

Overview

The OWLGEN code generator generates template C++ code for a Borland C++ OWL windows application from windows resource files. It is a drop-file client (i.e. you drag a file from a drop-file server, like File Manager, and drop it on the application icon). When you run OWLGEN, it will run as an icon on your desktop. Open File Manager and drag a windows dialog or menu file onto the icon.

Building Resources

To start with, you need to create the dialog and menu resource files that your application is going to use. Each of these resources should be kept in a different source file, i.e. one resource per source file. Consult the Borland Resource Workshop guide for saving resources to separate files.

Your Resource Workshop project file should contain nothing more than #includes of header files and other resource files. (In the example, this is *owltest.rc*) It includes the windows header file (*windows.h*) and *rcinc.h*, a file that contains all the resource ids of the resources used in the project. It then includes the main menu resource (*main.mnu*), a dialog resource (*sample.dlg*), and the menu that will appear on the sample dialog (*sample.mnu*).

Each resource is given a #define name. These names are used when generating the C++ variable names and class names. Therefore, you can't just use integers in the Resource Workshop, you need to give all your resources a #defined name. Save your resource ids into *rcinc.h*. This header file is included in all generated .cpp files.

After creating and saving your resources, you are ready to generate an application.

Generating Applications

Start the OWLGEN application. Open up File Manager and drag your resources onto the OWLGEN icon. When the generator is finished, it will tell you the names of the files it created. It should create a [filename].cpp file and a [filename].h file (where [filename] is the filename of the resource file). When you drop a menu file on the generator, it will create the code for a new application with the resource as the menu on the window. When you drop a dialog on the generator, it will create the code for the dialog. If there is a menu file that has the same name as the dialog file, it will also generate a menu for the dialog.

Generating the Sample Application

1. Start OWLGEN.
2. Drag the *main.mnu* file onto the OWLGEN icon. (This will create a *main.cpp* and *main.h* file.)
3. Drag the *sample.dlg* file onto the OWLGEN icon. (This will create a *sample.cpp* and *sample.h* file.)
4. Create a new project using the Borland IDE. Add the *main.cpp* and *sample.cpp* files to the project. Add the *owltest.rc* file.
5. Copy the **GetApplication()->ExecDialog(new TSampleDlg(this, DLG_Sample));** line from the *sample.cpp* file and paste it into the CMTest1 function in the *main.cpp* file. Add **#include "sample.h"** into *main.cpp* after the **#include "main.h"**.
6. Compile and run the application. You need to make sure your project has Borlands includes and libraries in your projects search path. Under the Options menu, the Directories option

displays a dialog box to set the search path for includes and libraries. Make sure the `classlib\include` and `owl\include` are there and the `classlib\lib` and `owl\lib` are in the library. See the Borland documentation for help. There are also sample `prj` files included in an example directory under the `owl` directory. You might want to copy one of these `prj` files and rename it to your own.

Choosing `Test1` from the main menu should invoke the sample dialog. You can edit the `.cpp` files to put in the application specific code.

The `owlapp.h` and `rcinc.h` files are `#included` into all your `.cpp` files. Remember to save all your resource ids into the `rcinc.h` file.

Regenerating

The generator does not keep track of changes you have made to the `cpp` file. i.e. if you modify a dialog and add a new control, when you re-generate the file it will not keep the changes you made to the original `cpp` file. It will create a `[filename].cpo` and a `[filename].ho` file. Your original `[filename].cpp` and `[filename].h` files remain unchanged. You can cut and paste the changes from the `[filename].cpo` and `[filename].ho` file into your original `[filename].cpp` and `[filename].h` files.

Files

The following files should be included:

| | |
|-------------------------|---|
| <code>owlgen.wri</code> | This file. |
| <code>owlgen.exe</code> | The Borland C++ OWL code generator. |
| <code>owlapp.h</code> | Header file included in generated <code>.cpp</code> files to include Borland OWL classes. |

Sample application files:

| | |
|-------------------------|--|
| <code>owltest.rc</code> | Sample application Borland Resource Workshop project file. |
| <code>main.mnu</code> | Sample application main menu resource. |
| <code>sample.dlg</code> | Sample application dialog resource. |
| <code>sample.mnu</code> | Sample application menu resource for sample dialog. |
| <code>rcinc.h</code> | Sample application <code>#defines</code> for resource ids. |

Creating Your Own Applications

After you generate the sample application and run it, examine the code it generated. Notice where you would add your application specific code into the generated code. Refer to your Borland C++ ObjectWindows Users Guide for more information.

Follow these steps when creating your own applications:

Save each resource file you create in Borland's Resource Workshop into its own source file. (We've used the convention of saving dialogs to `.dlg` files and menus to `.mnu` files.)

Name each resource and each control you create in Borland's Resource Workshop with a `#define` and save the `#define` to the `rcinc.h` file.

Drag and drop the main menu resource and each dialog resource onto the OWLGEN icon.

Cut and paste the generated `ExecDialog` statements from each generated dialog `.cpp` file into the appropriate function in its parent `.cpp` file. Add a `#include` for each header generated into your

main .cpp file for each dialog your main menu calls.

Create a project in Borland's IDE and add the .cpp and .rc files to it.

Registering Your Shareware

Toolbox 1.00 Registration Form/Invoice. Please remit to:

DISC

P.O. Box 677805

Orlando, FL 32867

Ordering by check: You can order by sending a check and this order form to the address above.

Please check one: 5.25" Disk ____ 3.5" Disk ____

Toolbox: quantity ____ @ \$ 20.00 ea. = ____

Shipping & handling \$ 3.00 + ____

Florida residents add 7% sales tax + ____

Total payment ____

Payments must be in US dollars drawn on a US bank. Prices guaranteed through 1993.

Name: _____ Date: _____

Company: _____

Address: _____

City, State, Zip: _____ Country: _____

Day Phone: _____ Eve: _____

Electronic Mail address: _____

Where did you get your copy of Toolbox? _____

Comments:

