

# 1 Machines

Below is a list of machines that people have attempted to compile GNUstep on. GNUstep compiles with little or no trouble on many of the more popular operating systems. Some machines marked with *Unstable* may have some trouble or may not work at all. Platforms marked *Needs Testing* are not actively tested by developers and need someone to help with reporting problems and fixes. Platforms marked *Obsolete* are very old distributions. No one really knows if GNUstep works on these although they may.

A recommended compiler is listed for each machine, if known. You should try to use the recommended compiler for compiling GNUstep, as GNUstep is quite complex and tends to provoke a lot of errors in some compilers. Even versions newer than the listed compiler may not work, so don't just get the latest version of a compiler expecting it to be better than previous versions.

Compiler notes: If a recommended compiler is not listed, take note of the following information before choosing the compiler you use.

'egcs or gcc < 2.95'

Might work, but few people use these now.

'gcc 2.95.x'

Probably the oldest compiler that GNUstep is regularly tested with.

'gcc 2.96' Not an official gcc release. Some versions (Redhat, Mandrake) have problems that prevent GNUstep from being compiled correctly and cause mysterious errors.

'gcc 3.0.x'

A fairly good compiler. Recommended.

'gcc 3.1' Several bugs were introduced in the version. It's probably better to avoid this one, although it might work fine.

'gcc 3.2.x'

Pretty good. Recommended.

'gcc 3.3.x'

Recommended. Fixes some bugs relating to protocols as well as other improvements.

'gcc 3.4' No info yet.

If you have compiled GNUstep on a specific machine, please send information about what you needed and any special instructions needed to GNUstep [bug-gnustep@gnu.org](mailto:bug-gnustep@gnu.org).

If you're having mysterious trouble with a machine, try compiling GNUstep without optimization. Particularly in the newer GCC compilers, optimization can break some code. The easiest way to do this is when configuring, 'CFLAGS="" ./configure'. Or when building, 'make OPTFLAG=""'.

Also if you manually upgraded gcc and/or make, we recommend reading the documentation at <http://www.LinuxFromScratch.org> for tips on compiling and installing gcc and make. If you had GNUstep previously installed, make sure you completely remove all of it, including installed init scripts.

## 1.1 Darwin/ix86

Currently tested on Darwin 7.x

### ‘Recommended compiler’

gcc 3.3.2. Older versions will not compile on Darwin.

Default compiler has unknown problems. Use the GNU runtime. Download the gcc compiler and configure it with `-enable-threads=posix`. You don’t need binutils or anything else.

### ‘Extra libs needed’

### ‘Special Instructions’

Read the [README.Darwin](#) file in the `gnustep-make/Documentation` directory for complete instructions.

## 1.2 Darwin/PowerPC

Currently tested on Darwin 6.x, 7.x

### ‘Recommended compiler’

gcc 3.3.2. Older versions will not compile on Darwin.

Default compiler has unknown problems. Use the GNU runtime. Download the gcc compiler and configure it with `-enable-threads=posix`. You don’t need binutils or anything else.

### ‘Extra libs needed’

Use `libffi` (not `ffcall`). This should be enabled by default in `gnustep-base` so you don’t have to type `-enable-libffi`. For 6.x, you need the `dlcompat` library (from [www.opendarwin.org](http://www.opendarwin.org)) to load bundles (not needed for 7.x).

### ‘Special Instructions’

Read the [README.Darwin](#) file in the `gnustep-make/Documentation` directory for complete instructions.

See also the MacOSX/PowerPC section

## 1.3 Debian/DEC-Alpha

### ‘Recommended compiler’

Unknown

### ‘Extra libs needed’

Unknown

### ‘Special Instructions’

None

## 1.4 Debian/i386

Tested on sid.

### ‘Recommended compiler’

Unknown

'Extra libs needed'  
Unknown

'Special Instructions'  
None

## 1.5 Debian/PowerPC

Tested on sid.

'Recommended compiler'  
Unknown

'Extra libs needed'  
Unknown

'Special Instructions'  
None

## 1.6 Debian/SPARC

Tested on sid.

'Recommended compiler'  
Unknown

'Extra libs needed'  
Unknown

'Special Instructions'  
None

## 1.7 FreeBSD 5.x

Tested on 5.0 and 5.1

'Recommended compiler'

'Extra libs needed'  
None.

'Special Instructions'  
Can install via /usr/ports/devel/gnustep, but not all required dependancies are installed. See the GNUstep-HOWTO for list of libraries.

## 1.8 FreeBSD 4.x

'Recommended compiler'

'Extra libs needed'  
Unknown

'Special Instructions'  
For gcc 3.0.4, make WANT\_THREADS\_SUPPORT=YES  
For libxml2 2.4.24, make WITHOUT\_PYTHON=YES

## 1.9 FreeBSD 3.x

Compiles "out of the box" on FreeBSD 3.4.

'Recommended compiler'

gcc 2.95.2

'Extra libs needed'

Unknown

'Special Instructions'

You need to use gmake not make to compile the GNUstep packages. A special port of gdb can be used with the Objective-C patches from <ftp://ftp.pcnet.com/users/eischen/FreeBSD/gdb-4.17-port.tar.gz>

The best compiler for GNUstep is the latest release of the GNU Compiler Collection (GCC). You can find it at <http://egcs.cygnum.com/>.

If you want to use the native POSIX threads support from 'libc\_r' pass `--enable-threads=posix` to configure. This is the recommended option as this is the FreeBSD threads package that gives the best results –with others you may be unable to run some examples like 'diningPhilosophers'.

The whole compilation process can fail if you have another threads library installed so watch out for installed packages like 'pth' and such. Besides the support for libc\_r, GNUstep will also look for 'pth' and 'pthreads', so if you have installed them and they aren't detected prepare to write a nice bug report.

This can be done more much easily by using the port version. Just cd to '/usr/ports/lang/egcs' and do a "make WANT\_THREADS=yes install". Easy.

If configure cannot find tiff.h or the tiff library and you have it installed in a non-standard place (even '/usr/local'), you may need to pass these flags to configure: CFLAGS="-I/usr/local/include" and LDFLAGS="-L/usr/local/lib".

## 1.10 FreeBSD 2.x (*Obsolete, Unstable*)

'Recommended compiler'

gcc 2.8.x

'Extra libs needed'

Unknown

'Special Instructions'

Only static libraries work on this system. Use /stand/sysinstall to install these packages if you have not already done so:

```
gmake          (GNU make)
gcc 2.8.x
```

Seems to compile ok, but some tests crash. Possibly due to a performace 'hack' in base. Might be a good idea to upgrade to FreeBSD 3.x. You need to use gmake not make to compile the GNUstep packages.

## 1.11 Irix 6.5/MIPS

‘Recommended compiler’

gcc 3.2.1

To use threads, it’s necessary to bootstrap a compiler yourself: configure with `-enable-threads=posix`, that will work as long as you link EVERY objective C executable with `-lpthread`, no matter what warnings the irix linker produces!

‘Extra libs needed’

Unknown

‘Special Instructions’

If you cannot link the library because of the very low default limit (20480) for the command line length, then you should either use `systune ncargs` to increase the value (maximum is 262144) or link the library by hand. No `libffi`-support: Use `ffcall`

## 1.12 MacOSX/PowerPC

Currently tested on MacOSX 10.2, 10.3

‘Recommended compiler’

Default.

‘Extra libs needed’

None.

‘Special Instructions’

By default, on Mac OS X, only the GNUstep extensions are built. This is if you want to build `gdl2`, etc on Mac OS X. Xcode project files exist, but they may not be up-to-date. Try:

```
cd make
./configure --with-library-combo=apple-apple-apple
make install

cd ../base
./configure --with-xml-prefix=/usr --disable-xmltest
make debug=yes install
```

See also the Darwin/PowerPC section.

## 1.13 MkLinux/PowerPC

Tested with R2 RC2 (2004/03/04).

‘Recommended compiler’

gcc 3.x although earlier ones should work if you also install `gnustep-objc`

‘Extra libs needed’

None.

‘Special Instructions’

Unknown.

