, the existing base collection is selected with this text box. If this collection is based on another collection, all the properties, methods, and events shells are copied from the base collection to this collection; none of the code that has been added to the properties, events, and methods shells of the base collection is copied over.

**Note** this copying of properties, methods, and events is done only when the collection is created; there is no link maintained between the base collection and this collection, and if the base collection is modified after this collection is created, the modifications will not be reflected in this collection.

**Collection Of** The class this collection is collecting.

- **Existing Class** — This collection is a collection of an existing class. All existing classes are listed in the list box; the class to collect should be selected from the list box.
- **New Class** — This collection is a collection of a new class. The name of the new class can be edited here, and the **New Class Properties** button will be enabled. Clicking the button will bring up a **Class Module Builder** dialog box so that the properties of the new class can be defined.

**New Class Properties** Button to bring up the **Class Properties** dialog box to create a new class for this collection. This button is enabled when the **Collection Of** property is set to "[New Class]".

**Instancing** Determines how an external application can create an instance of the collection. Other classes in the project containing this collection can always create an instance of this collection.

- **Private** — Other applications are not allowed access to the type library information about this class, and cannot create instances of it.

- **Public Not Creatable** – Other applications can use objects of this class only if the project this class is in creates the objects first. Other applications cannot use the **CreateObject** function or the **New** operator to create objects from this class.
- **Multi Use** – Allows other applications to create objects from this class. One instance of the component will provide the objects created with this option.
- **Global Multi Use** – Like Multi Use, with one addition: properties and methods of this class can be invoked as if they were simply global functions. It's not necessary for the invoking application to explicitly create an instance of this class first; one will automatically be created.
- **Single Use** – Allows other applications to create objects from this class, but every object of this class that a client creates starts a new instance of this component. Not allowed in ActiveX DLL projects.
- **Global Single Use** – Like Single Use, with one addition: properties and methods of this class can be invoked as if they were simply global functions. It's not necessary for the invoking application to explicitly create an instance of this class first; one will automatically be created. Not allowed in ActiveX DLL projects.

**This collection is a top level object**    Checking this option box makes this collection a top level (root) object.



## Cut, Copy, and Paste Commands (Class Builder Edit Menu)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbcmdCutCopyPasteClassBuilderEditC;vbproBooksOnlineJumpTopic"}

These commands are useful for copying and moving items in the class hierarchy. Properties, methods, and events can be cut and copied between classes and collections. Classes and collections may be cut and copied to rearrange the class hierarchy. A class or collection with the same name can exist in two or more places in the class hierarchy; when this happens, it means that there is only one class or collection with that name, and that class or collection appears in several places in the overall class hierarchy.

Classes and collections that existed in the project before the Class Builder was invoked, even classes and collections that were created by previous invocations of the Class Builder, cannot be edited or deleted by the Class Builder utility. Only classes and collections created in this invocation of the Class Builder utility may be edited or deleted.

**Cut Command** Cuts the currently selected item to the Clipboard.

Toolbar shortcut: . Keyboard shortcut: CTRL+X.

**Copy Command** Copies the currently selected item to the Clipboard.

Toolbar shortcut: . Keyboard shortcut: CTRL+C.

**Paste Command** Pastes the item currently on the Clipboard. If the item on the Clipboard is a class or collection, it is pasted into the presently selected location in the class hierarchy. If the item is a property, method, or event, it is added to the currently selected class or collection.

Toolbar shortcut: . Keyboard shortcut: CTRL+V.

## Delete Command (Class Builder File Menu)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbcmdDeleteFileC;vbproBooksOnlineJumpTopic"}

Deletes the currently selected item; either class, collection, property, method or event.

## Edit Menu (Class Builder Utility)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbmnuEditC;vbproBooksOnlineJumpTopic"}

Allows the editing of the currently selected item.

