$ K + ! HIERWIZ Sample

The HIERWIZ sample illustrates the steps necessary to convert an existing project into a Custom AppWizard with one or more wizard pages. In this case, the existing project is the HIERSVR sample, which is one of the MFC\OLE samples. Hierwiz Custom AppWizard has some custom pages that enable the user to 1) add a Text view to the resulting application and 2) specify the source of his/her default data.

The techniques that the HIERWIZ custom AppWizard is designed to demonstrate are:

1. converting an existing project into a Custom AppWizard
2. adding pages to a Project-based AppWizard
3. validating user entries before going from one page to the next
4. changing the state of the Next button depending on user options
5. conditional manipulation of project templates

   - adding a second view to an otherwise single view MDI application
   - adding code to open a default document on startup

Building the Sample

Follow the standard instructions for building the Microsoft Foundation Classes samples. {note: this should probably be changed to reference building a standard AppWizard sample. That will be a little different. The DLL created by building a custom AppWizard sample needs to go in \msvc30\bin\appwiz or equivalent.}

Running the Sample

To run the sample Custom AppWizard, build the sample project and copy the DLL to your MSVC30\BIN\APPWIZ directory. Once in this directory, your custom AppWizard becomes a project type and can be selected from the New Project dialog’s Project Type list.

CONVERTING AN EXISTING PROJECT INTO A CUSTOM APPWIZARD

You can take any existing AppWizard generated project and convert it into a Custom AppWizard by running AppWizard and selecting Custom AppWizard as the project type. On the first page, choose to base your Custom AppWizard on an existing project. The base project must be an AppWizard project.

ADDING PAGES TO A PROJECT-BASED CUSTOM APPWIZARD

A Custom AppWizard created from an existing project will not have any custom pages. You will have to define a CDialogChooser class for the AppWizard, and implement its virtual functions. The easiest way to do this is to create a Custom AppWizard project based on your Custom steps, and copy the generated chooser.* files into your project. You can now modify these files to fit your Custom AppWizard. (For a

_MFCSAMP_CATALOG

$ CATALOG

K CATALOG;Samples

+ main

! CBB(`btn_Contents',"FSC(`_MFCSAMP_CATALOG',`VC20BKS2.HLP')") complete description of the different function and macros in the CDialogChooser class, please refer to the user documentation on Custom AppWizard.)

VALIDATING USER INPUT BEFORE CHANGING PAGES

The CEdtDlg::OnDismiss() function in file "EDITDLG.CPP" performs data validation before the user moves on to the next/previous page. Take a look at this function to see how it works.

CHANGING THE STATE OF THE NEXT BUTTON DEPENDING ON USER OPTIONS.

It may be necessary to disable or enable the "Next" button of your Custom AppWizard depending on the user choices on a given page. Take a look at the CCustom1Dlg::OnNodata() function in file "CSTM1DLG.CPP" for an example.