
GNUstep

Installation and User's Guide

Version 0.6.5

February 8, 2000

Permissions

Copyright (C) 1999-2000 Free Software Foundation.

Permission is granted to make and distribute verbatim copies of this manual provided the copyright notice and this permission notice are preserved on all copies.

Permission is granted to copy and distribute modified versions of this manual under the conditions for verbatim copying, provided also that the entire resulting derived work is distributed under the terms of a permission notice identical to this one.

Permission is granted to copy and distribute translations of this manual into another language, under the above conditions for modified versions.

Apple™, OpenStep™, and the Apple logo are trademarks of Apple Computers, Inc. registered in the U.S. and other countries.

PostScript™, Display PostScript™ and Adobe™ are trademarks of Adobe Systems Incorporated and may be registered in certain jurisdictions.

Other company and product names mentioned herein might be trademarks of their respective companies.

About the Free Software Foundation

The Free Software Foundation (FSF) is dedicated to eliminating restrictions on copying, redistribution, understanding, and modification of computer programs. We do this by promoting the development and use of free software in all areas of computing---but most particularly, by helping to develop the GNU operating system.

Many organizations distribute whatever free software happens to be available. In contrast, the Free Software Foundation concentrates on development of new free software---and on making that software into a coherent system that can eliminate the need to use proprietary software.

Besides developing GNU, FSF distributes copies of GNU software and manuals for a distribution fee, and accepts tax-deductible gifts to support GNU development. Most of the FSF's funds come from its distribution service. This is why we urge you to order CD-ROMs and manuals (but especially CD-ROMs) from the FSF when you can. You may order GNU software from the FSF by filling in the order form located at the back of this document, and sending it to:

Free Software Foundation
59 Temple Place - Suite 330
Boston, MA 02111

+1-617-542-5942
Fax: +1-617-542-2652

Table of Contents

PERMISSIONS.....	2
TABLE OF CONTENTS	3
WELCOME TO GNUSTEP	5
WHAT IS OPENSTEP?	5
<i>The Origins of OpenStep</i>	5
WHAT IS OBJECTIVE-C AND WHY USE IT?	6
<i>The Origins of Objective-C</i>	7
WHAT IS GNUSTEP?.....	7
<i>The Origins of GNUstep</i>	8
<i>Beyond OpenStep</i>	8
<i>OpenStep Compliance</i>	8
<i>Our Time Frame</i>	9
GNUSTEP OVERVIEW	10
SOURCE TREE ORGANIZATION.....	10
SYSTEM REQUIREMENTS	11
GNUSTEP INSTALLATION.....	12
INTRODUCTION.....	12
<i>Basic Requirements</i>	12
<i>Obtaining the GNU Tools and GNUstep Distribution</i>	13
<i>Getting Libraries via Anonymous CVS</i>	14
<i>Compiling and Installing the GNU Tools</i>	15
Compiling GNU make.....	15
Compiling PThreads (Linux based systems only).....	16
Compiling GCC.....	17
Patching and Compiling GDB.....	19
Compiling the TIFF library.....	20
Compiling the Display Ghostscript System (DGS).....	21
INSTALLING GNUSTEP	22
<i>First Steps</i>	22
<i>Generic UNIX-based Installation</i>	23
Using the GNUstep 'core' Package.....	23
Installing Individual Packages.....	24
<i>Installing the Makefile Package</i>	24
<i>Building a Foundation Library (Foundation Kit)</i>	26
<i>Building the GUI Library (Application Kit)</i>	28
<i>Building the X-Windows DPS Backend</i>	29
<i>Building the Win32 Native Backend (Experimental)</i>	30
<i>Building the X-Windows GPS Backend (Primary)</i>	31
<i>Special Options for GNUstep Installation</i>	32
<i>Windows 95 / 98 / NT Specific Installation</i>	32
<i>Solaris Specific Installation</i>	33
<i>GNU / Linux Specific Installation</i>	33
<i>GNU / Linux PowerPC (LinuxPPC 4.x) Specific Installation</i>	33
<i>FreeBSD 3.x Specific Installation</i>	33
<i>Unixware Specific Installation</i>	34

<i>GNUstep Post Installation</i>	36
<i>Compiling Test Tools and Applications</i>	36
<i>Performance Tuning GNUstep Libraries</i>	37
Tuning Objective-C Runtime Debugging Information	37
THE GNUSTEP TOOLS	38
GDOMAP	38
OPENAPP	39
DEFAULTS	39
PLPARSE	40
SFPARSE	40
MAKE_SERVICES	40
SET_SHOW_SERVICE	41
FREE SOFTWARE FOUNDATION ORDER FORM	42
FSF DELUXE DISTRIBUTION	42
CD-ROMS, IN ISO 9660 FORMAT	43
MANUALS	44
REFERENCE CARDS	45
T-SHIRTS	45
OLDER ITEMS	46
OLDER CD-ROMS	46
TAX AND SHIPPING COSTS	48
SHIPPING INFORMATION	49

Welcome to GNUstep

An Introduction to the Free Software Implementation of OpenStep

What is OpenStep?

OpenStep is an open application-programming interface (API) derived from the operating system-independent subset of NeXTSTEP Version 3.2. OpenStep contains three major components:

- ❑ **Application Kit:** The Application Kit™ provides the basic software for writing interactive applications that use windows, draw on the screen, and respond to user actions on the keyboard and mouse. The Application Kit contains the components that define the OpenStep user interface.
- ❑ **Foundation Kit:** The Foundation Kit™ provides the fundamental building blocks that applications use to manage data and resources. It defines facilities for handling Unicode character sets, object persistency and distribution, and provides an interface to common operating system facilities.
- ❑ **Display PostScript System:** The Display PostScript® system provides OpenStep with its device-independent imaging model based on the PostScript language developed by Adobe Systems Incorporated.

The OpenStep API is expressed in the Objective-C language, an object-oriented extension of ANSI C.

The Origins of OpenStep

In November 1993, NeXT Computer Inc. (now part of Apple Computers, Inc.), and Sun Microsystems Inc. teamed up to develop a free object API layer based on the NeXTSTEP operating system. This agreement evolved into the OpenStep Specification that was published by NeXT in its first draft in the summer of 1994.

Both Sun Microsystems, Inc. and NeXT Computers, Inc. developed OpenStep implementations. However, Sun's implementation of OpenStep stopped after version 1.1 as Sun refocused its efforts into the development of Java. Currently the only commercially available OpenStep implementation is through Apple Computers, Inc. which bought NeXT Computers, Inc. in 1997.

What is Objective-C and Why Use It?

The Objective-C language was chosen for the OpenStep specification, for a variety of reasons:

- ❑ It is an object-oriented language. The kind of functionality that is defined in the OpenStep specification can only be delivered through object-oriented techniques.
- ❑ Objective-C is an extension of standard ANSI C, existing C programs can be adapted to use the software frameworks without losing any of the work that went into their original development. Since Objective-C incorporates C, you get all the benefits of C when working within Objective-C. You can choose when to do something in an object-oriented way (define a new class, for example) and when to stick to procedural programming techniques (define a structure and some functions instead of a class).
- ❑ Objective-C is a simple language. Its syntax is small, unambiguous, and easy to learn. Object-oriented programming, with its self-conscious terminology and emphasis on abstract design, often presents a steep learning curve to new recruits. A well-organized language like Objective-C can make becoming a proficient object-oriented programmer that much less difficult.
- ❑ Objective-C is the most dynamic of the object-oriented languages based on C. The compiler throws very little away, so a great deal of information is preserved for use at run time. Decisions that otherwise might be made at compile time can be postponed until the program is running. This gives Objective-C programs unusual flexibility and power. For example, Objective-C's dynamism yields two big benefits that are hard to get with other nominally object-oriented languages:
 - ❑ Objective-C supports an open style of dynamic binding, a style than can accommodate a simple architecture for interactive user interfaces. Messages are not necessarily constrained by either the class of the receiver or the method selector, so a software framework can allow for user choices at run time and permit developers freedom of expression in their design.
 - ❑ Objective-C's dynamism enables the construction of sophisticated development tools. An interface to the run-time system provides access to information about running applications, so it's possible to develop tools that monitor, intervene, and reveal the underlying structure and activity of Objective-C applications. Apple's Interface Builder (and GNUstep's Interface Modeler) could not have been developed with a less dynamic language.
- ❑ The Objective-C language is much easier to learn than C++, and shares a common object model with the Java programming language, making porting from Objective-C to Java and visa versa unusually easy.

The Origins of Objective-C

Brad J. Cox, whose primary purpose was to add the main features of Smalltalk-80 to the C language, designed Objective-C. His work led to an object-oriented language, with a complete programming environment inspired by Smalltalk-80, even comprising a large part of the latter's basic library.

from "Object Oriented Languages", Masini et al, Academic Press

What is GNUstep?

GNUstep is part of the GNU Project of the Free Software Foundation (<http://www.gnu.org/>). It's primary goal is to provide a free software implementation of the OpenStep specification portable to as many different variants of UNIX as possible, and perhaps even the Microsoft Windows 95/98/NT platforms.

GNUstep is best explained in terms of OpenStep:

- ❑ **The Foundation Kit and the GNUstep Base Library:** The implementation of the Foundation Kit as specified in the OpenStep specification is the GNUstep Base Library; this is the underlying support for all of GNUstep. It grew out of the `libobjects` library (see Origins of GNUstep, below), extended and repackaged to fulfill the Foundation Kit's functionality.
- ❑ **The Application Kit and the GNUstep GUI Library:** The Application Kit is closely tied to the Display PostScript System because all of the graphics drawing is implemented in the PostScript language. Implementing a Display PostScript System would be a substantial task, so GNUstep has taken an approach that would allow the Application Kit implementation to continue even before the Display PostScript System was completed.

The Application Kit implementation is separated into two parts; a front-end and a back-end. The front-end is independent of the graphics display system, so only implements behavior without performing any display operations. The back-end is specific to a graphics display system and only needs to perform the display operations for that graphic system. The front-end is called the GNUstep GUI Library, and at least two back-ends have been implemented. One of the backends is for the X-Windows System (`xgps`); the other, for the combination of the X Windows System and the Display PostScript System called the GNUstep X/DPS GUI Backend (`xdps`).

- ❑ **Display PostScript and Display GhostScript:** GNUstep developers have a free software version of the Display PostScript System called the Display GhostScript System (DGS) that is, in part, based upon the GhostScript program. On systems that have the Adobe Display PostScript System as its native backend, GNUstep will use the Adobe DPS system instead of DGS.

The Origins of GNUstep

Early on, many developers (educational and corporate) realized that the NeXTSTEP environment provided a great tool to developers by reducing the amount of time necessary for writing applications through feature-rich APIs. Still the portability issue was hard to solve since these applications were tied to machines running NeXTSTEP.

The team around Paul Kunz at the Stanford Linear Accelerator Center (SLAC) decided to port their NeXTSTEP application, HippoDraw, differently than others did. Instead of rewriting from scratch on the new system and only "reusing" the overall application design and ideas, they rewrote the objects their application depended upon — the NeXTSTEP object layer.

Consequently, they created the first version of `libobjcX`, which enabled them to port HippoDraw to UNIX systems running X Windows without changing a single line of their application's source.

After the OpenStep initiative became public, the next logical step was to make a new "libobjcX" which would adhere to the new APIs. `libobjcX` is the historical base for the GNUstep GUI library.

Beyond OpenStep

While implementing a free software version of the OpenStep specification is a great idea; GNUstep is growing beyond this initial task to become a powerful development environment and a sophisticated user environment. The Developer Suite provides the tools and applications that make up the GNUstep development environment; and the User Suite page of the GNUstep website (<http://www.gnustep.org/>) describes some of the applications we have in mind for the user environment.

The GNUstep Database Library and the GNUstep 3DKit are examples of two auxiliary projects that take OpenStep as the base API for extending into other areas; the former into database programming and the latter into 3D graphics.

As GNUstep has advanced from concept to implementation, increasingly more people and institutions are helping in this development. However, there are (as with any free software effort) many additional projects for which volunteers are always welcome. Several of them can be found on the GNUstep Website under the Resources page.

OpenStep Compliance

The motivation behind GNUstep is to make porting our applications easier, to leverage the benefits of the Objective-C language, and to push the idea of OpenStep as a common object oriented API. This is especially important on platforms that are not supported by commercial implementations of the OpenStep standard.

We are not claiming to be OpenStep compliant at this time nor in the near future as having Display PostScript on every target system remains a big issue. Additionally, we must pass Apple's compliance test to be considered an OpenStep compliant system.

Our Time Frame

Most GNU projects have no real time frame since all development is done for free. The current team is very interested in pushing the project ahead, but time and space will remain limited. So, if things are moving too slow for you... please join us and help!

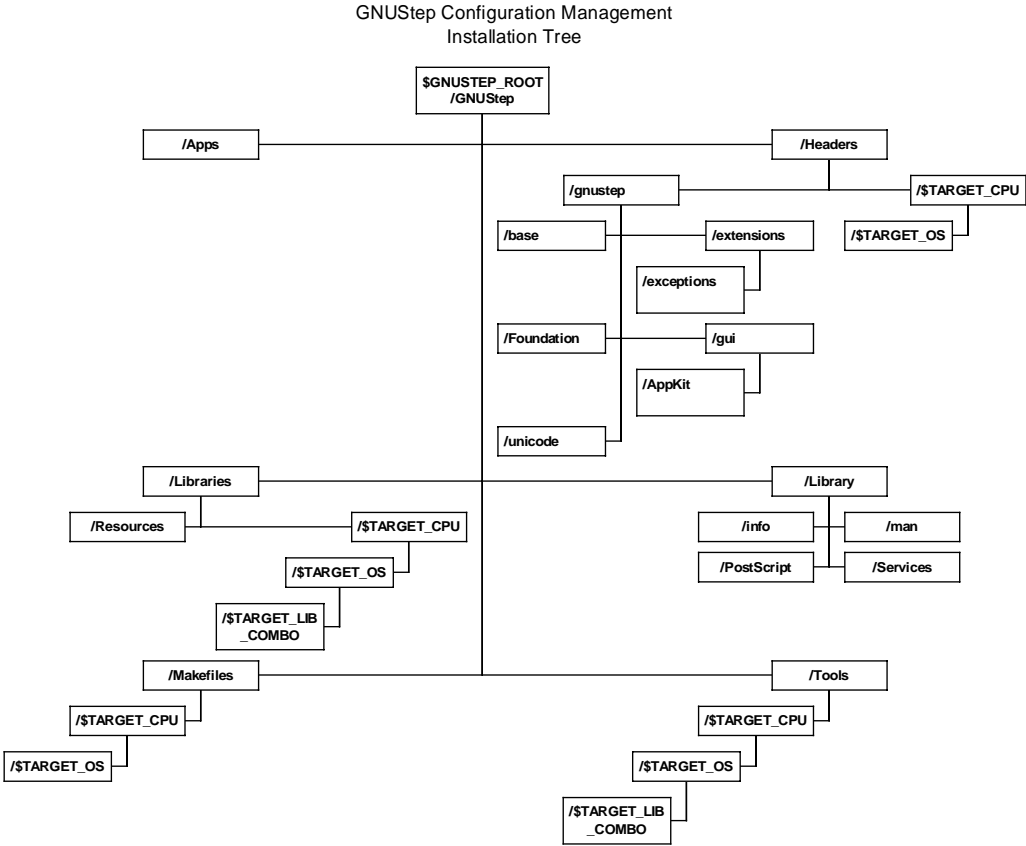
GNUstep Overview

System Requirements and Environment Organization

Source Tree Organization

The GNUstep distribution is designed for installation on most UNIX based systems, as well as Windows 95/98/NT using the Cygwin UNIX libraries available from Cygnus Solutions, Inc. (<http://www.cygnus.com/>).

The installed GNUstep distribution is organized so that it maintains the division between the Objective-C libraries and the host operating system and processor specific code. This organization maintains maximum portability between operating systems.



System Requirements

GNUstep makes few assumptions about the hardware platform on which it is installed. For the most part, any machine capable of running one of the supported operating systems outlined in this document will be able to support GNUstep.

The GNUstep environment is intended to provide a rich environment for application development. As such, a system should have both ample memory and free hard disk space to accommodate application development

In order to easily compile and debug GNUstep projects, you will need the GNU Objective-C compiler called “`gcc`” or “`egcs`” and a modified version of the GNU debugger “`gdb`”.

You will also need at least 250 MB of free hard disk space in order to compile the GNUstep project. Luckily, you can delete the object files and the uncompressed source code after each piece of the project has been successfully built and installed in order to save some space. The biggest offender is the `gcc` compiler, which eats up 50 MB of hard disk space while it compiles.

GNUstep Installation

Operating System Specific Installation Procedures

Introduction

Basic Requirements

This chapter explains how to build the GNUstep project along with all the tools required to develop applications with it. Pre-compiled binaries are also available for several operating systems. Refer to the GNUstep website (<http://www.gnustep.org/>) for more information.

Currently, you need to get and install the following packages:

- `gcc` or `egcs`
- `gdb` and `patch` to make it work better with Objective-C
- GNU `make`
- TIFF library
- DGS or Adobe DPS (if you intend to use the `gstep-xdps` backend).
- `gstep-core` which is composed of the individual packages:
 - `gstep-make`
 - `gstep-base` or `libFoundation`
 - `gstep-gui`
 - `gstep-xdps`
 - `gstep-xgps`

Obtaining the GNU Tools and GNUstep Distribution

Below is a table indicating the tools necessary to compile the GNUstep project, the minimum revision required, and a URL to a website where the tools can be found:

Tool or Library Name	Minimum Revision	Website URL ¹
<i>Compiler:</i>		
GCC	2.8.0	ftp://ftp.gnu.org/pub/gnu
EGCS	1.1.1	ftp://ftp.cygnum.com/pub/egcs
GDB debugger	4.17	ftp://ftp.gnu.org/pub/gnu
Objective-C Patch for GDB	980505	ftp://alpha.gnu.org/gnu/gnustep/contrib
GNU Make	3.75	ftp://ftp.gnu.org/pub/gnu
TIFF library	3.4 Beta 036	ftp://alpha.gnu.org/gnu/gnustep
Pthreads (PCThreads for Linux libc 5.x based machines)	1.1.0	ftp://alpha.gnu.org/gnu/gnustep
<i>GNUstep Sources:</i>		
Core (contains packages below) <i>Make, base, gui, xgps and xdps</i>	0.6.5	ftp://ftp.gnu.org/gnu/gstep
xgps (X-Windows backend)	0.6.5	
xdps (DPS backend)	0.6.5	
DGS (Display Ghostscript System)	0.5.0	ftp://alpha.gnu.org/gnu/gnustep
libFoundation (Alternate Foundation)	0.8.0	ftp://alpha.gnu.org/gnu/gnustep

You may get either one of the Foundations libraries `gstep-base` or `libFoundation`. You only need to take one of them, not both. Note that you should get the latest version available, not necessarily the one listed here.

¹ GNU Mirror Sites have these files available. Check the GNU website at <http://www.gnu.org/> for a list of mirror sites.

For the more adventurous, and since the project is still in development, it might be desirable to get the latest snapshot.

Special Note: *GNUstep snapshots are untested development releases. As such, they may not work correctly, or even compile. The reader should be aware that using snapshots is generally for active and experienced developers only – your mileage may vary.*

Snapshots are located in the snap or snapshot subdirectories at:

ftp://alpha.gnu.org/gnu/gnustep/snap

or the other appropriate ftp locations.

For the following instructions, some version numbers are replaced by XXX. You should replace the 'X's with the version of the package you have.

Getting Libraries via Anonymous CVS

If you did not get one of the snapshots, or if you want to be sure to stay on the bleeding edge, then you should get the core via CVS. For Windows 95/98/NT users, CVS can be obtained with a GUI front-end by downloading WinCVS from the website <http://mirror.openspace.ch/maccvs/>. The following steps will allow you to get the latest GNUstep source:

- ❑ First, set the CVSROOT environment variable. If you are using *bash/sh* you can do something like this at the prompt:

```
$export CVSROOT=":pserver:anoncvs@cvs.net-community.com:gnustep"
```

- ❑ Second, if this is the first time logging into the CVS server:

```
$ cvs login <press enter>
```

You should get a password prompt soon after:

```
(Logging in to anoncvs@cvs.net-community.com)
```

```
CVS password:
```

- ❑ Enter the password ``anoncvs'`. This should return you to your prompt. From here you can checkout any module in the CVS server you like. To checkout a module you do this:

```
$ cvs -z3 checkout modulename <press enter>
```

The `-z3` merely tells the CVS server to compress everything to a certain compression level before it sends it to you.

To obtain a list of modules, which are available on the CVS server, enter the following command:

```
$ cvs checkout -c <press enter>
```

- ❑ If you have not already done so, change to the directory, where you want the source to reside.

- ❑ Next, you want to get the whole core, so you do:

```
$ cvs -z3 checkout core
```

After you have checked out the source you can compile it as usual.

- ❑ To update the source, go into the directory of the source tree you want to update, for example, go into `xgps`, and type:

```
$ cvs -z3 -d update <press enter>
```

You don't have to re-checkout after you have the source, just update!

Compiling and Installing the GNU Tools

Note that the following instructions assume you are installing the libraries in a directory owned by 'root' (Administrator under Windows NT). If you are installing the libraries in a place that you as a user can write to, you can skip the ``su root'` instructions.

Special Note for Windows 95/98/NT Users: *These packages assume a pre-existing UNIX command layer exists on your machine. If you do not have either Cygwin or Mingw32 installed on your machine, you must obtain and install one of these packages first – see the section entitled 'Windows 95/98/NT Specific Installation'. After which, you may wish to go through the steps below, though currently Cygwin has the basic tools such as gcc/egcs, make, and gdb already compiled and installed as part of the system. The instructions below however are helpful if you wish to install updated versions of the software.*

Compiling GNU make

GNU `make` is required to compile all GNUstep packages. Most "free" operating systems come with GNU `make` already, however you may use these instructions to compile newer releases of the package. To determine the version on your system,

type 'make -- version'. Some systems also refer to the GNU make package as gmake. To compile the make package:

- ❑ Change your directory to a location where the GNU make source can be decompressed:

```
cd /usr/src
```

- ❑ Uncompress the make package into the directory:

```
tar fvzx make-XXX.tar.gz
```

- ❑ Change into the newly created 'make' directory:

```
cd make-XXX
```

- ❑ Configure the package by typing the following:

```
./configure
```

- ❑ Build the package by typing:

```
make
```

- ❑ To install the package you must have administrative or 'root' privileges. To install, switch to 'super-user' mode by typing:

```
su root
```

- ❑ Finally, install the package by typing the command:

```
make install
```

Compiling PThreads (Linux based systems only)

Special Note: *This threading package only works for Linux systems. Even some Linux systems do not require this package as they come with posix thread support. Refer to the **Compiling GCC** section below for threading options on other platforms.*

For GNU/Linux systems on Intel x86 processors, the pthreads library is known to work with GNUstep. PThreads version 1.1.0 is an enhanced version of PThreads 1.0.0 with better support for GNUstep. *PThreads may no longer be necessary on all GNU/Linux systems, such as RedHat 5.x or later, Debian 2.x or later, or other systems with glibc2.* To install the package, follow these steps:

- ❑ Change your directory to a location where the PThreads source can be decompressed:

```
cd /usr/src
```


- ❑ Uncompress the package into the directory:

```
tar fvzx pthreads-1.1.0.tar.gz
```

- ❑ Change into the newly created 'pthreads' directory:

```
cd pthreads-1.1.0
```

- ❑ Configure the package by typing the following:

```
./configure
```

If you are using the GNUstep makefile package and wish PThreads to be installed with the GNUstep directory structure; then you should specify the GNUstep system root as the prefix when you configure the package.

```
./configure --prefix=$GNUSTEP_SYSTEM_ROOT
```

- ❑ Build the package by typing:

```
make
```

- ❑ To install the package you must have administrative or 'root' privileges. To install, switch to 'super-user' mode by typing:

```
su root
```

- ❑ Finally, install the package by typing the command:

```
make install
```

Compiling GCC

The following commands assume that all the aforementioned files have been copied to the /usr/src directory on your machine. If you already have GCC 2.8.0 installed on your system then there is no need to compile it again, just set up your PATH so that the new GCC is used for compiling GNUstep.

Special note: Please check for machine specific instructions under the section titled **Special Options for GNUstep Configuration** to ensure there are not any specific compiler configurations required by your operating environment.

Here are the steps to install GCC (or 'egcs') on your system:

- ❑ Change your directory to a location where the GCC or egcs source can be decompressed:

```
cd /usr/src
```

- ❑ Uncompress the package into the directory:

```
tar fvxz gcc-2.8.0.tar.gz
```

- ❑ Change into the newly created 'gcc' directory:

```
cd gcc-2.8.0
```

- ❑ Configure the package by typing the following:

```
./configure
```

If you are using the GNUstep makefile package and wish `gcc` to be installed within the GNUstep directory structure; then you should specify the GNUstep system root as the prefix when you configure the package. This setup requires that you compile and install the GNUstep Makefile Package before configuring GCC.

```
./configure --prefix=$GNUSTEP_SYSTEM_ROOT
```

To enable multiple threads in the Objective-C library use the `--enable-threads=LIB` argument to configure, where `LIB` is the thread library to use or `--disable-threads` to use a single-threaded Objective-C runtime. If your system has a native thread library, like Solaris or SGI, then the `--enable-threads` parameter without the `LIB` specification will use that native thread library. Some of the 'LIB' options, which you may use, are:

- ◆ `--enable-threads=pthreads` (for use with PCThreads only).
 - ◆ `--enable-threads=posix` (for `glibc2` based machines, and other Posix compliant machines.).
 - ◆ `--enable-threads` (for Solaris, SGI, and Windows NT).
- ❑ Build the package by typing:

```
make bootstrap
```

It takes a substantial amount of time to compile 'gcc' (or 'egcs'). As a rule of thumb, here is your estimated wait time:

- ◆ *If you have a, Pentium II, or UltraSPARC - go drink a coffee.*
 - ◆ *If you have a fast Pentium or Pentium Pro - count from 0 to 500 in your head slowly.*
 - ◆ *If you have a slow Pentium, go eat a dozen bagels (more if you eat quickly).*
 - ◆ *If you have a 486, go watch the news.*
 - ◆ *If you have a 386, come back tomorrow.*
- ❑ To install the package you must have administrative or 'root' privileges. To install, switch to 'super-user' mode by typing:

```
su root
```

- ❑ Finally, install the package by typing the command:

```
make install
```

- ❑ Delete the 'gcc' directory if you're low on disk space:

```
rm -rf /usr/src/gcc-2.8.0
```

Patching and Compiling GDB

GDB is the GNU Debugger that will be an essential debugging tool while you develop applications using GNUstep. To install the package, follow these steps:

- ❑ Change your directory to a location where the GDB source can be decompressed:

```
cd /usr/src
```

- ❑ Uncompress the package into the directory:

```
tar fvzx gdb-4.17.tar.gz
```

- ❑ Uncompress the GDB patch for Objective-C package into the directory:

```
gunzip gdb-4.17-objc-unoff-980505.patch.gz
```

- ❑ Change into the newly created 'gdb' directory:

```
cd gdb-4.17
```

- ❑ Patch the source code with the following command:

```
patch -p1 < ../gdb-4.17-objc-unoff-980505.patch
```

- ❑ Configure the package by typing the following:

```
./configure
```

If you are using the GNUstep makefile package and wish `gdb` to be installed within the GNUstep directory structure; then you should specify the GNUstep system root as the prefix when you configure the package. This setup requires that you compile and install the GNUstep Makefile Package before configuring GDB.

```
./configure --prefix=$GNUSTEP_SYSTEM_ROOT
```

- ❑ Build the package by typing:

```
make
```

- ❑ To install the package you must have administrative or 'root' privileges. To install, switch to 'super-user' mode by typing:

```
su root
```

- ❑ Finally, install the package by typing the command:

```
make install
```

- ❑ Make sure GDB is where you think it is. On Linux/GNU, GDB should be installed in `/usr/bin`, whereas on other systems, the default installation path is usually `/usr/local/bin`. If you have a Linux/GNU box, you can do this:

```
ln -s /usr/local/bin/gdb /usr/bin/gdb
```

Note that this is only a suggestion. You are free to do whatever you want.

Compiling the TIFF library

You do not need a customized version of the TIFF library, or necessarily even the newest one, but the customized version located at the GNUstep ftp sites will install in the GNUstep directory structure.

If you do use the customized version, you need to make and install the GNUstep Makefile Package (described in the **Installing GNUstep** section below) first. To install the TIFF library:

- ❑ Change your directory to a location where the TIFF library source can be decompressed:

```
cd /usr/src
```

- ❑ Uncompress the TIFF package into the directory:

```
tar fvzx tiff-XXXXX.tar.gz
```

- ❑ Change into the newly created 'tiff' directory:

```
cd tiff-XXX
```

- ❑ Configure the package by typing the following:

```
./configure
```

- ❑ Build the package by typing:

```
make
```

- ❑ To install the package you must have administrative or 'root' privileges. To install, switch to 'super-user' mode by typing:

```
su root
```

- ❑ Finally, install the package by typing the command:

```
make install
```

Compiling the Display Ghostscript System (DGS)

If you have a real Display PostScript system already on your computer, or if you do not intend to use the `gstep-xdps` graphical backend, you do not necessarily need the Display Ghostscript System (DGS). Linux/GNU machines need DGS however if `gstep-xdps` is to be used. To install the Display Ghostscript System follow these steps:

- ❑ Change your directory to a location where the DGS source can be decompressed:

```
cd /usr/src
```

- ❑ Uncompress the DGS package into the directory:

```
tar fvzx dgs-XXXXXX.tar.gz
```

- ❑ Change into the newly created 'dgs' subdirectory:

```
cd dgs-XXXXXX/dgs
```

- ❑ Configure the package by typing the following:

```
./configure
```

- ❑ Build the package by typing:

```
make
```

- ❑ To install the package you must have administrative or 'root' privileges. To install, switch to 'super-user' mode by typing:

```
su root
```

- ❑ Finally, install the package by typing the command:

```
make install
```

Installing GNUstep

First Steps

If you have the `gstep-core` package, you can make and install the entire GNUstep core at one time, provided you already have a TIFF library and a Display PostScript System (DPS) installed (if desired).

Special Note: *For users who do not wish to use Display PostScript, or Display Ghostscript as your imaging model under GNUstep, you may use an optional X-Windows backend, or Win32 Native backend. Only the TIFF library is essential.*

If you have the TIFF library and either a DPS/DGS or optional backend installed, but you have the separate 'gstep' packages, you must install the Makefile Package first (see the section below - **Installing the Makefile Package**) before continuing. Otherwise, you may proceed to the GNUstep installation instructions for your particular operating system (if appropriate), or use the **Generic UNIX-based Installation** instructions.

For more advanced users, there are a number of options available for installation of GNUstep. Please refer to the section entitled **Special Options for GNUstep Installation**.

As a final note, please remember:

You must follow the operating system specific instructions (where applicable) closely to ensure proper GNUstep installation. If you are unable to find your particular operating system in this document, it is possible that either the Generic UNIX-based installation instructions are sufficient. It is also possible that specific instructions for your operating system are included in the GNUstep-HOWTO file included with the 'core' package. The GNUstep-HOWTO file always contains the most up-to-date instructions for system specific installation instructions.

Generic UNIX-based Installation

Before proceeding with a generic installation, ensure that you have reviewed the operating system specific instructions for your system (where applicable) and follow the instructions indicated.

If you have already completed any OS-specific pre-installation steps, you are now ready to install GNUstep's core libraries.

There are two methods for installing GNUstep: the first is to install it using the `gstep-core` package which contains all of the smaller packages together and has a build script which builds each package for you.

However, there may be times where either you do not have access to the `gstep-core` package, or you wish to build an individual library which has been updated without rebuilding the entire GNUstep source tree. The following details how to build each package starting with `gstep-core`:

Using the GNUstep 'core' Package

The GNUstep core package contains a number of smaller packages that provide all the basic components for GNUstep. If you do not have the 'gstep-core' package, you may either download it, or proceed to the next section for instructions (see **Installing Individual Packages**) on how to install each of the packages individually. If you have the 'gstep-core' package, the installation is quite simple:

- ❑ Change your directory to a location where the GNUstep core source can be decompressed:

```
cd /usr/src
```

- ❑ Uncompress the make package into the directory:

```
tar fvzx gstep-core-XXXXX.tar.gz
```

- ❑ Change into the newly created 'gstep-core' directory:

```
cd gstep-core-XXXX
```

- ❑ Configure the package (with the applicable prefix, if desired) by typing the following:

```
./configure --prefix=/usr/local/GNUstep
```

- ❑ Build the package by typing:

```
make
```

- ❑ To install, switch to 'super-user' mode by typing:

```
su
```

- ❑ Finally, install the package by typing the command:

```
make root-install
```

Skip to the **GNUstep Post-Installation** section to complete installation of the GNUstep system. Be sure however to perform any additional instructions which may be required by your particular operating system.