## 1.14 NetBSD

'Recommended compiler'

Unknown

'Extra libs needed'

libiconv

'Special Instructions'

See the [README.NetBSD](#) file located in the gnustep-make package.

## 1.15 Netwinder (*Unstable*)

'Recommended compiler'

Build #12 of the system.

'Extra libs needed'

Unknown

'Special Instructions'

See <http://www.netwinder.org/~patrix>

## 1.16 OpenBSD 3.x (*Needs Testing*)

'Recommended compiler'

Unknown

'Extra libs needed'

Unknown

'Special Instructions'

Try reading the [README.NetBSD](#) which might work the same on OpenBSD.

## 1.17 OSF Alpha (*Unstable*)

Information is for Version 3.2C

'Recommended compiler'

egcs-1.1.1/1.1.2, gcc-2.95

'Extra libs needed'

Unknown

'Special Instructions'

Can only compile with static libraries. Compiler may fail when linking executables (e.g. gdc). Standard ranlib and ar programs are too feeble to create libraries. Should use GNU binutils versions. Linker sometimes fails to find symbols, in which case you may need to link with a library twice. For instance, add an extra -lgnustep-gui in ADDITIONAL\_TOOL\_LIBS in the GNUmakefile(.preamble).

## 1.18 RedHat/Intel

'Recommended compiler'

Standard

'Extra libs needed'

Standard (ffcall or libffi)

'Special Instructions'

None

## 1.19 Slackware/Intel

'Recommended compiler'

Unknown.

'Extra libs needed'

Unknown.

'Special Instructions'

Unknown.

## 1.20 Solaris 2.5.1/Sparc (*Obsolete*)

This configuration is no longer being tested, but it may still work.

'Recommended compiler'

Unknown

'Extra libs needed'

tiff, Don't use the one in /usr/openwin

'Special Instructions'

See the Solaris 2.6 section for more instructions.

## 1.21 Solaris 2.[678]/Sparc

Tested on Solaris version 6, 7, and 8

'Recommended compiler'

gcc 3.2.1 or greater gcc 2.95.3. Version 2.95.2 has several bugs that GNUstep tickles. gcc 3.04. Not 3.1 - does not compile parts of GNUstep.

'Extra libs needed'

tiff, Don't use the one in /usr/openwin

'Special Instructions'

Using a POSIX shell (zsh or bash, which should come with Solaris) is highly recommended. In fact, some functions, such as compiling frameworks, will not work without it.

Some people have reported problems when using binutils assembler and linker. Using the native Solaris assembler and linker should work fine.

Older Instructions: If you are using threads, make sure the Objective-C runtime (libobjc that comes with gcc) is compiled with threads enabled (This is true by default) AND that

it is compiled with the `_REENTRANT` flag defined (This does not seem to be true by default). Or use the `gnustep-objc` package. Also make sure `THREADS` is set to `'posix'` not `'solaris'`.

## 1.22 Solaris 2.7/Intel

'Recommended compiler'

Unknown.

'Extra libs needed'

Unknown

'Special Instructions'

Make sure there are no `-g` compiler flags (i.e. compiling with `debug=yes` might be a problem). Unsure of correct bundle flags - You might need to use the alternate flags listed in `target.make`, line 989. Also, configuring `gnustep-make` with `'--disable-backend-bundle'` might be necessary if you can't get bundles to work. You will probably get a lot of text relocation warnings, which probably can be ignored. See the other Solaris instructions above for more information.

## 1.23 Suse 6.x/Intel

GNUstep has been tested on version 6.2-6.4 of Suse

'Recommended compiler'

Standard

'Extra libs needed'

None

'Special Instructions'

It seems that there is a problem with the default kernel build distributed with Suse which means that the socket binding used by `gdnc` doesn't work. If you recompile the kernel then it starts working.

## 1.24 Suse/Intel

GNUstep has been tested on version 7.0, 8.0, 8.1, 8.2, 9.0, and 9.1 of Suse

'Recommended compiler'

The default compiler that comes with Suse is fine. Also `gcc2.95.x`, `gcc3.0.x`, `3.1` and `3.2` work, but `2.95` is faster. Compile with `-threads-enabled` (non-standard).