[Cut Command](#)

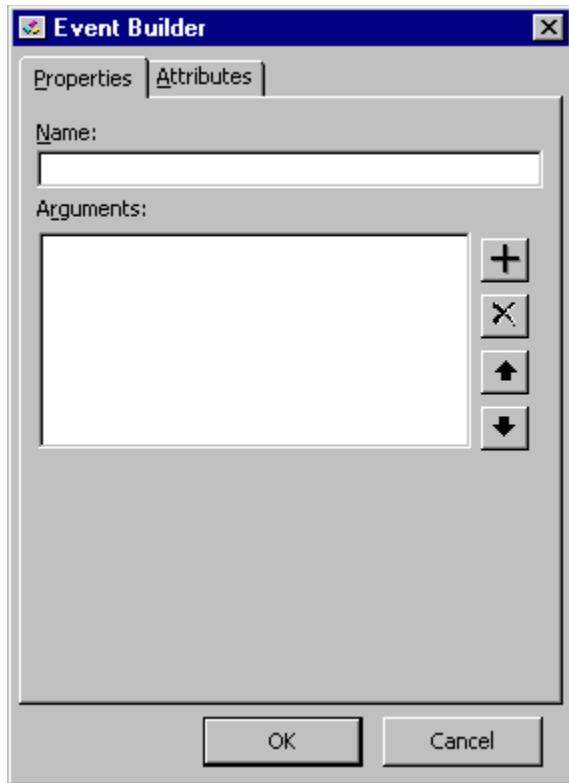
[Copy Command](#)

[Paste Command](#)

[Properties Command](#)

## Event Builder Dialog Box (Class Builder Utility)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbdlgEventBuilderDialogClassBuilderC;vbproBooksOnlineJumpTopic"}



Allows adding a new event or editing an existing event of a class module. With this dialog box, the various properties and attributes can be specified for a new event or edited for an existing event.

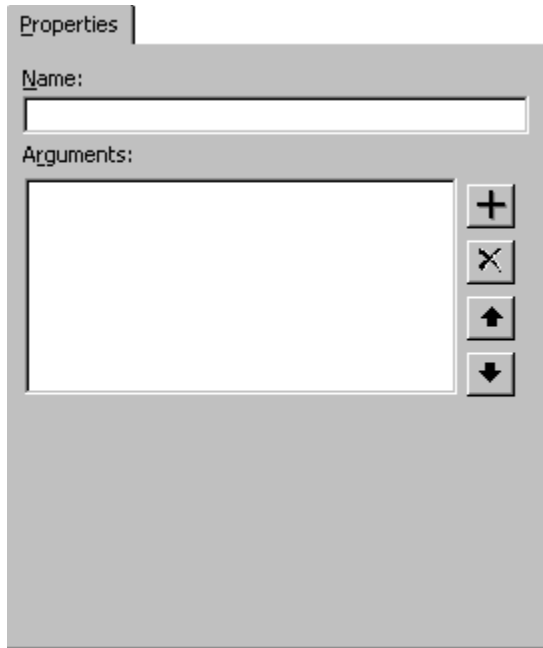
The **Event Builder** dialog box has these tabs:

Properties

Attributes

## Properties Tab (Event Builder Dialog Box)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbdlgEventEventBuilderDialogC;vbproBooksOnlineJumpTopic"}

The image shows a screenshot of the 'Properties' tab in a dialog box. At the top, there is a tab labeled 'Properties'. Below it, there is a label 'Name:' followed by a text input field. Underneath the 'Name' field is a label 'Arguments:' followed by a larger list box. To the right of the list box are four buttons: a plus sign (+), a minus sign (-), an up arrow (↑), and a down arrow (↓).

Displays and allows editing of the properties of the event being added or edited. The **Name** property is required.

### Tab Options

**Name** The name of the event.

**Arguments** The arguments of the event procedure.

- If this box is blank, the event will have no arguments.
- To add an argument, press the "+"; this will bring up the **Add Argument** dialog box.
- To delete an argument, press the "-" when the argument to delete is selected.
- When an argument is selected, the up arrow and down arrow keys will move the argument up or down the list.

## Events Tab (Class Builder Utility Dialog Box)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbdlgEventsTabClassBuilderC;vbproBooksOnlineJumpTopic"}



Displays all the events of the currently selected class or collection, as well as the arguments and their data types. The list of events can be sorted by clicking on the column header of the column to sort by. Events can be added with the **File:New:Event** command or by selecting the **New Event** button in the tool bar; this will bring up the **Event Builder** dialog box. Events can be edited with the **Edit:Events** command or by right-clicking on the event and choosing **Properties**. Events can also be renamed, cut, copied, and deleted.