Installing Individual Packages

If you have not used the `gstep-core` package to install GNUstep, then you will need to install each package individually. After installing each package, you will need to continue by performing the steps in the section entitled **GNUstep Post Installation**. As well, be sure to complete any operating system specific instructions that may apply.

Installing the Makefile Package

Special Note: *Installing this package separately is only required in special cases such as where you need to install the TIFF library, or install the DGS system, and you wish these packages to be placed in the GNUstep source tree. It is also the first package that must be installed if you are compiling each of the GNUstep libraries individually instead of using the 'core' package. For most installations, we recommend that you simply install GNUstep by using the 'core' package and following the instructions for your particular operating system.*

First, decide where you want the GNUstep files to go, and use this as an argument to configure. If you do not choose a location, the default is `/usr/GNUstep`. The example below shows how to configure the packages for installation in `/usr/local/GNUstep`. To install the GNUstep Makefile package, follow these steps:

- ❑ Change your directory to a location where the GNUstep makefile source can be decompressed:

```
cd /usr/src
```

- ❑ Uncompress the make package into the directory:

```
tar fvzx gstep-make-XXXXXX.tar.gz
```

- ❑ Change into the newly created 'make' directory:

```
cd gstep-XXXXXX/make
```

- ❑ Configure the package by typing the following:

```
./configure --prefix=/usr/local/GNUstep
```


If you are using, or intend to compile the GCC Objective-C runtime with threads, you need to specify the thread package you are using when running configure:

```
LIBS=-pthread; ./configure --prefix=/usr/local/GNUstep
```

or if you have glibc2:

```
LIBS=-posix; ./configure --prefix=/usr/local/GNUstep
```

- ❑ Build the package by typing:

```
make
```

- ❑ To install the package you must have administrative or 'root' privileges. To install, switch to 'super-user' mode by typing:

```
su root
```

- ❑ Finally, install the package by typing the command:

```
make install
```

- ❑ After this you should add the shell script 'GNUstep.sh' in the makefile package to your shell startup file (such as the '.profile' file). For instance:

```
. /usr/local/GNUstep/Makefiles/GNUstep.sh
```

It defines environment variables that are needed to find GNUstep files and executables. Users of `csh` need to use the 'GNUstep.csh' script. Read the 'README' for more info.

The last step ensures that GNUstep packages will automatically install in this directory structure. You can install other packages (such as GCC) in the same structure if you specify the GNUstep system root as the prefix when you configure the package.

```
./configure --prefix=$GNUSTEP_SYSTEM_ROOT
```

Building a Foundation Library (Foundation Kit)

GNUstep comes with two variants of Foundation Library. Both of these libraries provide the classes and functionality of the OpenStep (and Apple's) 'Foundation Kit'. 'gstep-base' is the default library, with 'libFoundation' being provided as another option. As an implementation of the OpenStep Foundation Kit, and therefore a potential replacement for `gnustep-base`, `libFoundation` is more slim than `gstep-base`, and lacks a few features of the Foundation Kit. However, it implements some extensions to Foundation that are also available separately for use with `gnustep-base` or NeXT's FoundationKit.

Special Note: *The `libFoundation` library does not support Distributed Objects or work with `gdomap` (see Chapter 4 – Tools). This, for the most part, prevents proper functioning with the GNUstep GUI library. `libFoundation` also lacks several Mac OS X implemented classes. The library however can be used for standalone non-GUI projects, including those which require `gstep-db` (the GNUstep database library) and has built-in support for garbage collection. As well, `gstep-core` is provided under the GNU Public License – `libFoundation` has a different licensing scheme.*

To build GNUstep using the `gstep-base` libraries perform the following steps:

- ❑ Change your directory to a location where the GNUstep base source can be decompressed:

```
cd /usr/src
```

- ❑ Uncompress the make package into the directory:

```
tar fvzx gstep-base-XXXXXX.tar.gz
```

- ❑ Change into the newly created 'gstep-base' directory:

```
cd gstep-core-XXXXXX/base
```

- ❑ Configure the package (with the applicable prefix, if desired) by typing the following:

```
./configure --prefix=/usr/local/GNUstep
```

- ❑ Build the package by typing:

```
make
```

On some Linux systems, you may have to change the optimization level (particularly if you get an error compiling `mframe.m`), in this case use 'make' with this option:

```
make OPTFLAG=-O
```

- ❑ To install the package you must have administrative or 'root' privileges. To install, switch to 'super-user' mode by typing:

```
su root
```

- ❑ Finally, install the package by typing the command:

```
make install
```

If you wish to use the libFoundation libraries, follow these steps:

- ❑ Change your directory to a location where the libFoundation source can be decompressed:

```
cd /usr/src
```

- ❑ Uncompress the make package into the directory:

```
tar xzf libFoundation-0.8.0.tar.gz
```

- ❑ Change into the newly created 'libFoundation' directory:

```
cd libFoundation-0.8.0/libFoundation
```

- ❑ Configure the package (with the applicable prefix, if desired) by typing the following:

```
./configure --with-gnustep
```

- ❑ Build the package by typing:

```
make
```

- ❑ To install the package you must have administrative or 'root' privileges. To install, switch to 'super-user' mode by typing:

```
su root
```

- ❑ Finally, install the package by typing the command:

```
make install
```

You now have the GNUstep equivalent to the OpenStep Foundation Kit installed.

Building the GUI Library (Application Kit)

GNUstep has, as mentioned in Chapter 1, a GUI subsystem that is divided into two parts: a front-end and a back-end. The front-end is a library with provides the API equivalent to the OpenStep / Apple Application Kit – this is the function of the `gstep-gui` library. After building this library, you will need to build the appropriate backend for the graphic display system you are using.

To build the GUI Library, follow these steps:

- ❑ Change your directory to a location where the GNUstep GUI source can be decompressed:

```
cd /usr/src
```

- ❑ Uncompress the GUI package into the directory:

```
tar zxf gstep-gui-XXXXXX.tar.gz
```

- ❑ Change into the newly created 'Foundation' directory:

```
cd gstep-XXXXXX/gui
```

- ❑ Build the package (you don't need to configure the package, it's automatic) by typing:

```
make
```

- ❑ To install the package you must have administrative or 'root' privileges. To install, switch to 'super-user' mode by typing:

```
su root
```

- ❑ Finally, install the package by typing the command:

```
make install
```

At this point, you must install a GUI backend library before you will be able to compile any GUI-related applications, including the GNUstep development tools.

Special Note: *Version 0.5.5 of GNUstep contains several potential back-end libraries. However, the library that is the most mature is the "xgps" back-end. Ultimately, the target is for the "xdps" back-end to be the default (primarily on X Windows machines). Other back-end libraries such as "w32gui", the Microsoft Windows back-end, are still in early stages of development and as such will not provide the depth of functionality that the "xgps" library currently provides. Please adjust your expectations accordingly.*

Building the X-Windows DPS Backend

The 'xdps' package provides the glue between the DPS/DGS system that you have (or installed) on your system and the GNUstep GUI library. This library provides the translation from GUI library command into DPS.

To build the xdps package, follow these steps:

- ❑ Change your directory to a location where the xdps source can be decompressed:

```
cd /usr/src
```

- ❑ Uncompress the xdps package into the directory:

```
tar xzf gstep-xdps-XXXXXX.tar.gz
```

- ❑ Change into the newly created 'xdps' directory:

```
cd gstep-XXXXXX/xdps
```

- ❑ Configure the package (with the applicable prefix, if desired) by typing the following:

```
./configure --prefix=/usr/local/gnustep
```

- ❑ Build the package by typing:

```
make
```

- ❑ Build the Testing package by typing:

```
cd Testing
```

```
make
```

- ❑ Return to the parent menu for installation by typing:

```
cd ..
```

- ❑ To install the package you must have administrative or 'root' privileges. To install, switch to 'super-user' mode by typing:

```
su root
```

- ❑ Finally, install the package by typing the command:

```
make install
```

*Building the Win32 Native Backend (**Experimental**)*

The 'w32gui' package provides a native Win32 (Windows 95/98/NT) interface for use by the GNUstep GUI library. This library provides the translation from GUI library command into native Win32 drawing commands. This backend does not support Microsoft's DirectX technology insofar as it does not give direct access to these APIs.

