

This week is going to add a 'File Open' option allowing selection of a file from the disk to be read into the wxTextCtrl. This allows any text file to be loaded into the program for editing.

The existing wxEditor program (PW3) will be modified to include a file dialog according to the following instructions. 3 items need to be added in the header file, and 3 items need to be added in the source code file.

Modifications to the wxEditor project.

Header file: In wxEditorMain.h is the wxEditorFrame class declaration

1. **Add a declaration** for a wxFileDialog pointer in the Private access area just below the wxTextCtrl pointer declaration:

```
wxFileDialog* openFileDialog;
```

2. The existing enum list declares ID items as follows:

```
enum
{
    idMenuQuit = 1000,
    ID_TextBox,
    idMenuAbout
};
```

Add a new ID entry item for a 'File Open' menu item below the ID_TextBox: idMenuOpen,

3. there is a list for menu event processing functions

```
void OnClose(wxCloseEvent& event);
void OnQuit(wxCommandEvent& event);
void OnAbout(wxCommandEvent& event);
```

Add a prototype for an OnOpen command event member function. This should be just like the OnQuit and OnAbout prototypes.

Sources file: In wxEditorMain.cpp are the Frame members of the program.

4. There is an event table which connects event ID's to wxEditorFrame event member functions

```
BEGIN_EVENT_TABLE(wxEditorFrame, wxFrame)
    EVT_CLOSE(wxEditorFrame::OnClose)
    EVT_MENU(idMenuQuit, wxEditorFrame::OnQuit)
    EVT_MENU(idMenuAbout, wxEditorFrame::OnAbout)
END_EVENT_TABLE()
```

Add another EVT_MENU item to connect idMenuOpen to the OnOpen function.

5. The wxEditorFrame constructor code creates a menu bar

```
// create a menu bar
wxMenuBar* mbar = new wxMenuBar();
wxMenu* fileMenu = new wxMenu(_T(""));
fileMenu->Append(idMenuQuit, _("&Quit\tAlt-F4"), _("Quit the application"));
mbar->Append(fileMenu, _("&File"));
```

The first line of code creates the wxMenuBar instance. The second line creates a wxMenu item called fileMenu.

The third line of code Appends an item in the fileMenu to 'Quit' the program. This line connects the menu item with the event identification number - idMenuQuit.

Add another fileMenu item before 'Quit' for a 'File Open' item. Append an item 'Open', with explanatory text 'Open a file' connecting the event identification - idMenuOpen.

```
fileMenu->Append(idMenuOpen, _("&Open\tAlt-F5"), _("Open a file"));
```

6. **Add a member function** which actually create a File Dialog instance. 'File Open' code is added to open/load a selected file.

```
void wxHelloWorldFrame::OnOpen(wxCommandEvent &event)
{
}
}
```

7. Put the following code inside the OnOpen function

1. Create an instance of the wxFileDialog

```
wxFileDialog *openDialog = new wxFileDialog(this, wxT("Choose a file"), wxT(""), wxT(""),
    wxT("Text Files (*.txt)|*.txt|C++ Files (*.cpp)|*.cpp|Header Files (*.h)|*.h"),
    wxFD_OPEN );
```

2. Cause the instance to 'pop-up' (ShowModal)

```
int response = openDialog->ShowModal(); //get response from the dialog
```

3. Check if the response is 'OK' then load the selected file into the wxTextCtrl inside the wxFrame

```
if(response == wxID_OK)
{ //if response ok, then load contents into textControl
    this->textControl->LoadFile(openDialog->GetPath());
}
```

Build and run the finished program. Turn in a screen copy with the file open dialog activated and the Application Frame with the Time and Date in the Status Bar.

see below:

