

QUICK START GUIDE:

This Quickstart is only good if you have already installed the *PennyWise Application Framework* as described in the Installation Guide. If you have not done that yet, do it now.

1—Launch Think Project Manager.

An open dialog will appear so you can select what project you wish to use.

2—Open a PennyWise Application Framework Project Model.

- Press *New*, since we will be creating a New Project.
A dialog will appear asking you what type of project to create.
- Select *PennyWise Application Framework* from the list of project types.
A save dialog will appear.
- Name your project *Hello, World* and save it.
Save it wherever you normally save your projects.

3—First Compile

- Select *Bring Up To Date* from the *Project* Menu.
Think Project Manager will compile the *Hello, World* project.

NOTE: Each time you begin a new project the first three steps will be the same as above. (Of Course, the Name will not be *Hello, World*)

4—Creating *HelloWindow.c*

- Open the file *TEMPLATE_WINDOW.c*
It will appear as an *untitled* window.
- Save this window as *HelloWindow.c*

5—Creating *HelloWindow.h*

- Open the file *TEMPLATE_WINDOW.h*
It will appear as an *untitled* window.
- Save this window as *HelloWindow.h*

NOTE: You can see the top of *HelloWindow.c* has directions on how to use it. The directions in this Quickstart guide are specific to this example, but the directions at the beginning of this file are general for use in any project.

6—Replace all instances of `TEMPLATE_WINDOW` in the files

- Perform a search and replace on the *HelloWindow.c* file. Replace all instances of `TEMPLATE_WINDOW` with `HelloWindow`. (28 Instances)
- Perform a search and replace on the *HelloWindow.h* file. Replace all instances of `TEMPLATE_WINDOW` with `HelloWindow`. (5 Instances)

7—Add the Message

- Find the function `ThisWindowUpdate` in *HelloWindow.c*
- Add the lines

```
MoveTo(40, 70);  
DrawString("\pHello, World");
```

in the function where it says to draw your window here.

8—Add *HelloWindow.c* to the project.

- With *HelloWindow.c* frontmost select *Add 'HelloWindow.c'* from the *Source* menu.

9—Add the *HelloWindow's* ID to the list.

- Open the file *WindowID.h*

- Add the line

```
#define      kWINDOW_ID_HelloWindow  1
```

- Change the line

```
#define      kMAX_WINDOW_IDS  1
```

to read

```
#define      kMAX_WINDOW_IDS  2
```

- Save and Close *WindowID.h*

10—Add the Init code to the Application

- Open the file *InitAppliaction.c*

- Add `#include "HelloWindow.h"` to the list of includes

- Add the line

```
InitHelloWindowHandlers();
```

to the end of the *InitAppliaction* routine.

- Save and Close *InitAppliaction.c*

11—Add the Open Procedure to the Menu Handler

- Open the file *HandleMenus.c*

- Add `#include "HelloWindow.h"` to the list of includes

- Add the line

```
HelloWindowOpen();
```

to the *GlobalMenuDispatch* routine under the line

```
case kFILE_NEW:
```

- Save and Close *HandleMenus.c*

12—Compile and Run!

Congratulations, you've just built your first PennyWise Application Framework project. Looking through the source code for this program you can see that all of the functionality for this window can be added to this source file.

The Demo Project *PennyView* is a good place to look as well. It implements three different WindowIDs. One for Text, Pictures, and Movies. From that example you can see how easily it separates the tasks for each type of window.