

# Using imake with OpenWindows

Paul DuBois  
dubois@primate.wisc.edu

Wisconsin Regional Primate Research Center  
Revision date: 1 May 1996

Sun supplies versions of *imake*, *xmkmf*, and the *imake* configuration files on its systems that have been modified from standard X11 versions. Presumably the aim of these modifications is to make *imake* easier to use under Solaris and OpenWindows, but the modified tools have been the cause of some difficulty over the years. This document describes how to overcome these difficulties. Its aims are two:

- To make it easier to use Sun's *imake/xmkmf* setup with the configuration files that Sun supplies.
- To make it easier to use Sun's *imake/xmkmf* setup with other configuration files, such as the standard X11 files.

I believe the information presented here is correct, but at the moment it's being presented on a "field test ready" basis. Feedback will improve its usefulness. If you try the instructions, please let me know the results (whether or not the instructions worked for you). If you find errors, please let me know. Suggestions and comments are also welcome. I can be reached at [dubois@primate.wisc.edu](mailto:dubois@primate.wisc.edu).

## Version Numbers

The table below shows some of the (for our purposes) more important versions of Solaris, SunOS, and OpenWindows, and how they correspond. At OpenWindows 2.0, *imake* support is X11R4-based. Beginning with OpenWindows 3.3, *imake* support is X11R5-based (with some R6 modifications), with the exception that *xmkmf* is based on the OW 2.0 version of *xmkmf*. Beginning with OpenWindows 3.4, Motif header files and libraries are included for Common Desktop Environment (CDE) support.

Solaris	SunOS	OpenWindows	Comment
1.0	4.1.1B	2.0	<i>imake</i> tools are X11R4-based
1.1	4.1.3	3.0	
2.0	5.0	3.0.1	SunOS becomes SVR4-based
2.3	5.3	3.3	<i>imake</i> tools become X11R5-based (+X11R6 mods)
2.4	5.4	3.4	Motif header files and libraries now included
2.5	5.5	3.5	Current version

Sun's *imake* tools are installed under `/usr/openwin`, so I'll refer to them in terms of the OpenWindows version in which they first appeared. For example, "OW 2.0 *imake*" means the *imake* included with OpenWindows versions 2.0 through 3.2, whereas "OW 3.3 *imake*" means the *imake* included with OpenWindows versions 3.3 and up. If you don't know what version of OpenWindows you have, run `uname -r` to find out your SunOS version number, then see the table for the corresponding OpenWindows version.

## Assumptions

In this document, I make the following assumptions about directory and file locations:

- The OPENWINHOME environment variable has a value of */usr/openwin*. (If the value is different on your system, substitute that value whenever you see */usr/openwin* below.)
- *imake* and *xmkmf* are located in */usr/openwin/bin* (and also in */usr/openwin/bin/xview*, which is a sym-link to */usr/openwin/bin*). I assume that your search path is set up so that you get the OpenWindows versions of *imake* or *xmkmf*, not versions installed somewhere else such as */usr/bin/X11* or */usr/X11R6.1/bin*.
- The configuration files are located in */usr/openwin/lib/config*.

## Problems

Briefly, you can expect severe problems if you have the OW 2.0 tools (i.e., OpenWindows earlier than 3.3):

- Use of the environment variable OPENWINHOME by *imake* may cause configuration files not to be found.
- *xmkmf* passes incorrect arguments to *imake*.
- The configuration files don't know where any of the OpenWindows stuff is.
- The OW 2.0 tools don't handle XCOMM or NullParameter.

Sun straightened out most of these problems in later releases. Therefore, you can expect fewer difficulties if you have the OW 3.3 tools (i.e., OpenWindows 3.3 or later), which behave much more reasonably. The main problems are these:

- *xmkmf* (still) passes incorrect arguments to *imake* (although in a different way).
- The wrong *install* command can be selected.
- The configuration files don't know how to find any CDE stuff.

There are other miscellaneous issues as well, such as that no version of OW *xmkmf* knows about the *-a* option that X11 *xmkmf* understands from R5 on.

The causes of these problems and how to deal with them are described in the sections following. Some of the solutions involve replacing Sun's tools. For instance, each Sun version of *xmkmf* is broken in some way, although the particular problems are different for each version. For purposes of illustration, I show what the problems are and how to edit *xmkmf* to fix them. But I recommend instead that you just grab an already-modified version that fixes the problems and also adds *-a* option support. See the section **Obtaining Replacement Tools** for instructions on how to get these alternate versions.

I advise you to make a backup copy of any file you modify or program you replace, in case you make a mistake or want to undo your actions. That also allows you to *diff* the original file with your modified copy later to see what you did.

## Fixing OW 2.0 imake Support

This section describes what you need to do to make the OW 2.0 tools usable.