## Exit Command (Class Builder Utility File Menu)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbcmdExitFileClassC;vbproBooksOnlineJumpTopic"}

Exits from the Class Builder utility. Before exiting, the Class Builder will ask if the project should be updated with the changes.

## File Menu (Class Builder Utility)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbmnuFileC;vbproBooksOnlineJumpTopic"}

This menu has menu items for manipulating the file containing the classes and collections the Class Builder utility is building. The file can be deleted, renamed, saved, or a new file can be started. You exit the Class Builder by choosing the **Exit** menu item.

New Command

Delete Command

Rename Command

Update Project Command

Exit Command



## Help Menu (Class Builder Utility)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbmnuHelpC;vbproBooksOnlineJumpTopic"}

Brings up menu items for information about the Class Builder utility.

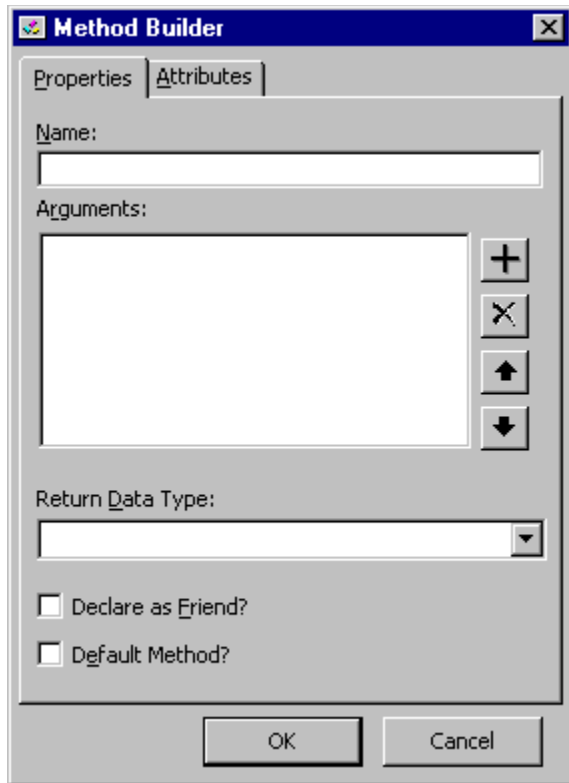
### Commands

**Help Topics Command** Brings up Help for the Class Builder utility.

**About Class Builder Command** Brings up the **About Class Builder** dialog box, which displays the version number and other information about the Class Builder utility.

## Method Builder Dialog Box (Class Builder Utility)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbdlgMethodBuilderDialogClassBuilderC;vbproBooksOnlineJumpTopic"}



Allows adding a new method or editing an existing method of a class module. With this dialog box, the various properties and attributes can be specified for a new method or edited for an existing method.

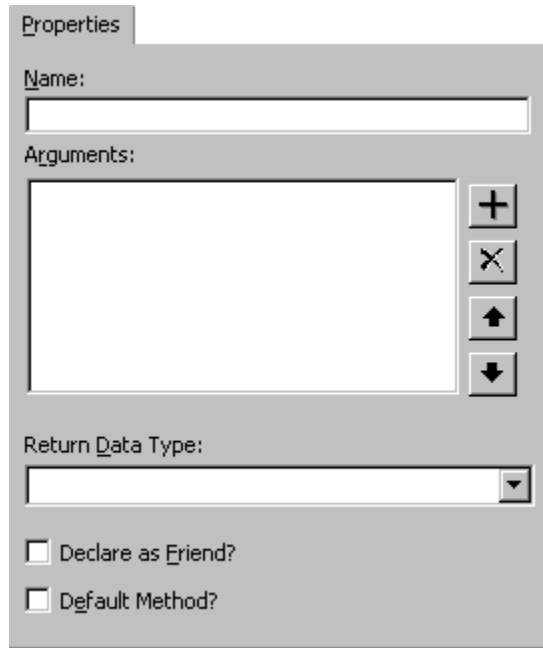
The **Method Builder** dialog box has these tabs:

Properties

Attributes

## Properties Tab (Method Builder Dialog Box)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbdlgMethodMethodBuilderC;vbproBooksOnlineJumpTopic"}



Displays and allows editing the properties of the method being added or edited.

