Introduction to GMF and GUIL 1

This chapter provides an overview of the GIL-to-Motif (`gmf`) and GIL-to UIL (`guil`) utilities (GIL is the Guide Interface Language output of Devguide).

Devguide, GMF, and GUIL

Devguide is a development tool that enables you to create and test user interfaces without writing any code.

Devguide produces GIL files that you can convert into Motif C code with the utility, `gmf`, or into UIL with the utility, `guil`.

Names and Terminology

The word "Guide" in the full name, "OpenWindows Developer's Guide," is an acronym for Graphical User Interface Design Editor. The abbreviated name, "Devguide," refers to the graphical interface tool you use to develop the user interface.

The term *gmf* refers to the utility that takes the GIL file produced by Devguide to produce Motif source code for your user interface. This code is referred to as the *UI* (user interface) code, which contrasts with the *application code*.

1

Installation and Environment Setting for Devguide, GMF, and GUIL

To install Devguide and the `gmf` and `guil` utilities, follow the instructions in the *Software Developer Kit Installation Guide*. See the *OpenWindows Developer's Guide: User's Guide* for a more complete discussion of the process of setting environment variables.

Setting Environment Variables for GUIDEUTILHOME

By default, the `SUNWgmfu` package is installed in the directory, `/opt/SUNWgmfu`. Wherever `SUNWgmfu` is installed, the GUIDEUTILHOME directory should be set to the directory where the package is stored. For example, if the package were installed in `/export/home/SUNWgmfu`, then:

for `C` shell:

% **`setenv GUIDEUTILHOME /export/home/SUNWgmfu`**

for Bourne shell:

$ **`GUIDEUTILHOME=/export/home/SUNWgmfu`**
$ **`export GUIDEUTILHOME`**

If you want to access `gmf` and `guil` from anywhere, `GUIDEUTILHOME` needs to be set and the `PATH` environment variable needs to have `GUIDEUTILHOME/bin` in it.

For access to the `gmf` and `guil` man pages, `MANPATH` needs to have `GUIDEUTILHOME/man` in it.

Setting Environment Variables for MOTIFHOME

In addition to setting the variable for GUIDEUTILHOME, you should set MOTIFHOME to the location where Motif is installed (`/usr/opt/SUNWmotif`, for example).

If you are using a non-Sun version of Motif, you may have to edit the Makefile to change `-I$MOTIFHOME/include` to `-I$MOTIFHOME/usr/include` and to change `-L$MOTIFHOME/lib` to `-L$MOTIFHOME/usr/lib`.

How you interact with Devguide and GMF

The basic interaction with Devguide is very simple. When you use Devguide to develop your user interface with the OPEN LOOK look and feel by dragging and positioning glyphs on interface windows, Devguide saves a description of the interface in a GIL file. This process is explained in detail in the *OpenWindows Developer's Guide: User's Guide.*

Note – To see the interface with the Motif look and feel, you have to compile the generated code.

To create Motif C code from a GIL file, run the `gmf` utility on the GIL file, shown below as `<fn>.G`. The `.G` suffix is the standard suffix for GIL files. `gmf` produces the four files shown in Figure 1-1.

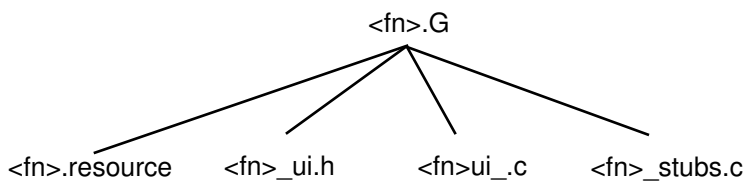

Figure 1-1 GMF-Generated Source Files

`gmf` also automatically generates a program Makefile.

Devguide allows you to organize several GIL files into projects. Devguide saves projects in files that have a `.P` extension. When you run `gmf` on a project file, it generates the files shown in Figure 1-1 for each `.G` file in the project. It also generates `<projectname>.c` and `<projectname>.h` files.

Note – `gmf`-generated code is identical to handwritten Motif code; you can use `gmf`'s output to learn Motif.

1