**Problem:** Use of the environment variable OPENWINHOME by OW 2.0 *imake* may cause configuration files not to be found.

**Description:** Standard X11 *imake* looks for *-Ipathname* in its argument list to find out the pathname of the directory in which the configuration files are located. OpenWindows *imake* is nonstandard because it also uses OPENWINHOME to locate configuration files. The 2.0 and 3.3 versions of OW *imake* differ somewhat, though.

OW 3.3 *imake* looks in the *lib/config* directory under OPENWINHOME if that environment variable is set and no *-I* argument is given. OPENWINHOME typically has a value of */usr/openwin*, so the effect is that *imake* defaults to looking in */usr/openwin/lib/config* for configuration files when no *-I* argument is given. This makes it easier to invoke *imake* if you're using Sun's configuration files. Since you can override the use of OPENWINHOME by specifying *-I* on the command line to specify any set of configuration files you want, OW 3.3 *imake* presents no special problems.

However, OW 2.0 *imake* is less cooperative. It insists on looking under OPENWINHOME if that variable is set. It does this even if you specify an *-I* argument on the command line. This slavish dependence on OPENWINHOME makes *-I* useless for indicating where the configuration files are. If you only want to use the OpenWindows configuration files, that may not be a problem. But if you want to use any other files, you can't easily do so. Any *imake* that makes this difficult must be considered broken.

If you don't know which version of *imake* you have, make sure OPENWINHOME is set, then run the following command:

```
% imake -v -s/dev/null -T/dev/null -f/dev/null -I/abc
```

This command tells *imake* to display the *cpp* command it uses to generate Makefiles. (You can safely run this *imake* command anywhere, because the */dev/null* arguments keep it from actually creating any files.) If you see only *-I/abc* in the *cpp* command, you're okay. If you see *-I/usr/openwin/lib/config* preceding *-I/abc*, your *imake* is broken.

There are two workarounds if you have a broken *imake*, although both are problematic:

- If you unset OPENWINHOME, *imake* uses command-line *-I* arguments in the normal way. This is a poor solution because it breaks any other programs that assume OPENWINHOME has the correct value. One of these is the OW 2.0 *xmkmf*, so this is a serious problem.
- If you set the environment variable *IMAKEINCLUDE* to an *-I* argument that specifies where to look for configuration files, *IMAKEINCLUDE* takes precedence even over OPENWINHOME. This is a poor solution if you routinely use different sets of configuration files because you have to change the value every time you want to specify a different set.

A better solution to the problem is to leave OPENWINHOME set but replace Sun's *imake* with the one from X11R6.1. However, note that replacing OW 2.0 *imake* is not sufficient; you also need to fix *xmkmf* and the configuration files, as described below.

**Problem:** *xmkmf* passes incorrect arguments to *imake*.

**Description:** OW 2.0 *xmkmf* determines which arguments to pass to *imake* in a section that looks like this:

```
if [ -n "$stopdir" ]; then
    args="-I$stopdir/config -DTOPDIR=$stopdir -DCURDIR=$curdir"
elif [ -n "$OPENWINHOME" ]; then
    args="-DUseInstalled "$OPENWINHOME/lib/config
else
    args="-DUseInstalled "/usr/lib/X11/config
fi
```

The first case handles using configuration files located within the X11 source tree, and does not concern us here. The second and third cases determine whether to use installed OpenWindows or X11 configuration files. If OPENWINHOME is set, *xmkmf* tells *imake* to use the OpenWindows files. Otherwise *xmkmf* specifies the standard X11 files in */usr/lib/X11/config*. *-DUseInstalled* is passed in both cases to indicate use of installed configuration files. (This is important so that a subsequent *make Makefile* command continues to use the same installed files as those that were used to build the *Makefile* in the first place.)

The error in the *xmkmf* fragment just shown is that neither of the last two cases has *-I* before the configuration directory pathname. This causes *imake* not to interpret them as locations in which to look for configuration files. To fix this, add *-I* before the pathnames:

```

if [ -n "$stopdir" ]; then
  args="-I$stopdir/config -DTOPDIR=$stopdir -DCURDIR=$curdir"
elif [ -n "$OPENWINHOME" ]; then
  args="-DUseInstalled "-I$OPENWINHOME/lib/config
else
  args="-DUseInstalled "-I/usr/lib/X11/config
fi

```

**Problem:** The OW 2.0 configuration files don't know where any of the OpenWindows stuff is.

**Description:** OW 2.0 OpenWindows configuration files are essentially identical to the corresponding X11R4 files. In fact, they are so close that the parameters indicating where to find things like X11 header files and libraries have the same values as in the X11R4 files. That's a problem, because OpenWindows locates those files under the */usr/openwin* hierarchy. Consequently, any *Makefile* built using the OW 2.0 configuration files won't be able to find any OpenWindows stuff, and therefore probably won't build applications successfully.