### Tab Options

**Name** The name of the method.

**Arguments** The arguments of the method procedure.

- If this box is blank, the method will have no arguments.
- To add an argument, press the “+”; this will bring up the **Add Argument** dialog box.
- To delete an argument, press the “-” when the argument to delete is selected.
- When an argument is selected, the up arrow and down arrow keys will move the argument up or down the list.

**Return Data Type** The return data type of the method.

If the data type is “(None)”, then the method will be a **Sub**, otherwise it will be a **Function**.

**Declare as Friend?** Indicates whether or not this method is to be declared as a **Friend**.

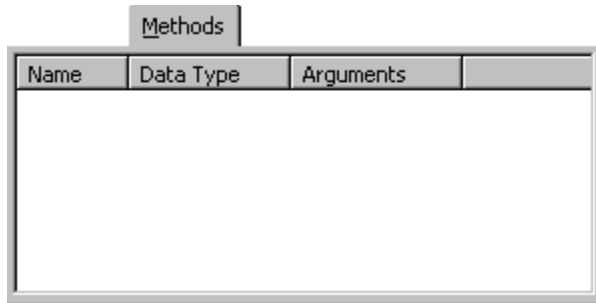
- Checked – This method is declared as a **Friend** method.
- Unchecked – This method is not declared as a **Friend** method.

**Default Method?** Indicates whether or not this method is the default method. The default method is the method that is implicitly accessed when the Class Module is used without specifying a method. Only one method in a class can be the default method, however, a class module may have no default methods.

- Checked – This method is the default method.
- Unchecked – This method is not the default method.

## Methods Tab (Class Builder Utility)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbdlgMethodsTabClassBuilderC;vbproBooksOnlineJumpTopic"}



Displays all the methods of the currently selected class or collection, as well as their arguments and return data type, if any. The list of methods can be sorted by clicking on the column header of the column to sort by. Methods can be added by the **File:New:Method** command or by selecting the **New Method** button in the tool bar; this will bring up the **Method Builder** dialog box. A method can be edited with the **Edit:Method** command or by right clicking on the method and choosing **Properties**. Methods can also be renamed, cut, copied, and deleted.

## New Commands (Class Builder File Menu)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbcmdNewFileC;vbproBooksOnlineJumpTopic"}

Allows adding a new class or collection, or a new member to the currently selected class or collection.

**Class Command** Adds a new class to the currently selected class or collection, and brings up the **Class Module Builder** dialog box.

Toolbar shortcut: 

**Collection Command** Adds a new collection to the currently selected class or collection, and brings up the **Collection Builder** dialog box.

Toolbar shortcut: 

**Property Command** Adds a new property to the currently selected class or collection, and brings up the **Property Builder** dialog box.

Toolbar shortcut: 

**Method Command** Adds a new method to the currently selected class or collection, and brings up the **Method Builder** dialog box.

Toolbar shortcut: 

**Event Command** Adds a new event to the currently selected class or collection, and brings up the **Event Builder** dialog box.

Toolbar shortcut: 

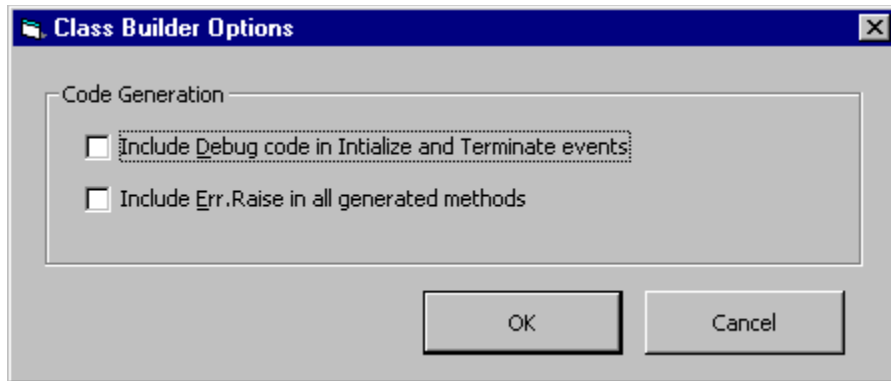
## Options Command (Class Builder View Menu)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbcmdOptionsCommandClassBuilderViewMenuC;vbproBooksOnlineJumpTopic"}

Brings up the **Options** dialog box, which allows setting the options for the Class Builder utility.

## Class Builder Options Dialog Box (Class Builder Utility)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbdlgOptionsDialogBoxClassBuilderAddinC;vbproBooksOnlineJumpTopic"}



Displays and allows editing the options for the Class Builder utility.

### Dialog Box Options

**Code Generation** Allows setting various options for the generation of the code that supports the classes and collections.

- Include debug code in Initialize and Terminate events — Generates conditionally compiled code that will print a notification to the Debug class when objects are created and destroyed. This code is useful to verify that objects are being created and destroyed in the correct sequence, and at the correct time.
- Include Err.Raise in all generated methods — Generates code for handling errors in all **Let**, **Set** and **Get** property procedures and all methods. In addition, a code module called **modErrorHandling** will be added to the project; this module contains a number of useful error handling routines. The code in this module can get an error text string from a resource file and use this string in raising an error.

## Properties Command (Class Builder Edit Menu)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbcmdPropertiesClassBuilderEditC;vbproBooksOnlineJumpTopic"}

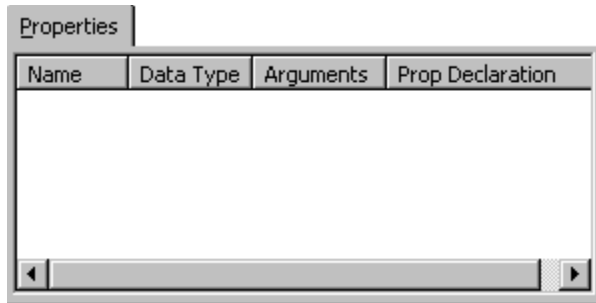
Brings up the **Class Module Builder** dialog box for the currently selected property, method, or event, with the Properties tab selected. This allows editing of the currently selected class member.

Toolbar shortcut: 



## Properties Tab (Class Builder Utility)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbdlgPropertiesTabClassBuilderC;vbproBooksOnlineJumpTopic"}



Displays all the properties of the currently selected class or collection, as well as their data type and property declaration. The list of properties can be sorted by clicking on the column header of the column to sort by. Properties can be added with the **File:New:Property** command or by selecting the **New Property** button in the toolbar; this will bring up the **Property Builder** dialog box. A property can be edited by using the **Edit:Properties** command or by right clicking on the property and choosing **Properties**. Properties can also be renamed, cut, copied, and deleted.

## Property Builder Dialog Box (Class Builder Utility)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbdlgPropertyBuilderDialogClassBuilderC;vbproBooksOnlineJumpTopic"}

The screenshot shows the 'Property Builder' dialog box with the 'Properties' tab active. The 'Name' field contains 'Collection1'. The 'Based On' dropdown is set to '(New Collection)'. Under the 'Instancing' section, the 'In Same Process' radio button is selected. On the right side, under 'Collection Of', the 'Existing Class' radio button is selected, and there is an empty list box below it. The 'New Class' radio button is also present, with a text box containing 'Class1' and a 'New Class Properties...' button below it.

Allows adding a new property or editing an existing property of a class module. With this dialog box, the various properties and attributes can be specified for a new property or edited for an existing property.

The **Property Builder** dialog box has these tabs:

Properties

Attributes

## Properties Tab (Property Builder Dialog Box)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbdlgPropertyPropertyBuilderDialogC;vbproBooksOnlineJumpTopic"}

The screenshot shows the 'Properties' tab of a dialog box. It contains a 'Name' label followed by a text input field. Below that is a 'Data Type' label followed by a dropdown menu currently showing 'Variant'. The 'Declaration' section is enclosed in a rounded rectangle and contains three radio button options: 'Public Property (Let, Get, Set)' (which is selected), 'Friend Property (Let, Get, Set)', and 'Public Variable'. At the bottom of this section is an unchecked checkbox labeled 'Default Property?'.

Displays and allows editing the properties of the property being added or edited.

### Tab Options

**Name** The name of the property.

**Data Type** The data type of the property. The drop down box lists all possible data types and known classes and collections of the project, as well as a blank entry. The name of an unknown data type can also be typed in.