To build the w32gui package, follow these steps:

- ❑ Change your directory to a location where the Win32 GUI source can be decompressed:

```
cd /usr/src
```

- ❑ Uncompress the w32gui package into the directory:

```
tar zxf gstep-w32gui-XXXXXX.tar.gz
```

- ❑ Change into the newly created 'w32gui' directory:

```
cd gstep-XXXXXX/w32gui
```

- ❑ Configure the package (with the applicable prefix, if desired) by typing the following:

```
./configure --prefix=/usr/local/gnustep
```

- ❑ Build the package by typing:

```
make
```

- ❑ Build the Testing package by typing:

```
cd Testing
```

```
make
```

- ❑ Return to the parent menu for installation by typing:

```
cd ..
```

- ❑ To install the package you must have administrative or 'root' privileges. To install, switch to 'super-user' mode by typing:

```
su root
```

- ❑ Finally, install the package by typing the command:

```
make install
```

Building the X-Windows GPS Backend (Primary)

The 'xgps' package provides the glue between the X-Windows system that you have (or installed) on your system and the GNUstep GUI library. This library provides the translation from GUI library command into DPS primitive commands, which are then implemented in terms of X Windows commands. As such, it does not provide full PostScript WYSIWYG compatibility as in the 'xdps' back-end, however, it is maturing faster than the 'xdps' back-end and may provide more functionality sooner.

To build the xgps package, follow these steps:

- ❑ Change your directory to a location where the xgps source can be decompressed:

```
cd /usr/src
```

- ❑ Uncompress the xdps package into the directory:

```
tar zxf gstep-xgps-XXXXXX.tar.gz
```

- ❑ Change into the newly created 'xdps' directory:

```
cd gstep-XXXXXX/xgps
```

- ❑ Configure the package (with the applicable prefix, if desired) by typing the following:

```
./configure --prefix=/usr/local/gnustep
```

- ❑ Build the package by typing:

```
make
```

- ❑ Build the Testing package by typing:

```
cd Testing
```

```
make
```

- ❑ Return to the parent menu for installation by typing:

```
cd ..
```

- ❑ To install the package you must have administrative or 'root' privileges. To install, switch to 'super-user' mode by typing:

```
su root
```

- ❑ Finally, install the package by typing the command:

```
make install
```

Special Options for GNUstep Installation

You can specify compilation of alternate libraries by using the `'-with-library-combo'` option. You need to have these libraries unpacked in a subdirectory under the `gstep` directory, then you can use the command:

```
./configure --with-library-combo=gnu-gnu-gnu-xdps
```

to compile with the `xdps` library rather than the `xgps`. Currently the `xdps` library is less stable than `xgps`, so this may be something you may wish to do with caution.

With the various GNUstep libraries you can also use two additional switches: `'shared='` and `'debug='`. The makefiles allow you to build the libraries using the following:

```
make shared=no debug=yes
```

You must make sure you use the same switches for every package you compile, and when you install.

Windows 95 / 98 / NT Specific Installation

- ❑ Installing GNUstep under Windows 95/98/NT requires the pre-existence of a UNIX layer on your Windows box. To achieve this, you must install the Cygwin package available from Cygnus Solutions (<http://www.cygnus.com/>). Once you have installed this package, you may proceed with the installation of GNUstep.
- ❑ Also note that Windows has its own thread package, therefore, if you chose to compile either `'gcc'` or `'egcs'` with a multi-threaded Objective-C runtime, you should use the `'--enable-threads'` option without any `'LIB'` clause. Alternatively, Cygnus Solutions also has an implementation of `'pthreads'` which is currently under development. If you chose to experiment with this package, the `'LIB=posix'` clause would be required.
- ❑ **At this point, you may continue to install the GNUstep libraries by following the directions in the section entitled "Generic UNIX based Installation". Note that you must use the GNUstep special installation option `'-with-library-combo'` to use the Win32 graphical backend. The format of the configuration stage of the `'gstep-core'` command is as follows:**

```
./configure -with-library-combo=gnu-gnu-gnu-w32gui
```

Note that under Windows, all executables must have a `' .exe'` extension. This is achieved by adding the clause `'--program-suffix=.exe'` to the configuration command above.

Solaris Specific Installation

- ❑ There are no specific instructions for installing GNUstep on Solaris. However, please review the **Compiling GCC** section for a potential problem related to the use of 'gcc' on Solaris/Intel, and for information regarding patches for 'gcc-2.8.0' on Solaris/SPARC.
- ❑ Also note that Solaris has its own thread package, therefore, if you chose to compile either 'gcc' or 'egcs' with a multi-threaded Objective-C runtime, you should use the '--enable-threads' option without any 'LIB' clause.
- ❑ **Otherwise, please follow the 'Generic UNIX-based Installation' instructions.**

GNU / Linux Specific Installation

- ❑ Prior to attempting to compile GNUstep's libraries under GNU/Linux (RedHat 5.2 or later), you must tell your system to use the `egcs` compiler by using the following command line argument:

```
$ export CC=egcs
```

- ❑ **Install GNUstep libraries by following the directions in the section entitled "Generic UNIX based Installation".**

GNU / Linux PowerPC (LinuxPPC 4.x) Specific Installation

- ❑ Prior to attempting to compile GNUstep's libraries under GNU/Linux (RedHat 5.2 or later), you must tell your system to use the `egcs` compiler by using the following command line argument:

```
$ export CC=egcs
```

- ❑ The recommended compiler is `egcs-2.90.25 980302` (`egcs-1.0.2 prerelease`). **egcs-1.1.2 does not work**. Also, `gdb 4.16.97` and `glibc-0.961212-1h` are recommended.
- ❑ **Install GNUstep libraries by following the directions in the section entitled "Generic UNIX based Installation".**

FreeBSD 3.x Specific Installation

- ❑ 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'`

- ❑ When using egcs-1.1.2 under FreeBSD, Add the following line at line 301 in target.make ``-u __deregister_frame_info \'`
- ❑ Prior to attempting to compile GNUstep's libraries under FreeBSD 3.x, you must add the JPEG library to your `CONFIG_SYSTEM_LIBS` variable in the file `"/core/make/config.make"` as follows:

```
CONFIG_SYSTEM_LIBS = -ljpeg
```

- ❑ **Install GNUstep libraries by following the directions in the section entitled "Generic UNIX based Installation".**

Unixware Specific Installation

Unixware 2.1.X systems have a couple of extra steps before you can install GNUstep.

- ❑ Tune the kernel to increase the argument space so that GNUstep can pass long command-line argument strings to processes (which the makefiles do):

```
/etc/conf/bin/idtune ARG_MAX 102400
```

- ❑ If you plan to use 'gcc-2.8.1', you must first patch the gcc source code and recompile. The patch involves the file `crtstuff.c` and needs to be applied against Unixware 2.1.3 (and possibly other versions). This patch needs to be applied against the function `__do_global_dtors_aux()`. If you are using the latest version of egcs, then this patch is not needed:

```
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;
}
```

- Once you have applied the patch, recompile 'gcc' as specified in the **Compiling GCC** section, except you should use the following substitution for the configuration of gcc:

```
./configure --with-gnu-as --with-gnu-ld --with-stabs
```

- **Install GNUstep libraries by following the directions in the section entitled "Generic UNIX based Installation".**
- The SIOCGIFCONF flag of ioctl does not work on some versions of Unixware. Therefore, if you encountered difficulties when installing the GNUstep "gstep-base" or "libFoundation" it may be necessary to use the '-a' flag to gdomap to give it the name of a file containing the IP address and netmask information for the network interfaces on the system.
- You will need to set up a file (the suggested filename of which is '/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.
- As an example, 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/Makefiles/GNUstep.sh  
gdomap -a /etc/gdomap_addresses
```

If you do not set gdomap up correctly, the GNUstep Distributed Objects will not work.

Proceed now to the section entitled **"GNUstep Post Installation"**.

GNUstep Post Installation

- If you haven't already done so when installing the Makefile package, you should add the shell script ``GNUstep.sh`` in the Makefile package to your shell startup file (such as ``.profile``). For instance:

```
. /usr/local/GNUstep/Makefiles/GNUstep.sh
```

in your ``.profile`` file will work. It defines environment variables that are needed to find GNUstep files and executables. Users of `csh` need to use the ``GNUstep.csh`` script. Read the `README`` for more info.

- Set up your home GNUstep directory. This is where user defaults are kept, and in the future, other files may be kept there.

