

A member of ComboPro ActiveX Controls Suite

FontCombo[®] ActiveX Control

Operation Manual

Version 1.0.0.1

Contents

1. General	2
I. Installation	2
II. Register control	2
III. Add this control into your project	2
IV. Set Properties in Properties Browser at Design Mode	5
V. Handle Properties by programming	6
VI. Handle Methods by programming.....	6
VII. Handle Events by programming	6
VIII. Some support before running your application.....	7
IX. Deregister control	7
X. Uninstall this control	7
XI. Distribute Application combined this product	7

1. General

FontCombo ActiveX Control is an ActiveX control based on standard combo box for all software developers. it is a member of ComboPro ActiveX Controls Suite.

This control is an ideal tool to select fonts from enumerated font families. it has TrueType/OpenType and other font types recognizing ability, and support multi-language charsets enumerating and font scripts features. It offers powerful items data management capabilities such as add, insert, delete, find, move, get/set information and so on. Item type can be General, Default, Browse and Separate style, all types has own properties, methods and events. Item type, font script, font height, fore/back color, and other properties are completely customizable. This control is Full support XP theme with a .NET development.

I. Installation

ActiveX Control's installation only can be copying the control to any directory simply, but it commends that you install it into system directory, that will registry self by Windows Register Server. This control has a standard installation package file, and a NON end-user (Developer) can run the setup file to extract all files to a specified location.

II. Registry controls

Any ActiveX control must have a registry in current system before using it. You must register this control with Microsoft® Register Server (REGSVR32.EXE), the register server is located at system directory, your control file FontCombo.dll can be placed at any directory, the register command likes following:

C: \WINNT\system32\REGSVR32.EXE <path>\FontCombo.dll

After the registration, you can found the control in any environment which can hold ActiveX control, for example, VB, VC++, HTML, Office Documents and other any application which can handle COM technologies. These applications can create multiple instances to any control, user can specify any supported properties value through modifying the control's property sheet, and enjoy the control's power features using given methods and events.

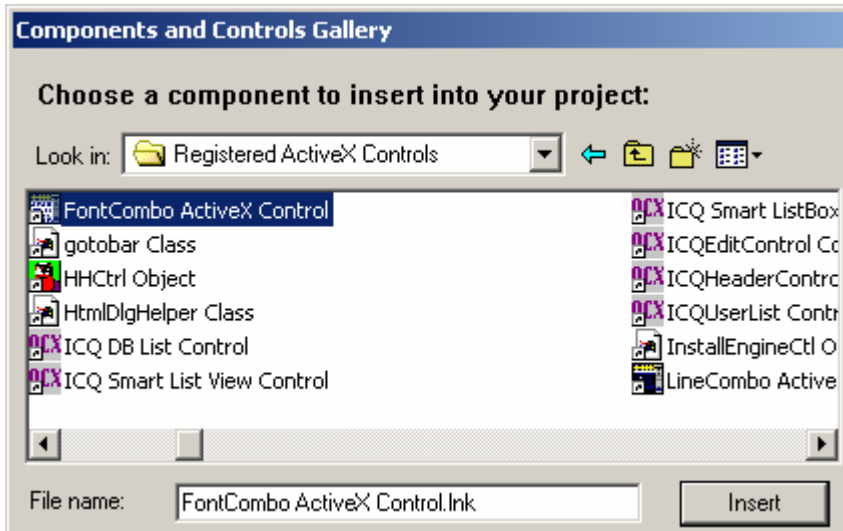
III. Add this control into your project

After made a registry to the control, to use this control, at first developer should add or import it into your project, create a control instance or generate the control wrapper class(es), done these steps, developer can modify this control properties value, handle this control methods and handle events manual or programming.