**Declaration** Determines how the property is implemented.

- **Public Property (Let, Get, Set)** – The property is implemented with two procedures: a **Get** procedure, and either a **Let** or **Set** procedure. If the data type of the property is an object, a **Set** procedure is used, otherwise a **Let** procedure is used.
- **Friend Property (Let, Get, Set)** – The property is implemented with two procedures: a **Friend Get** procedure, and either a **Friend Let** or **Friend Set** procedure. If the data type of the property is an object, a **Friend Set** procedure is used, otherwise a **Friend Let** procedure is used.
- **Public Variable** – The property is a public variable.

**Default Property?** Indicates whether or not this property is the default property. The default property is the property that is implicitly accessed when the Class Module is used without specifying a property. Only one property in a class can be the default property, however, a class module may have no default properties.

- **Checked** – This property is the default property.
- **Unchecked** – This property is not the default property.

## Rename Command (Class Builder File Menu)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbcmdRenameFileC;vbproBooksOnlineJumpTopic"}

Allows the renaming of the currently selected class, collection, property, method or event.

## Update Project Command (Class Builder File Menu)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbcmdSaveFileClassC;vbproBooksOnlineJumpTopic"}

Updates the project with all the Class Builder changes, including generating the code if that option is selected.

Keyboard shortcut: CTRL+S.

## Set As Collection or Class (Class Builder Shortcut Menu)

{ewc HLP95EN.DLL,DYNALINK,"See

Also": "vbcmdSetAsCollectionOrClassClassBuilderShortCutC;vbproBooksOnlineJumpTopic"}

Allows changing the currently selected class to a collection, or the currently selected collection to a class. This option is available by right-clicking a class or collection.

## Toolbar Command (Class Builder View Menu)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbcmdToolbarClassBuilderViewC;vbproBooksOnlineJumpTopic"}

Enables or disables the Toolbar; if this command is checked, the Toolbar is displayed.

## View Menu (Class Builder Utility)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbmnuViewC;vbproBooksOnlineJumpTopic"}

Brings up menu items to either toggle viewing of the Toolbar or to change various options of the Class Builder utility.

Toolbar Command

Options Command



## Object Creation Tab (Class Module Builder Dialog Box)

{ewc HLP95EN.DLL,DYNALINK,"See

Also":"vbdlgObjectCreationTabClassModuleBuilderDialogBoxC;vbproBooksOnlineJumpTopic"} {ewc

HLP95EN.DLL,DYNALINK,"Specifics":"vbdlgObjectCreationTabClassModuleBuilderDialogBoxS"} {ewc



Displays and allows editing of the object creation options of the class module being added or edited.

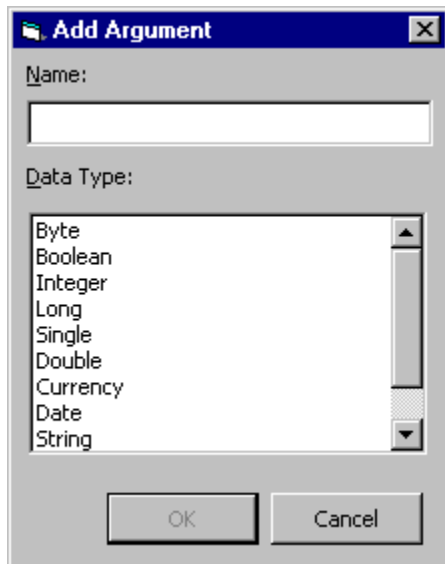
### Tab Options

**Class Initialize Event** The object is created when the parent is initialized. This option slows down the startup of the parent, but does not slow down the accessing of the object property.

**Property Get on first access** The object is not created until the object property **Get** procedure is used. This option slows down all object property accesses including the initial one, but does not slow down the initialization of the parent.

## Add Argument Dialog Box (Property Tab)

```
{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbdlgAddArgumentDialogBoxPropertyTabC;vbproBooksOnlineJumpTopic"}  
{ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbdlgAddArgumentDialogBoxPropertyTabS"}
```



Allows adding an argument to a method or event.

### Dialog Box Options

**Name** The name of the argument.

**Data Type** The data type of the argument. The box lists all possible data types and classes and collections of the project.