To fix this problem, add the following lines to *site.def* to tell the configuration files about the */usr/openwin* hierarchy:

```

#ifdef OpenWinHome
#define OpenWinHome /usr/openwin
#endif

OPENWINHOME = OpenWinHome

#ifdef BinDir
#define BinDir $(OPENWINHOME)/bin
#endif

#ifdef LibDir
#define LibDir $(OPENWINHOME)/lib
#endif

#ifdef IncRoot
#define IncRoot $(OPENWINHOME)/include
#endif

#ifdef StandardIncludes
#define StandardIncludes -I$(INCRROOT)
#endif

#ifdef ExtraLoadFlags
#define ExtraLoadFlags -L$(OPENWINHOME)/lib
#endif

```

The OW 3.3 configuration files don't have this problem (as long as OPENWINHOME is set correctly in your environment), because *site.def* contains the following line:

```
#define ProjectRoot $(OPENWINHOME)
```

This line equates *ProjectRoot* to the value of OPENWINHOME. Location parameters for OpenWindows-related stuff are anchored to *ProjectRoot*, so they get the correct values.

The next two problems are not Sun-specific; they occur with any X11R4-based configuration files if you're building Makefiles from Imakefiles that were written assuming the use of configuration files based on R5 or later. Since OW 2.0 *imake* tools are R4-based, they're subject to these two problems.

**Problem:** XCOMM is not handled correctly by the OW 2.0 *imake* tools.

**Description:** As of X11R5, XCOMM is used for writing comments that should appear in a *Makefile*. For example:

```
XCOMM this is a comment
```

When the above line is written in an *Imakefile*, XCOMM is supposed to be translated to #, resulting in a line in the *Makefile* that looks like this:

```
# this is a comment
```

However, the OW 2.0 *imake* tools are X11R4-based and don't know about XCOMM, resulting in literal instances of XCOMM in your *Makefiles*.

The best way to address the problem is to replace *imake* with the current version from X11R6.1. R6.1 *imake* handles XCOMM internally. Another strategy that often works (and may be easier) is to add the following lines to the top of *Imake.tmpl*:

```
#ifndef XCOMM
#define XCOMM #
#endif
```

There is a widely-circulated patch to *xmkmf* that adds `-DXCOMM=/**/#` to the arguments passed to *imake*. That's only half a solution: it causes XCOMM to be processed when you run *xmkmf*, but not if you run *make Makefile* later.

**Problem:** `NullParameter` is not handled correctly by the OW 2.0 *imake* tools.

**Description:** `NullParameter` originally appeared in X11R5, defined as the empty token. It's used in rule invocations to indicate explicitly that an argument is empty. However, the OW 2.0 *imake* tools are X11R4-based and don't know about `NullParameter`, resulting in literal instances of `NullParameter` in your *Makefiles*. To fix the problem, add the following line to your *Imake.rules* file:

```
#define NullParameter
```

### Fixing OW 3.3 *imake* Support

The OW 3.3 tools are pretty usable as supplied by Sun. This section describes some things you can do to make them more usable.

**Problem:** *xmkmf* passes incorrect arguments to *imake*.

**Description:** OW 3.3 *xmkmf* determines which arguments to pass to *imake* in a section that looks like this:

```
if [ -n "$stopdir" ]; then
  args="-I$stopdir/config -DTOPDIR=$stopdir -DCURDIR=$curdir"
elif [ -n "$OPENWINHOME" ]; then
  args=" -I$OPENWINHOME/lib/config"
else
  args=" -I/usr/lib/X11/config"
fi
```

The first case handles using configuration files located within the X11 source tree, and does not concern us here. The second and third cases determine whether to use installed OpenWindows or X11 configuration files. If OPENWINHOME is set, *xmkmf* tells *imake* to use the OpenWindows files. Otherwise *xmkmf* specifies the standard X11 files in `/usr/lib/X11/config`. There are no missing `-I`'s in the second and third cases (as there are with the OW 2.0 *xmkmf*), but notice that `-DUseInstalled` has been deleted. My guess is that Sun intends this not to matter for the OpenWindows case, because Sun defines `UseInstalled` in *site.def* to force it on. However, the definition is incorrect; see discussion below. At any rate, it's important not to delete `-DUseInstalled` for the X11 case because the X11 files expect it to be passed on the command line. If *xmkmf* doesn't specify that argument, any subsequent *make Makefile* command won't find the installed X11 files. To fix the problem, change the fragment just shown to restore the `-DUseInstalled` arguments:

```
if [ -n "$stopdir" ]; then
```

```

    args="-I$topdir/config -DTOPDIR=$topdir -DCURDIR=$curdir"
elif [ -n "$OPENWINHOME" ]; then
    args="-DUseInstalled "-I$OPENWINHOME/lib/config
else
    args="-DUseInstalled "-I/usr/lib/X11/config
fi

```

In addition to modifying OW 3.3 *xmkmf*, you should fix OW 3.3 *site.def*, which defines `UseInstalled` incorrectly. The relevant line looks like this:

```
#define UseInstalled YES
```

There are two problems here:

- First, the definition is misleading. As written, the definition seems to imply that if you changed the value to NO, it would make a difference. But it wouldn't: the configuration files never test the value of `UseInstalled`, they only test whether or not it's defined. `UseInstalled` should not be defined as YES or NO; it should simply be defined as nothing or left undefined.
- Second, defining `UseInstalled` unconditionally (as is done above) may result in "macro redefinition" errors if *imake* happens to be invoked with `-DUseInstalled` on the command line (as it will be if you run *make Makefile* later).

You could get around these problems by rewriting the definition as follows:

```

#ifndef UseInstalled
#define UseInstalled
#endif

```

But you're better off to remove the definition from *site.def* entirely. If you've made the change to *xmkmf* shown above, *xmkmf* defines `UseInstalled` for you anyway.

**Problem:** `InstallCmd` in the OW 3.3 configuration files may select the wrong *install* command.

**Description:** The value of `InstallCmd` is simply `install`. Depending on how your search path is set up, you may get either the System V-based `/usr/sbin/install` or the BSD-based `/usr/ucb/install`. The install rules in the configuration files expect a BSD version, so change the value of `InstallCmd` in *sun.cf* to explicitly select the proper one as follows:

```
#define InstallCmd /usr/ucb/install
```

**Problem:** The OW 3.3 configuration files don't know how to find any CDE stuff.

**Description:** Those systems on which Sun ships CDE Motif include Motif libraries and header files, but the configuration files do not include any CDE support. If you wish to develop applications under CDE Motif, you must modify your OpenWindows configuration files. For a set of patches to the files that make the appropriate changes, see the section **Obtaining Replacement Tools**.

### Miscellaneous Issues

This section describes what to do about various other issues that don't fall into the categories already discussed.

**Problem:** *xmkmf* (all OW versions) doesn't understand the `-a` option.

**Description:** Beginning with X11R5, standard X11 *xmkmf* takes a `-a` argument that tells it to run the following commands after generating the *Makefile*:

```

make Makefiles
make includes
make depend

```

Sun's *xmkmf* was initially based on X11R4 *xmkmf*, and subsequent versions have never been updated to provide this functionality. See **Obtaining Replacement Tools** to obtain a version that understands *-a*.

**Problem:** The OW 3.3 configuration files don't support *gcc*.

**Description:** Sun stopped including a C compiler with their systems as of SunOS 5.x. It's possible to get *gcc* for free, but the configuration files shipped with OW 3.3 aren't set up to use it very well. See the section **Obtaining Replacement Tools** for some notes on changes you can make to your configuration files to add *gcc* support.

### Obtaining Replacement Tools

Replacement or auxiliary tools that make it easier to use *imake* under OpenWindows are available at either of these locations:

```
http://www.imate.wisc.edu/software/imate-stuff
ftp://ftp.imate.wisc.edu/software/imate-stuff
```

The tools are provided in the form of a *tar* file that has been *gzip*'ed (*openwin-support.tar.gz*) or *compress*'ed (*openwin-support.tar.Z*). Transfer the one you want and run *gunzip* (or *uncompress*) to recover *openwin-support.tar*. Then extract the files:

```
% tar xf openwin-support.tar
```

You may need to use the following command instead on System V systems:

```
% tar xof openwin-support.tar
```

The *tar* command unpacks the distribution into a directory *openwin-support* that contains the following:

- A version of *xmkmf* that doesn't have the problems of the OW 2.0 and 3.3 *xmkmf*, and that understands the *-a* option.
- Binaries of the X11R6.1 version of *imake* and *makedepend* that run under BSD-based SunOS 4.x or System V-based SunOS 5.x.
- Configuration file modifications. These include an addendum for OW 2.0 *site.def* to let the 2.0 configuration files know about the OpenWindows hierarchy, a set of patches for the OW 3.3 configuration files to add CDE support, and notes on using the OW 3.3 configuration files with *gcc*.
- Installation instructions.

### Acknowledgements

The following people provided assistance in determining how various versions of OpenWindows behave or supplied examples of what they did to make *imake* work under OpenWindows: Jayachander Balakrishna, Philip Brown, John Evans, Bob Friesenhahn, Charlie Havener, James McIninch, Howard Moflich, Monty Solomon, Adam Stein, and Larry Virden.

The table of system version numbers was derived from a more extensive table in Casper Dik's Solaris FAQ, which is available in plain text or HTML forms at:

```
ftp://ftp.fwi.uva.nl/pub/solaris/solaris2.faq
http://www.fwi.uva.nl/pub/solaris/solaris2.html
```