Some guidelines to using AdPipe in VC++:

Do not change any of the properties of the AdPipe control other than DeveloperID. Other properties, such as AutoStart, Glass, Enabled and AutoSize were held over from the development of the control. Changing any of these can result in unpredictable behavior.

Don't forget to sign up for your DeveloperID at http://www.hoccer.com/adpipe. You can use the default ID (9999) to get an idea of how AdPipe works, but before you distribute your program, you need to sign up for your own ID at our Web page. There is no charge for signing up to use the ID, but it is required so that we can track from what programs clicks are generated. In other words, if you want to make money with AdPipe, we need to distinguish users of your program from users of other programs. This is done with the DeveloperID.

The option that allows VC++ programs to contain ActiveX controls is usually set when the project is created. If this option is not enabled, and you attempt to compile with AdPipe, you will receive some undeclared type errors. If this option was not set when you created your project, you can enable it by going to PROJECT | ADD TO PROJECT | COMPONENTS AND CONTROLS. Then go into DEVELOPER STUDIO COMPONENTS and insert ACTIVEX CONTROL CONTAINMENT COMPONENT..

VC++ AdPipe Sample Application:

In this sample, the AdPipe control has been added to a standard MFC multiple-document program created with AppWizard.

Below are step-by-step instructions describing how to integrate AdPipe with this project. You should be able to use these instructions, with minimal modifications, to integrate AdPipe into any of your Visual C++ applications.

If you haven't registered the control in your system yet, you will need to do so before attempting to add it to a VB project. Do START | RUN and type
"regsvr32 c:\path\adpipe.ocx", where c:\path points to the directory in which the OCX file is located.

- First, add the AdPipe ActiveX control to the project. Do this by selecting PROJECT | ADD TO PROJECT | COMPONENTS AND CONTROLS from the menus. Browse to REGISTERED ACTIVEX CONTROLS and insert AdPipe Control. Press OK to create the default classes associated with the control.

- It is easest to place an ActiveX control directly onto a dialog box. However, since this project uses a multiple-document interface, this cannot be done. Instead, we will insert a dialog bar. Again, navigate the PROJECT menu to into DEVELOPER STUDIO COMPONENTS, and insert a dialog bar. If prompted, be sure to associate it with your main window. Since banner ads are horizontal, do not choose LEFT or RIGHT for the default docking position. In this sample, docking was disabled.

- Now the AdPipe control can be added to the dialog bar from the resource editor. This can be done from the tool bar, or if you right-click on the dialog and do INSERT ACTIVEX CONTROL.

- Align the top left corner of the control with the top left corner of the dialog bar.

- You can now run the program to get an idea of what it will look like. After AdPipe analyzes your system for a few seconds, an image will appear. If you are connected to the Internet, AdPipe will start downloading some sample ads. Although it's not necessary, it is recommended that you let the program run for a minute or so while it downloads the initial ads. You will know it has completed this when the image changes from the default image. Letting the download complete means that from now on AdPipe will immediately show the new ads, rather than the default one. When you are ready, close the program.

- You will probably need to adjust the height of the dialog bar to accommodate the entire image. You do not need to adjust the size of the AdPipe control; it will do so automatically when run.

- Now we will create some code to manage the AdPipe control. Open the VCSAMTYPE menu in the resource editor. Open the view menu, and double click on the blank entry. In the ID box, assuming your dialog bar is named MyDialogBar, select the CG_ID_VIEW_MYDIALOGBAR ID. Set the caption property to "Advertisements." This menu will now demonstrate the ability to disable AdPipe.

This is the end of the VC++ sample. Other things you should consider adding include:

1. code to center the advertisement
2. code to allow the user to move a docking dialog bar containing the ad
3. a Messagebox that displays the first time the program is run (determine this w/ a flag in the registry) and explains why advertisements are in the program (see the VB sample for this).