'Extra libs needed'

None

'Special Instructions'

None.

## 1.25 Suse 7.x/PPC

GNUstep has been tested on version 7.0 of Suse/PPC

'Recommended compiler'

Standard. gcc2.95.x, gcc3.0.x and gc3.1 work, but 2.95 is faster. Compile with  
-threads-enabled (non-standard).

'Extra libs needed'

None

'Special Instructions'

## 1.26 Unixware-2.1.3/Intel

'Recommended compiler'

Unknown

'Extra libs needed'

Unknown

Special Instructions for GNUstep installation on Unixware 2.1 systems

1 Tune the kernel to increase the argument space so that we can pass long  
command-line argument strings to processes (which the makefiles do)  
(/etc/conf/bin/idtune ARG\_MAX 102400)

2 Install raft of the latest GNU software

gzip (you need this to unpack other stuff)

make (to build everything)

m4 (for autoconf etc)

autoconf (if you need to change anything)

bison

flex

binutils (required by gcc if you want to debug)

gcc-2.8.1

(configure -with-gnu-as -with-gnu-ld -with-stabs)

NB. gcc-2.8.1 needs a fix to \_\_do\_global\_dtors\_aux()

in crtstuff.c on Unixware 2.1.3

(and possibly other unixware versions)

The fix is already in recent versions of egcs.

```
=====
static void
__do_global_dtors_aux ()
{
    static func_ptr *p = __DTOR_LIST__ + 1;
    static int completed = 0;

    if (completed)
        return;

    while (*p)
    {
```

```

        p++;
        (*(p-1)) ();
    }

#ifdef EH_FRAME_SECTION_ASM_OP
    __deregister_frame_info (__EH_FRAME_BEGIN__);
#endif
    completed = 1;
}
=====

```

3 Having got gcc working - it's probably a good idea to rebuild all your GNU software using it!

4 Build gstep as normal.

5 The SIOCGIFCONF ioctl sometimes doesn't work on unixware after applying some of the OS patches.

So I have added a '-a' flag to gdomap to give it the name of a file containing IP address and netmask information for the network interfaces on the system.

You need to set up a file (I suggest '/etc/gdomap\_addresses') containing the information for your machine and modify your system startup files in /etc/rc?.d to run gdomap, telling it to use that file.

eg. If your machine has an IP address of '193.111.111.2' and is on a class-C network, your /etc/gdomap\_addresses file would contain the line

```
193.111.111.2 255.255.255.0
```

and your startup file would contain the lines

```
. /usr/local/GNUstep/Library/Makefiles/GNUstep.sh
gdomap -a /etc/gdomap_addresses
```

If you don't set gdomap up correctly, Distributed Objects will not work.

## 1.27 Windows with CYGWIN (*Needs Testing*)

'Recommended compiler'

gcc 3.3.1 or later (with libobjc and libjava (if using libffi))

'Extra libs needed'

Objective-C library DLL (<ftp://ftp.gnustep.org/pub/gnustep/windows/cygwin>) for shared libs. It's a good idea to remove the libobjc.a that comes with gcc (gcc -v for location) so that it isn't accidentally found. For ffcall, you should get version 1.8b or above (the earlier ones don't compile). There are still some problems with structure passing, but that is generally not supported on any architecture. libffi also works.

'Special Instructions'

Make sure you have good shared libraries for everthing. Sometimes a bad shared library (like libtiff) will cause odd and untraceable problems. See [README.Cygwin](#) for information on compiling.

## 1.28 Windows with MinGW

'Recommended compiler'

See below.

'Extra libs needed'

See below.

'Special Instructions'

See the [README.MinGW](#) file located in the gnustep-make Documentation directory for instructions. Windows NT/2000/XP only. Win98 machines and earlier are very buggy and are not supported. Native GUI backend is alpha version.

## 1.29 Yellowdog/PowerPC

'Recommended compiler'

Standard

'Extra libs needed'

Standard (ffcall or libffi)

'Special Instructions'