

This article has been written to explain how the ActiveX object works in Multimedia Fusion. Throughout this series, we've seen that many users fail to make use of some powerful features simply because of a lack of detailed documentation. To rectify this, we'll create a very simple application and use it to explain poorly documented or unclear features.

Overview

First of all, I think it could be useful to explain why we wrote the ActiveX object. The ActiveX object is a common MMF object, like Ftp, Picture, ODBC and so on. The only difference is that this object has the potential to do anything: its actions, conditions and expressions change and are not fixed at compilation time - it depends on the control you want to use. If you're a beginner, a way to think of an ActiveX control is as a set of functions and events. Buttons, sliders, Web Browser are ActiveX controls on Windows platforms.

Under Windows, ActiveX controls are usually stored in your system directory and are registered in the registry. This can be done by using the regsrv32 command, or automatically by an installer or Web browser.

Let's start now

We are going to create a totally useless application that will show you how to retrieve data via parameters and how to setup an ActiveX object properly. We hope that you will understand the basis of the using the ActiveX object and will be able to go further.

The simple application we're creating is based on DBGrid control which can be found on every system. First of all, create a new application, then add a blank frame and go to Frame Editor. Then go to Insert menu and select "Insert Object". Select ActiveX object and click on the OK Button. Click again to place it where you want it.

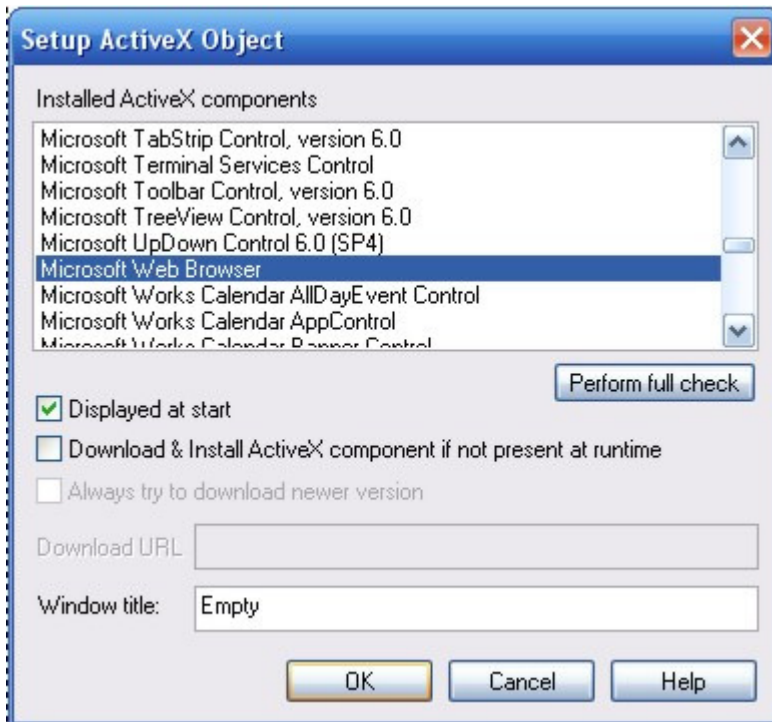
You will now see a dialog box with a progress bar. Although this dialog box may appear strange at first, it's an essential feature. It appears the first time you add an ActiveX object in a MMF session, and for good reason: during development we've seen that many controls which seem to be correct have been badly registered and cannot be used in MMF. So this step checks to see if all the registered controls can be instantiated and created. The result of this is a file located in programs\Extensions and named CCATX.log. This file details whether each control can be instantiated or not.

To speed things up, this first check is only cursory. To make a complete registry test, you should use the "Perform Full Check" button, in setup dialog box. But be warned: this full check can crash your system and application, so you should save all your data before trying it. An object that makes the computer crash is removed from

the list for the next Full Check, therefore, you should run the full check until it stops crashing your computer if you want a clean list of ActiveX objects...

