Motif Environment 2

This chapter contains information for developers writing CDE Motif applications for either the OpenWindows or Solaris CDE environment.

Motif Packaging in the Solaris 2.5 and Solaris CDE 1.0 Software	*7*
Compiling and Linking Motif Programs	*8*
Shared Library Policy	*9*
Future Motif Compatibility	*9*

Motif Packaging in the Solaris 2.5 and Solaris CDE 1.0 Software

Motif support in the Solaris 2.5 software is packaged as it was in the Solaris 2.4 software. This packaging provides run-time support in the run-time product and developer support in the developer's environment.

Appendix A, "CDE Motif Packaging," describes the packaging scheme and briefly describes the contents of all the Motif packages in the Solaris CDE 1.0 release.

The Motif run-time support in the Solaris 2.5 and Solaris CDE 1.0 software includes the following:

- Shared libraries
- Header files
- Key bindings

2

The Motif developer support in the Solaris 2.5 and Solaris CDE 1.0 software includes the following:

- Debug libraries
- `uil` compiler
- Man pages
- Demos and sample source

The Motif Window Manager (`mwm`) is available in the Solaris 2.5 software, but not in the Solaris CDE 1.0 software. See Appendix C, "Solaris 2.5 Motif Window Manager," for information on `mwm`.

The Motif header files required for application development are located in `/usr/dt/include`. The Motif libraries are located in `/usr/dt/lib`.

Note – Motif applications built on the Solaris 2.4 or Solaris 2.5 version (`libXm.so.3`) of the Motif run-time package will not work when used with the Motif shared libraries delivered in the Solaris 2.3 version (`libXm.so.2`).

Compiling and Linking Motif Programs

When you compile Motif programs, include the following compiler syntax to enable the compiler to find the Motif and X Window System™ header files:

```
-I/usr/dt/include -I/usr/openwin/include
```

Use the following compiler syntax to direct the linker to the correct shared libraries as shown in the following:

```
-R/usr/dt/lib -R/usr/openwin/lib -L/usr/dt/lib -L/usr/openwin/lib
```

The following is an example of a compile-and-link line for a Motif application:

```
cc -o myprog -I/usr/dt/include -I/usr/openwin/include myprog.c -R/usr/dt/lib \
-R/usr/openwin/lib -L/usr/dt/lib -lXm -L/usr/openwin/lib -lXt -lX11
```

Shared Library Policy

SunSoft™ will increment the major version number of each shared Motif library whenever there are binary-incompatible differences from the previous release. SunSoft will make available (either on the Motif distribution or through some other channel) all prior versions of each library. This will ensure that your applications linked with a particular release can continue to run, even after a new Motif release has been installed.

Future Motif Compatibility

Developers who used Motif 1.2 to subclass widgets need to be aware of potential binary compatibility problems with future releases. Internal Motif widget data structures may be changed, breaking any subclass that relies on the position of fields in these data structures.

Motif provides a mechanism that developers can use to avoid such problems. Refer to the *Motif Programmer's Reference* manual. A demonstration of this mechanism is provided in the `/usr/dt/examples/motif/dogs` directory.

2