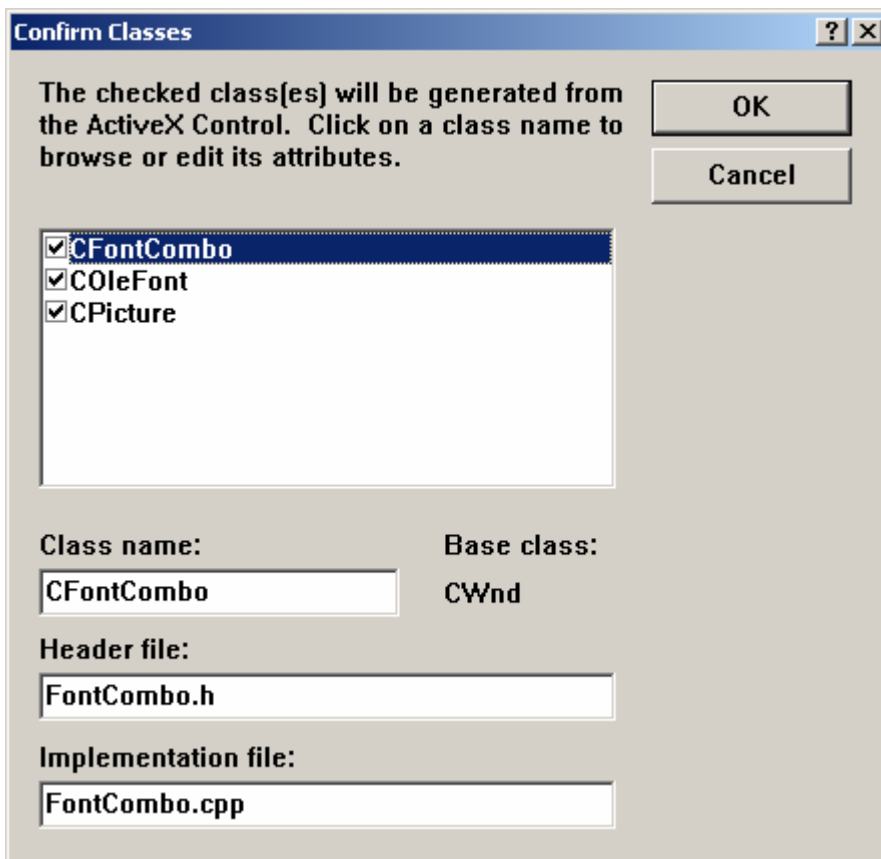
- **Browse this control from Components list**

In Visual C++:

Select "Project"->"Add to Project"->"Components and Controls..."



Select FontCombo ActiveX Control and click "Insert" button.

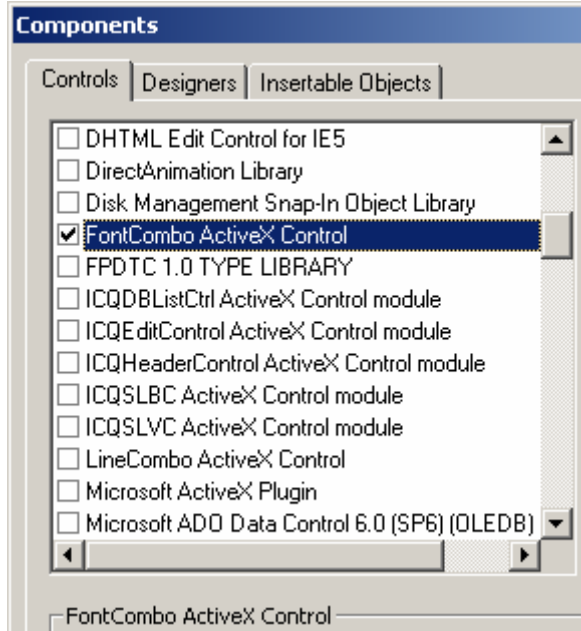


There are three classes in the ActiveX Control Confirm Classes dialog, you must select the first wrapper class(es) **CFontCombo** generated from the ActiveX Control, It's a requisite. **COleFont** class and **CPicture** class are two OLE dispatch driver wrapper classes, they can implement exchange between general windows Font/Picture objects and OLE Font/Picture object, if you hope these classes become intermediates, you can add them into your application. If you are familiar with **IFont** COM font object, which is an object wrapper around a Windows font object, and **IPicture** COM picture object, which is an object wrapper around a Windows picture object, then you can replace them with MFC wrapper classes **CFontHolder** and **CPictureHolder**.

Developer can modify these classes' names to any fair-sounding names.

In Visual Basic:

Select "Project"->"Components..."

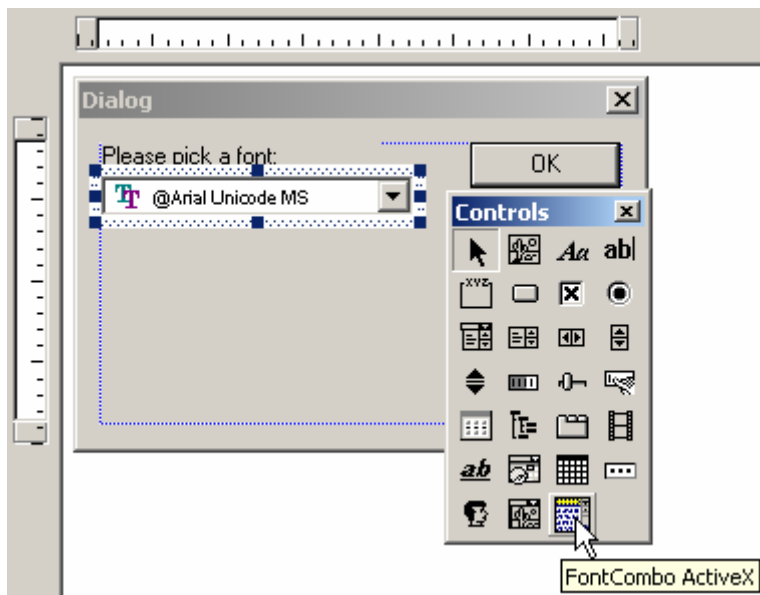


You can pick this control in the components list box, and click OK to add it into your project workspace.