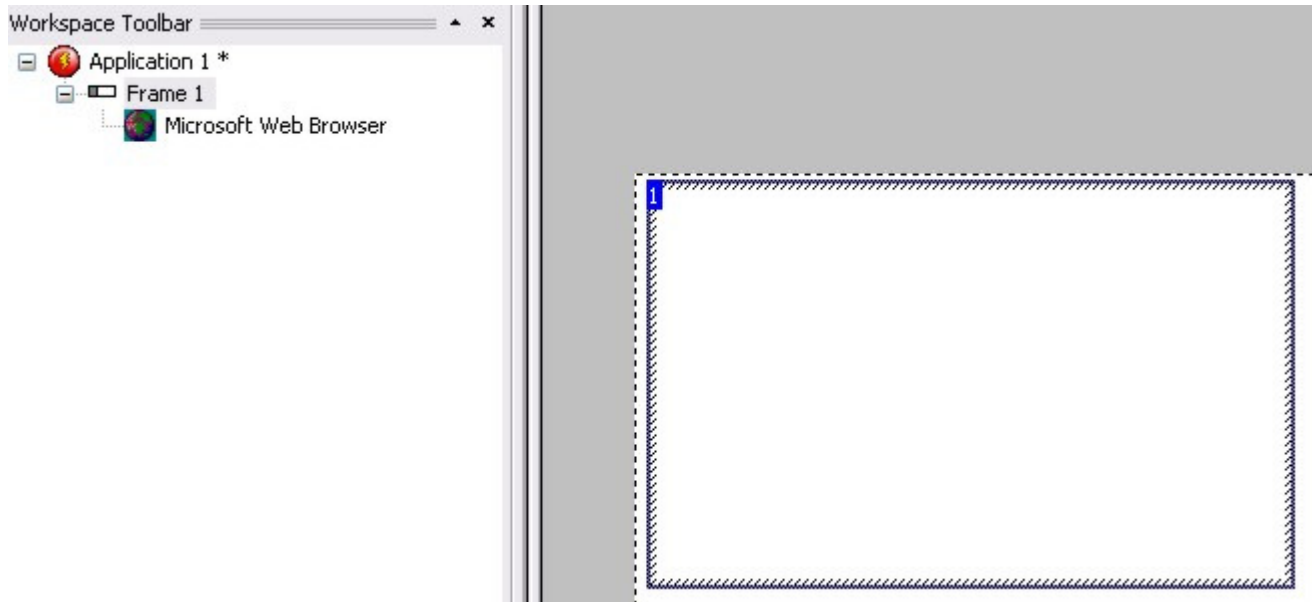
For techies, please note that some controls, though correctly installed, cannot be used by MMF. This can be due to the fact the ActiveX object accesses controls through TypeInfo and some controls don't implement this access mode. In addition, some functionalities could not be made available because they use data types that cannot be used in MMF.

OK, now let's add the object :



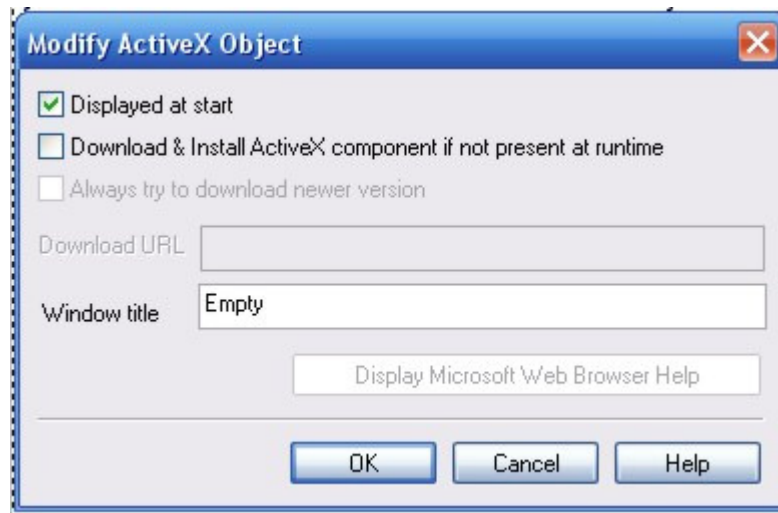
We choose the Microsoft Web Browser which is part of the Windows installation. Other check boxes are useful to make your application capable of running on any computer. As with Internet Explorer, you can ask the object to download the control from a specified URL if it's not installed on the current system. If you check the third checkbox, the system will always try to connect to the URL to check if a newer version is available. If not checked, the current version will be used.

The window title is sometimes useful, even if you will not see it in most cases.



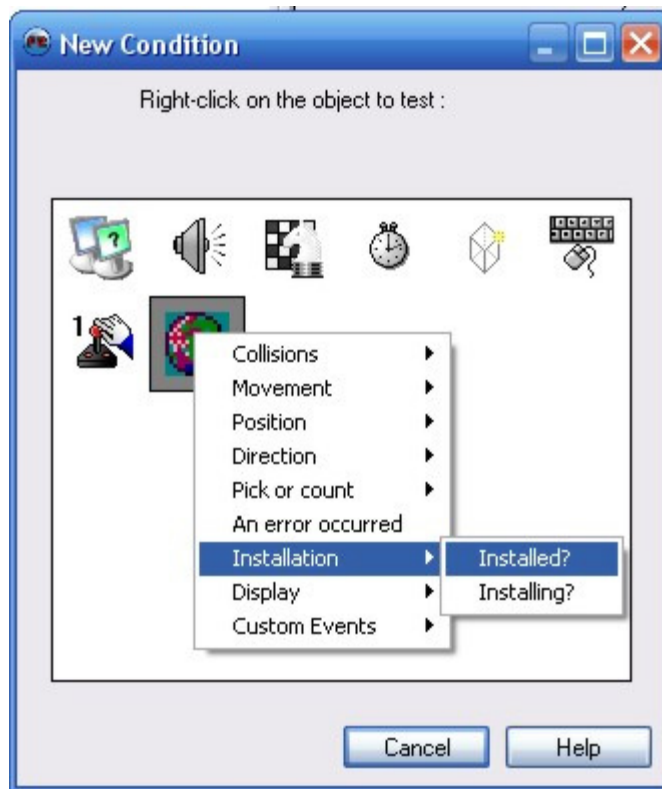
If you click on the OK button, the object will be added, with a default size of 320*200. You can adjust the dimensions to whatever you want. Notice that the icon and name of the object correspond to the control you've chosen.

If you want to change the object's settings, right click on the object and select Edit menu option to see what you can alter. You'll get the following dialog box :



As you can see, there are elements shared with the setup dialog box plus a new button: if the control has a help file available, you will be able to see it. Modify whatever you want, and click on the OK Button.

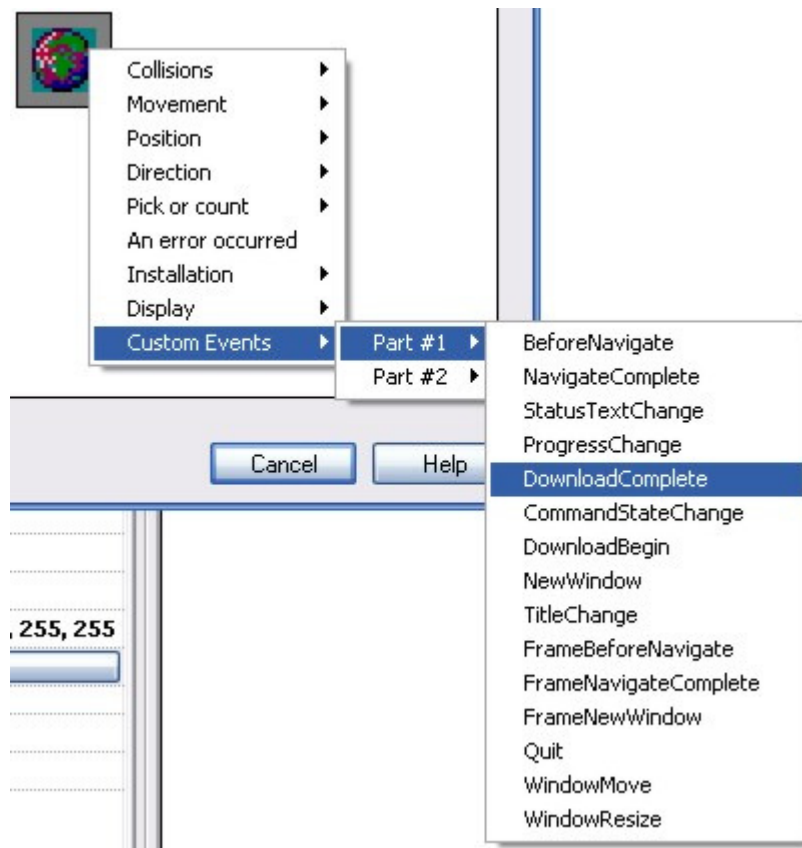
Next, let's go to the event editor. Click on new condition and right click on the object you've created :



You'll see a common event menu. The last three items in the menu are of interest to the ActiveX object.

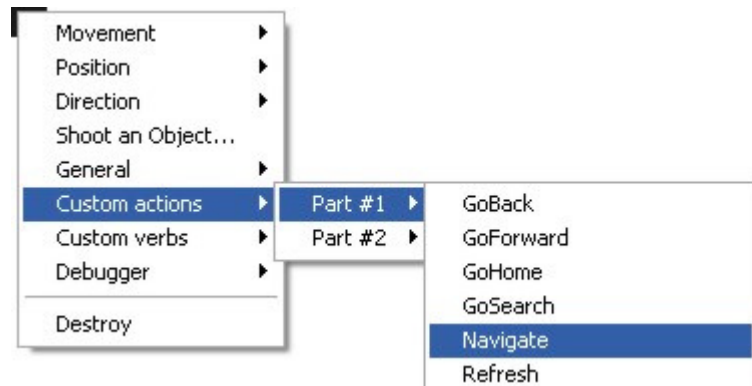
The installation sub-menu is useful if you are not sure the control is installed on the system. Because the installation process is asynchronous, your application can continue to work during installation - so you should check that the control is installed before using it. Display sub-menu is used to show/hide the menu.

The most interesting part is custom events. Custom events are events that change from one control to other. Custom events are grouped by 15 events : there might be several



popups of customs events
for one object.

We will add an event to open the Clickteam website at the start of the frame. Right click on the **New Condition**, right click on the **storyboard controls** icon and select **Start of Frame**. Now under the Web Browser column right click and chooise Custom actions / Part #1 / Navigate.



You may have noticed that there are many commands that can be passed to this Active X component. Active X objects will provide most, if not all, of the commands for that particular component.

Now type in the address of "http://www.clickteam.com". Be sure to include the quotes on both ends.



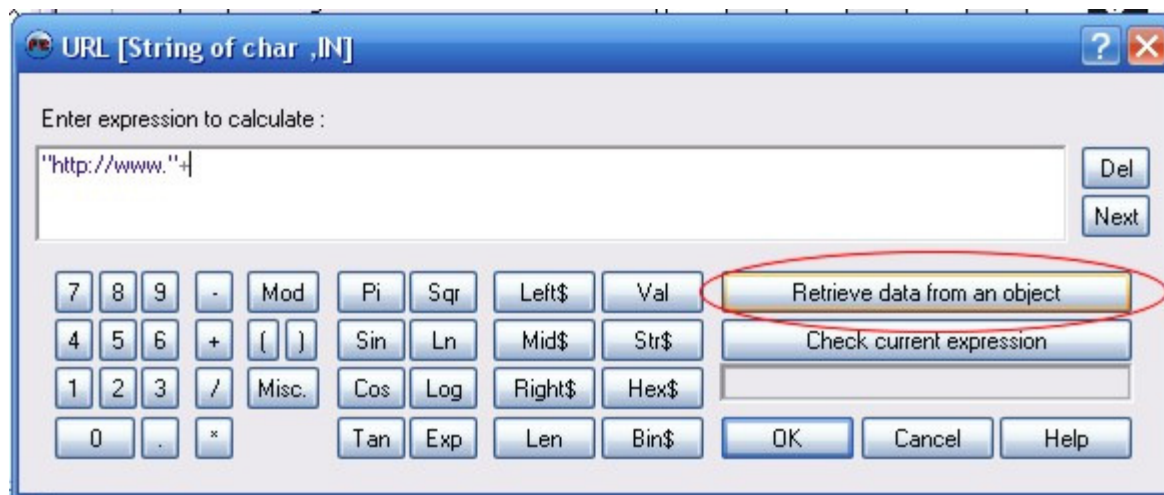
When you run the application you will see the Active X object loads the Clickteam website in a what every size window that YOU defined in the Frame Editor.

Now lets combine some a couple objects to make it more interesting. Return to the Frame Editor and add an Edit Box below the Web Browser Active X object and return

to the Event Editor. Right click on **New Condition**, right click on the **Keyboard / Mouse** icon and select **The keyboard / Upon pressing a key**.

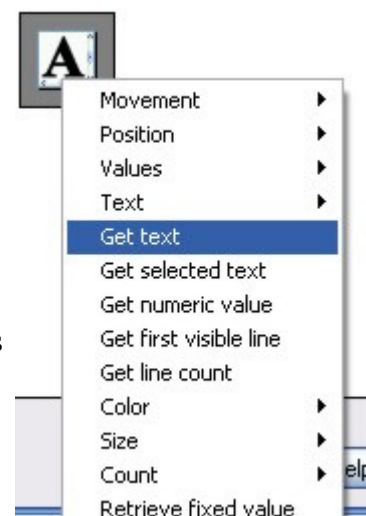


Now press the Enter key to create the condition **Upon pressing "Enter"**. Here is a short cut to programming in Multimedia Fusion 2. Click and hold the checkmark created in our previous event. Now just drag this to our new condition and let go. You now have the same event on our new line. Right click on this new action checkmark and select Edit. To make this even easier just highlight the word "clickteam" and replace this with the following. We will discuss how this works in a moment.



This will be the beginning of our webpage address that we will navigate too.

Now we need the contents from the edit box we added in the Frame Editor. Click on the **Retrieve data from an object** button. (show above) This will display all of the objects contained on this frame. Right click on the Edit Box and select the **"Get text"**.



Finally complete the expression with (+".com") You can use the Check current expression button to see if you have the syntax correct. (shown below)



Hopefully you can see how we added the first text portion, "http://www." to the contents of the Edit Box and then added the ending of, ".com". Now when we run our application we will see the Clickteam website load at the start of the frame and any site name we enter (such as yahoo, google, or vcade to name a few) will be supplied with the proper naming to load the webpage.

☑ Conclusion

The ActiveX object was certainly the most difficult object to program in Fusion but undoubtedly is the most powerful. We had to deal with the ActiveX interface, and it was not easy to create an extension that deals with ... extensions! This explains why such aspects of the Fusion implementation of the ActiveX object are rather complicated. Our hard work however makes adding almost any Active X component that the user has on their computer.

Anyway, you should really dig into it the power offered by ActiveX within Fusion is just amazing...

Copyright © 2006 Clickteam. All rights reserved.
Written by David Screve. Rewrites by Mike Bibby & D.T. Holder.
Please **do not copy or re-use** without written permission.