```
cd
```

```
mkdir GNUstep
```

- Next, set your local time zone. There are three ways to do this, pick one (see ``${GNUSTEP_SYSTEM_ROOT}/Libraries/Resources/gnustep/NSTimeZones/zones`` for typical time zones):
 - ◆ Use the `defaults` utility to set "Local Time Zone" to your local time zone (defaults is installed with GNUstep in the Tools directory e.g. `defaults write NSGlobalDomain "Local Time Zone" GB``).
 - ◆ Set the `TZ` environment variable.
 - ◆ Create the file ``${GNUSTEP_SYSTEM_ROOT}/ Libraries/ Resources/ gnustep/ NSTimeZones/localtime`` with the name of the local time zone in it.
- If you are using a built-in DPS server, you may need to set the `PSRESOURCEPATH` environment variable (on Solaris, it is set to `'/usr/openwin/lib/X11``).

Compiling Test Tools and Applications

Test tools for the base library are in ``base/checks``, test applications the GUI are in the ``Testing`` directory under these libraries. To make these test tools and applications, just `cd` to the appropriate directory and type `make`. If you have not

already installed the libraries, you will at least need to set `$GNUSTEP_SYSTEM_ROOT` to point to the top-level `gstep` directory. Here's an example:

```
cd xgps/Testing
```

```
GNUSTEP_SYSTEM_ROOT=/home/users/fedor/src/gstep-0.6.5 make
```

Finally, run the tests in the Testing directory. Use the `openapp` utility that is part of the GNUstep makefile package (and stored in ``$GNUSTEP_SYSTEM_ROOT/Tools'`). Usage is:

```
openapp [--library-combo=...] application [arguments to app]
```

Performance Tuning GNUstep Libraries

This section contains instructions specific to machine types, plus other various caveats, hints, and suggestions to help you get the most out of your GNUstep system.

Tuning Objective-C Runtime Debugging Information

Normally, the Objective-C runtime is compiled with debugging information. This can actually makes debugging of user applications frustrating because stepping into a method call will actually cause `gdb` to step into the internal Objective-C method call mechanism, which you don't want most of the time.

The way to avoid this is to compile the Objective-C runtime library without debugging information. Here's how:

```
cd egcs-build-directory/gcc
```

```
rm -f objc/*.o
```

```
make CFLAGS=-O2 libobjc.a
```

```
cp libobjc.a your-egcs-installation-dir
```

This will allow you to debug your code, but not make debugging more confusing by the inclusion of symbols which are not part of your code base.

The GNUstep Tools

Using the GNUstep Tools to Build and Run Applications

gdomap

'gdomap' is the name server for the GNUstep distributed objects system – it handles coordination between processes that use Distributed Objects.

You must have a copy of `gdomap` running on your machine in order to make use of distributed objects - in particular, you need this in order to communicate with the pasteboard server (`gpbs`) and the distributed notification center.

All applications automatically try to register themselves with `gdomap` on startup so that they can receive requests from other applications. These requests might tell the application to activate itself, ask it to open a document, or ask it to provide a service for another application.

Applications also need to use the distributed objects system in order to support GNUstep inter-application operations such as “cut-and-paste” and “drag-and-drop”.

The `gdomap` process must (normally) be run as root - since it listens on a well-known TCP/UDP port (538 is the official Internet Assigned Numbers Authority [IANA] allocated port number) which only root processes are allowed to listen on. It is possible to compile the system to use another hard-coded port - for people who cannot run the server as root. However, all machines running applications that expect to communicate with each other must have copies of `gdomap` running on the same port number.

If the `'make install'` in the `'core/base/Tools'` directory is run as root, the appropriate entries will be added to the `'/etc/services'` file so that `gdomap` will use the correct port number. Additionally, the `gdomap` program will call the `setuid` function to run `gdomap` as root. If `make install` was not run as root, these entries must be added manually.

Ideally, `gdomap` should be placed in your systems startup file so that it will be started up immediately after the standard UNIX network services. As different versions of UNIX vary too much for this to be easily automated, modifying the system startup file is left to the person installing the software.

In order for `gdomap` to work, a machine must have correctly set up TCP/IP networking. Processes will attempt to connect to a copy of `gdomap` on their own machine, using the hostname `'localhost'`, to perform name lookups. At a minimum, the host `'localhost'` must be correctly set up (if you can type `'telnet localhost'` and get a login prompt on your own system, things should be ok).

There are a number of command-line options you can supply to `gdomap`. However, these options are for trouble-shooting and are unnecessary for normal operation. Should you require any of these options, you probably need to be looking at the source code too. You can list these options by typing `'gdomap -H'`.

openapp

`'openapp'` is the shell script used to run applications under GNUstep. For example, to run the ProjectCenter application, a user would type:

```
openapp [--library-combo=...] PC [arguments to app]
```

```
openapp [--library-combo=...] PC.exe [arguments to app] (under  
Windows NT)
```

The `openapp` script searches the GNUstep environment paths to find the application. Once found, it runs the application. Because GNUstep is customized to both the CPU type, and operating system on which it is installed, `openapp` removes the need of the user to know the specific path to the application.

defaults

This is a command-line tool for modifying a users defaults database. In GNUstep preference and configuration information for every program is stored in a database for each user. Normally, this database is modified when a user changes settings in an applications preferences panel. The `defaults` command provides a command-line mechanism for doing the same job.

You can list the options with `'defaults --help'`. This program works like the OPENSTEP program of the same name, but with an additional `'-u name'` option. This allows the system manager to modify the defaults of another user, and with the ability to read in entire databases from standard input.

To display all the global default settings:

```
defaults read NSGlobalDomain
```

To display all settings for `Workspace.app`:

```
defaults read Workspace
```

To set the 'bar' default of the application 'foo' to have the value 'feep':

```
defaults write foo bar feep
```

plparse

Many applications make use of property-list files to store information in a human-readable format that can be edited using a simple text editor. These files must have the correct syntax in order to be read back in - if you edit a property-list file manually, it is quite easy to make little mistakes that leave the file unusable by the applications.

The `plparse` utility will scan a property-list file to check that it is syntactically correct - and display an error message telling you where any problem is.

sfparse

In addition to the property-list format, simpler data structures (simple dictionaries) are often stored in strings-file format consisting of a sequence of key-value pairs. The `sfparse` utility checks that strings-files are syntactically correct and report errors.

make_services

The `make_services` utility is used to scan standard directories (`/usr/GNUstep/Apps`, `/usr/GNUstep/Local/Apps`, `~/Apps`) to make a list of applications and the services provided by those applications. This information is cached so that a GNUstep application knows what services are available from other applications and how to access those services.

The `make_services` utility should be run whenever you login - and should be run after you install a new application.

set_show_service

The `set_show_service` utility allows the user to disable and re-enable a service.

The name of the service must be the name you see on the “Services” menu in your application. If the service is on a submenu of the “Services” menu then its name is the name of the submenu followed by a slash and then the name of the item on the submenu.

Usage:

```
set_show_service --enable foo  
  
set_show_service --disable foo  
  
set_show_service --disable 'Mail/Quote text'
```

Free Software Foundation Order Form

This is an "order form" in the conventional sense: you can print it and use it to send an order to the Free Software Foundation. We would like to accept orders electronically through the World Wide Web, but that will have to wait until there is free software we can use to accept payment while providing proper security for the money (yours and ours). In the mean time, please use paper mail, fax, or voice telephone to place an order.

All items are distributed with permission to copy and to redistribute. Texinfo source for each manual and source for each reference card is on the appropriate CD-ROM; however, our CD-ROMs do not come with printed manuals except when explicitly stated. All items are provided on an "as is" basis, with no warranty of any kind. Please allow six weeks for delivery (though it will not usually take that long).

If you are ordering from Europe, you might find it more convenient and cheaper to order from our European distributor, GNU Distribution Europe, Belgium.

FSF Deluxe Distribution

In the Deluxe Distribution, we compile all the GNU software for your choice of computing platform. Please contact us with any questions about machine, operating system, and media types.

_____ @ \$5000 = \$ _____ The Deluxe Distribution, with manuals, etc.

Machine: _____

Operating system: _____

Media type: _____

Version of X Window System to build: _____

Deluxe Distribution Subtotal \$ _____

CD-ROMs, in ISO 9660 format

GNU Source Code CD-ROM set, Version 14 with X11R6.4, August 1999:

___ @ \$280 = \$ _____ for corporations and other organizations.

___ @ \$ 70 = \$ _____ for individuals.

Subscriptions, next 4 updates, of the Source Code CD-ROM:

___ @ \$840 = \$ _____ for corporations and other organizations.

___ @ \$210 = \$ _____ for individuals.

GNU Compiler Tools Binaries CD-ROM, Version 5, March 1998 Edition:

___ @ \$220 = \$ _____ for corporations and other organizations.

___ @ \$55 = \$ _____ for individuals.

GNU Software for MS-Windows and MS-DOS (book w/CD-ROM

___ @ \$140 = \$ _____ for corporations and other organizations.

___ @ \$35 = \$ _____ for individuals.

CD-ROM Subtotal \$ _____

Manuals

These are the GNU manuals that we publish on paper. Normally, the latest version of each manual will be shipped; please contact us if you want a specific version.

____ @ \$ 30 = \$ _____ GNU Emacs manual, with a reference card
____ @ \$ 60 = \$ _____ GNU Emacs Lisp Reference manual
____ @ \$ 60 = \$ _____ GNU Emacs Lisp Reference, Japanese Edition
____ @ \$ 50 = \$ _____ Using and Porting GNU CC
____ @ \$ 50 = \$ _____ GNU C Library Reference Manual
____ @ \$ 50 = \$ _____ GNU Emacs Calc manual, with a reference card
____ @ \$ 20 = \$ _____ Programming in Emacs Lisp: An Introduction
____ @ \$ 20 = \$ _____ Debugging with GDB, with a reference card
____ @ \$ 25 = \$ _____ GAWK: The GNU Awk User's Guide
____ @ \$ 20 = \$ _____ GNU Make
____ @ \$ 20 = \$ _____ Bison manual, with a reference card
____ @ \$ 20 = \$ _____ Flex manual, with a reference card
____ @ \$ 25 = \$ _____ Texinfo manual
____ @ \$ 15 = \$ _____ Termcap manual

Manuals Subtotal \$ _____

Reference Cards

We normally sell reference cards in packets of ten for \$10. Single copies are \$1.50.

___ @ \$ 10 = \$ _____ GNU Emacs version 20 reference cards

___ @ \$ 10 = \$ _____ GNU Emacs Calc reference cards

___ @ \$ 10 = \$ _____ GDB reference cards

___ @ \$ 10 = \$ _____ Bison reference cards

___ @ \$ 10 = \$ _____ Flex reference cards

Reference Cards Subtotal \$ _____

T-shirts

GNU/FSF T-shirts, thick 100% cotton in sizes ranging from S to XXXL and in a range of colors: black, natural (off-white), blue-green, and burgundy. Please see <http://www.gnu.org/order/t-shirts.html> for sizing and color availability before ordering.

Please list 1st, 2nd, and 3rd choice of color.

___ @ \$ 18 = \$ _____ Size _____

Color choice: 1st _____ 2nd _____ 3rd _____

___ @ \$ 18 = \$ _____ Size _____

Color choice: 1st _____ 2nd _____ 3rd _____

___ @ \$ 18 = \$ _____ Size _____

Color choice: 1st _____ 2nd _____ 3rd _____

___ @ \$ 18 = \$ _____ Size _____

Color choice: 1st _____ 2nd _____ 3rd _____

T-shirt Subtotal \$ _____

Older Items

Older items are only available while supplies last.

_____ @ \$ 5 = \$ _____ GNU Emacs version 18 reference cards, in packs of ten.

Single copies free with order or donation!

_____ @ \$ 15 = \$ _____ GNU Emacs Manual for Emacs version 19.29

_____ @ \$ 10 = \$ _____ Gawk Manual, Edition 0.16 for version 2.16

_____ @ \$ 50 = \$ _____ GNU CC Manual, for GCC 2.7.2

_____ @ \$ 20 = \$ _____ GNU CC Manual, for GCC 2.7.2

Printed in 8.5x11 format with plastic comb binding.

_____ @ \$ 15 = \$ _____ Debugging with GDB for Version 4.14

Older Items Subtotal \$ _____

Older CD-ROMs

Please fill in the number of copies of each older CD-ROM you order:

CD-ROM Title	For Corporations	For Individuals and other Organizations
GNU Compiler Tools Binaries CD-ROM January 1997 (Version 4)		
GNU Compiler Tools Binaries CD-ROM December 1995 (Version 3)		
GNU Compiler Tools Binaries CD-ROM December 1994 (Version 2)		
GNU Compiler Tools Binaries CD-ROM December 1993 (Version 1)		

CD-ROM Title	For Corporations	For Individuals and other Organizations
GNU Source Code CD-ROM February 1999 (Version 13) with X11R6.4		
GNU Source Code CD-ROM November 1998 (Version 12) with X11R6.4		
GNU Source Code CD-ROM March 1998 (Version 11) with X11R6.3		
GNU Source Code CD-ROM Jan. 1997 (Version 9) with X11R6.3		
GNU Source Code CD-ROM June 1996 (Version 8) with X11R6		
GNU Source Code CD-ROM Dec. 1995 (Version 7) with X11R6		
GNU Source Code CD-ROM June 1995 (Version 6) with X11R6		
GNU Source Code CD-ROM May 1994 (Version 4) with X11R6		
GNU Source Code CD-ROM Nov. 1993 (Version 3) with X11R5		
GNU Source Code CD-ROM May 1993 (Version 2) with X11R5		
GNU Source Code CD-ROM October 1992 (Version 1) with X11R5		

Please put the total count and cost of the above older CD-ROMs here:

_____ @ \$ 80 = \$ _____ for corporations and other organizations.

_____ @ \$ 20 = \$ _____ for individuals.

Older CD-ROM Subtotal \$ _____

Tax and Shipping Costs

United States, Canada, and Puerto Rico

- + \$ _____ For addresses in Massachusetts: add 5% sales tax or give tax exempt number. There is no sales tax on T-shirts.
- + \$ _____ Shipping fee for addresses in Alaska, Hawaii, or Puerto Rico:
- \$ 5.00 Base Charge.
- + \$ _____ \$ 5.00 for **each** Emacs Calc or Emacs Lisp Reference
- + \$ _____ \$ 20.00 for **each** CD-ROM subscription (\$20.00 * # of Subscriptions);
- + \$ _____ \$ 1.00 for **each** item other than the above (shipping for all other items = \$ 1.00 * # of Other Items).

Other Destinations

- + \$ _____ Express shipping for most Foreign Destinations: (Please do **not** use this formula for addresses in Bahrain, Jordan, Israel, Kuwait, Lebanon, Saudi Arabia, Oman, Qatar, and the United Arab Emirates. Please fax, or contact us for an exact shipping quote.)
- ◆ \$ 20.00 Base Charge.
 - ◆ \$ 80.00 for **each** CD-ROM subscription (\$80.00 * # of Subscriptions);
 - ◆ \$ 10.00 for **each** item other than the above (shipping for all other items = \$ 10.00 * # of Other Items).
- Air mail: contact fsforder@gnu.org for a quote.*

- + \$ _____ Optional donation (tax-deductible in the U.S.). We suggest 5% if paying by credit card.

GRAND TOTAL \$ _____ We pay for shipping via UPS ground transportation in the contiguous 48 states and Canada. For very large foreign orders, ask about actual shipping costs for that order.

Shipping Information

Name: _____

Mail Stop/Dept. Name: _____

Organization: _____

Street Address: _____

City/State/Province: _____

Zip Code/Postal Code/Country: _____

Telephone number in case of a problem with your order. _____

For international orders, please include a Fax number. _____

E-mail Address: _____

Orders filled only upon receipt of check, money order, or credit card order in U.S. dollars. Unpaid orders will be returned to the sender. We do not have the staff to handle the billing of unpaid orders. Please help keep our lives simple by including your payment with your order.

For orders from outside the U.S.:

You are responsible for paying all duties, tariffs, and taxes. If you refuse to pay the charges, the shipper will return or abandon the order. **Please make checks payable to the "Free Software Foundation". Checks must be in U.S. dollars, drawn on a U.S. bank.**

For Credit Card Orders:

The Free Software Foundation takes these credit cards: Carte Blanche, Diner's Club, Discover, JCB, MasterCard, Visa, and American Express.

Please note that we are charged about 5% of an order's total amount in credit card processing fees. Please consider paying by check instead, or adding on a 5% donation to make up the difference. To place a credit card order, please give us this information:

Card type: _____

Account Number: _____

Expiration Date: _____

Cardholder's Signature: _____

Cardholder's Name: _____

Do not send your credit card number in email! It might be intercepted and used by someone else. Please use telephone, fax, or snail mail to send credit card orders.

If you wish to pay by wire transfer or you are a reseller, please contact us or write us for details. Please mail orders to:

Free Software Foundation
59 Temple Place - Suite 330
Boston, MA 02111
+1-617-542-5942
Fax: +1-617-542-2652

Prices and contents may change without notice.