- **Add the control into project**

In Visual C++

There are two methods to add a control into a project, the first is picking a control, drag&drop it and placed it on a dialog from Controls toolbar directly, as following:

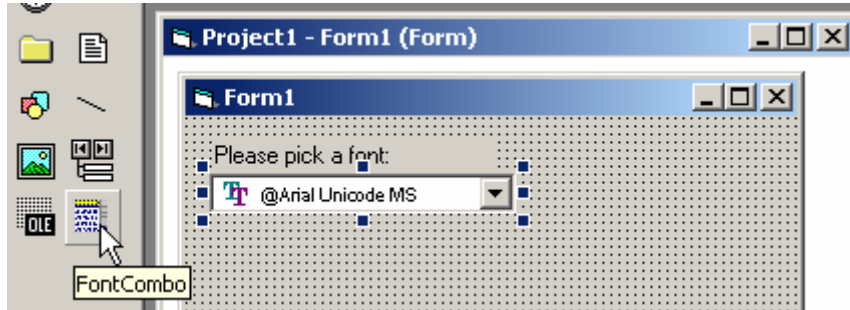


The second method to add the control into a project is to create an instance through programming, not need any integration with user interface directly. You can map this control through Class Wizard's help, and it is the same that to initialize a member

pointer and new a control object before processing it.

In Visual Basic

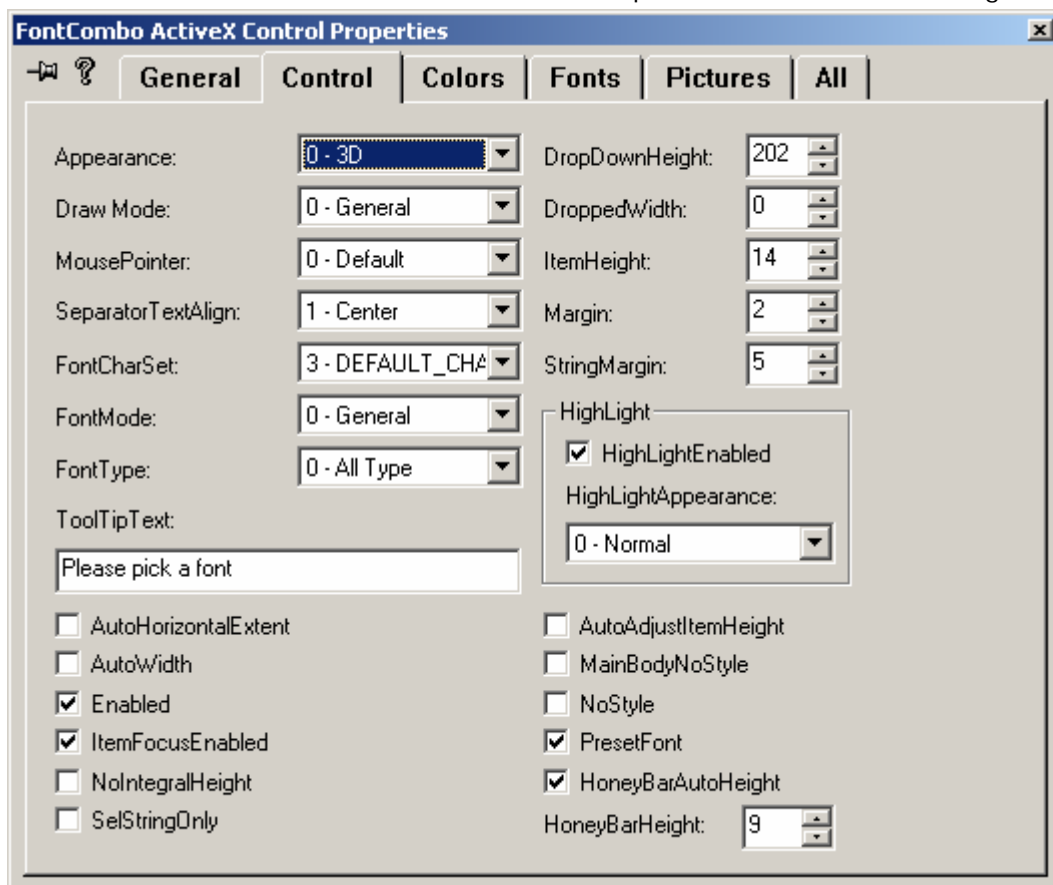
In Visual Basic all handling to the control is easy, you can pick this control from Toolbox toolbar and place it on a Form, as following:



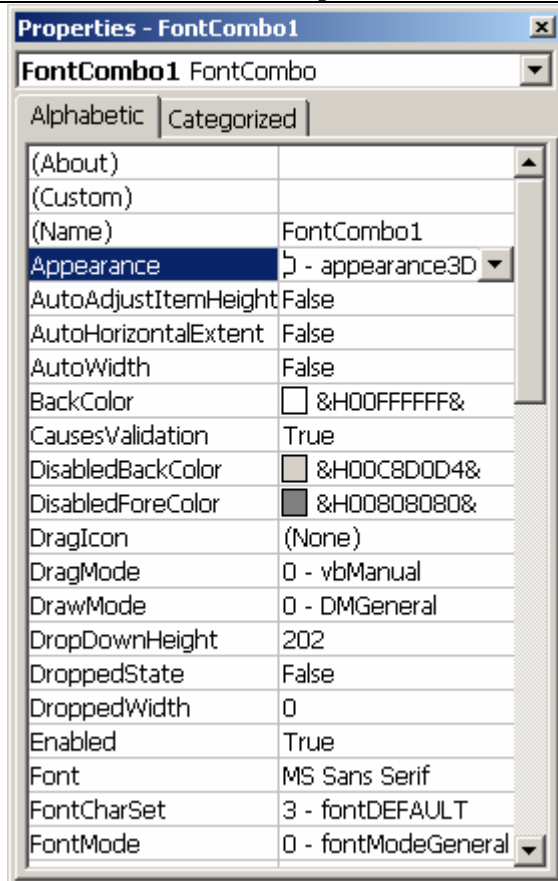
Once the control is added into the form, the component will be a global object, you can use it at anywhere within the project, generally it is named as FontCombo1.

IV. Set Properties in Properties Browser at Design Mode

ActiveX control's one of important character is can specify and modify properties at design mode directly in Properties Browser dialog, it must be any environment which can host an ActiveX control. In Visual C++, this Properties Browser likes following:



In Visual Basic, this Properties Browser likes following:



You can specify and modify all properties according to this browser's list. All specified properties value will be stored in the current dialog resource statically.

V. Handle Properties by programming

Handling properties of the application at runtime is more important to a developer, programming with these properties can apply all features of this control to satisfy your application's requirements, the properties operations should be bounded under the control's generated interfaces, especially in data structure, data type and exception. Every development platform's data type definition is not the same with an exception, You should reference current development platform language and see the control's programmer's guide for more information.

VI. Handle Methods by programming

The Control method is an important character to a control. Like control properties, these methods can provide lots of operations to the control, and many methods also return a value, developer can follow next handling through it. Generally, these methods have some different data type parameters. Handling these methods must follow the generated interface definition. Developer can reference current development platform language and see the control's programmer's guide for more information.

VII. Handle Events by programming

Control event is an interaction interface between the control's inner message handling and user's input/output with input device. Although all events is predefined in control, but there is different executive result when firing this event, so developer can add more

features to handle the issues to satisfy user's requirements. Like properties and methods, these events also have predefined function prototypes, developer must follow these data type to transfer data. The details of the control events can be found at the control's programmer's guide.

VIII. Some supports before running your application

One of important job is to sure the combined control has been registered before an application is running. If an inner control has not registered, the application will appear a warning or error asserts, even it will do nothing besides quitting this program.

IX. Deregister the control

If user hopes a control become unavailable, then deregister it. Deregistering a control just can be to clean all information in registry database and it will not appear in all registered ActiveX controls list anymore, of course this control file still stay at your system, you can reregister it.

If you hope deregister a control, you can use following command:

```
C:\WINNT\system32\REGSVR32.EXE -u <path>\FontCombo.dll
```

After unregistered the control, you can not found and use this control anymore, if you hope reuse this control, you must register this control again follow above steps.

X. Uninstall this control

If you don't need this control anymore, you can uninstall this control, to run the product installation file, follows the installation wizard, you can uninstall this control. All files related to this control will be removed from your system, and you will never found and use it at anywhere.

XI. Distribute Application combined this product

Any application combined an ActiveX control should contains this control while distributing your application, and this control must self-registry when it is copied to end user system, this operation can be done through your installer application, Like InstallShield, WiseInstaller, etc. you can see these product's Operation Manual for more details while making an application installation archive. Of course this operation also can be done manually after the installation is finished.

Anyway, the control file can be release to any end user specified directory. Just need a successful register, it's a key that insure your application can smooth running.