

Code components, formerly called OLE servers, are libraries of objects that provide an easy way to package your code for reuse. For example, you can create libraries of procedures that can be used with Visual Basic or with desktop applications that host Visual Basic for Applications, such as Microsoft Office.

Note If you are using the Control Creation Edition of Visual Basic, some of the material covered in this document may not be applicable. With the full editions of Visual Basic you have the ability to create applications, ActiveX documents, and other types of components. Although some of the terminology may relate to application specific objects such as forms, in most cases the underlying concepts also apply to ActiveX control creation.

The following topics describe key features of code components created with Visual Basic.

Contents

- Displaying Forms From Code Components

Displaying Forms from Code Components

Code components can display modal or modeless forms, subject to certain restrictions.

In-Process Components that Show Modeless Forms

In-process components (.dll and .ocx files) can display modeless forms only in client processes that implement the `IMsoComponentManager` interface.

The reason is that modeless forms cannot function correctly unless they communicate with the client's message loop. This communication is provided by the `IMsoComponentManager` interface.

For many applications, support for `IMsoComponentManager` depends on the version.

- Applications created with Visual Basic 5.0 automatically provide `IMsoComponentManager` support. Older versions of Visual Basic do not.
- Microsoft Office 97 supports `IMsoComponentManager`. Older versions of Microsoft Office applications do not.
- Version 3.0 of Microsoft Internet Explorer does not provide `IMsoComponentManager` support.

Determining `IMsoComponentManager` Support Run Time

To allow in-process components to detect whether a client application supports the IMsoComponentManager interface, Visual Basic provides the Boolean NonModalAllowed property of the App object.

An in-process component should test this property before showing a modeless form. If the value is True, the form can be shown vbModeless. If the value is false, showing a modeless form will cause run-time error 369. The component should degrade gracefully by showing the form vbModal instead.

Important Debugging Limitation

When you're debugging an in-process component in the development environment, using a non-Visual Basic application as an out-of-process test program, modeless forms will appear to work even if the non-Visual Basic application doesn't provide IMsoComponentManager support.

Testing the NonModalAllowed property of the App object is of no use in this situation; the property will return True regardless of the client. The reason for this is that when debugging an in-process component with an out-of-process test application, Visual Basic cannot determine whether the test application supports IMsoComponentManager.

Before attempting to write an in-process component that will show modeless forms, for use with a specific non-Visual Basic application, you should compile a small test DLL with a method displays the value of the NonModalAllowed property of the App object in a message box. Call this method from the non-Visual Basic application, to determine whether it supports IMsoComponentManager.

Forms Displayed by Out-Of-Process Components

Modal and modeless forms displayed as a result of method calls to out-of-process components will not necessarily appear in front of the client application's forms. The reason is that Automation does not define a z-order relationship between forms in different applications.