GNUstep Filesystem Hierarchy Document

Authors: Tim Harrison, Martin Brecher, Adam Fedor, Nicola Pero, Richard Frith-Macdonald

Table of Contents

GNUstep Filesystem Hierarchy

On GNUstep, there are four separate places where files related to GNUstep are installed: these places are called "domains". These four domains are the System domain, the Local domain, the Network domain, and the User domain. Each of these domains serve a special purpose.

The following is a general overview of the GNUstep domains. A detailed explanation of the directory structure contained within each domain is found later in this document.

The System Domain

The System domain is found in the 'System' folder of the GNUstep installation. This directory contains all files which were included in the default GNUstep installation or distribution. Normally these include the basic GNUstep libraries (Foundation and AppKit), and might include essential system applications (the Workspace Manager, the Editor, applications related to system administrative tasks), the developer applications (Project Center and Gorm, as well as header files), essential extensions (bundles for XML, SSL, RTF, etc), as well as all software installed by the manufacturer of your distribution. These files are usually essential for having a fully functional system. Thus, making modifications to these files is highly discouraged. In addition, only the system administrator ('root' on most UNIX systems) should have permissions to write to that domain.

The Local Domain

While at first glance, the Local domain seems very similar to the System domain, there are several differences between them. The most important thing is the differing purpose of the Local domain, as it is meant as the location for installing software which was not included with your GNUstep distribution and which you or your local sysadmin compile and/or install manually. These may include third party applications, custom extension libraries and their related header files, etc. The Local domain is - as the name suggests - usually installed as 'Local' on your GNUstep system. Every software (except for gnustep-make, gnustep-base, gnustep-gui and gnustep-back which by default install into the System domain) should install by default into the Local domain, so that if you download a source tarball of the software and you install it, it installs by default in the right place for this operation (the Local domain). Distributions should override this default manually when they package the software they want to distribute as part of their distribution, so that in that case the software is installed in the System domain.

The Network Domain

The 'Network' domain is optional and is currently coalesced with the 'Local' domain by default; this is particularly appropriate for use on stand alone systems such as your home workstation. However, the Network domain can be of great use in networked, corporate environments. Its main purpose is to hold files exported from a central server in your network or from other workstations. Most times, remote directories containing applictations or general data used by several workstations in the network are mounted using the Network File System (NFS). Such usage gives administrators the possibility of providing application or resources to a vast number of workstations while only having to manage the software in one place. This is especially useful when workstations are used by several users with different tasks and requirements. If you want to take advantage of the 'Network' domain, you need to enable it when you configure gnustep-make.

The Users Domain

On systems where GNUstep is installed optionally, the Users domain can usually be found in a subdirectory of the user's home directory called 'GNUstep'. This location is configurable, and some installations may put this directly in the user's directory (and typical user's home directories would be located in a 'Users' folder). As the name suggests, the main purpose of the Users domain is to hold GNUstep related files which shall not be available to other users on the system but only to the user owning them. This includes the GNUstep defaults database, which holds system settings, application preferences and customized resources, as well as temporary data related to services and file type associations for programs. It also holds user installed applications and tools (each user has the ability to install their own version of an application or tool). In addition to these special files, the User domain features the same structure as the other domains.

Hierarchy

System

```
System/
        Applications/
        Library/
        Tools/
        share/
```

Local

```
Local/
        Applications/
        Library/
        Tools/
```

Network

```
Network/
        Applications/
        Library/
        Tools/
```

User

```
User's GNUstep root dir/
        Applications/
        Library/
        Tools/
```

Library Folder

ApplicationSupport		
Bundles		
ColorPickers		
Colors		
Defaults		*user only*
DTDs		
DocTemplates		
Documentation		
Fonts		
Frameworks		
Headers		
Images		
KeyBindings		
Libraries		
	Resources	
	gnustep-base	*system only*
		.lproj folders
		CharacterSets
		TimeZones
		Languages
	gnustep-gui	*system only*

                                              .lproj folders
                                              TextConverters
Makefiles   *system only*
                    Additional
PostScript
                    PPD
Services
Sounds

Description

System

   The System directory is the location of the GNUstep makefile package,
base, gui and backend libraries, and any accompanying software that is dis-
tributed as part of whatever distribution of GNUstep you are using. This
directory MUST exist for a proper installation of GNUstep.

   Using the –prefix option to the configure script in gnustep-make, an in-
stallation of GNUstep may be placed wherever the installer wishes; the Sys-
tem, Local (and optionally Network) domain by default will be subdirectory
of this root location.

   Common options are:
```
 /usr/GNUstep
 /usr/local/GNUstep
 /opt/GNUstep
 /
```
All directories referenced in this document are relative to this root loca-
tion.

Local

   The Local domain is the location of libraries, frameworks, bundles, and
supporting files for locally installed applications or tools that are not dis-
tributed as part of the distribution that you are using, but that are compiled
and/or installed manually by you or your sysadmin. This directory MUST
exist for a proper installation of GNUstep.

Network

   The Network Domain is the location for all exported applications, re-
motely mounted filesystems, and remote home directories for users made
available via directory services. It is optional, and disabled by default.

Applications

The 'Applications' directory contains applications. Applications are programs that typically have a GUI interface and contain associated resource files, such as images, localization files and other program elements.

Important applications which are part of GNUstep and which are often distributed as part of a core GNUstep distribution (and so installed in the 'System/Applications' folder) include:

```
Gorm.app
ProjectCenter.app
GSDefaults.app
GWorkspace.app
Preferences.app
```

Tools

The 'Tools' directory contains tools and executable scripts. Tools are programs which generally have a command-line interface. Most are not meant to be used by the average user.

Tools that are written in languages other than Objective-C, or are developed to work with other runtime environments may have their own directory within the Tools directory (for example: 'Tools/Java').

share

The share directory is used for configuration and installation of the core GNUstep libraries and any additional libraries that need configuration information. It is used by the configure (autoconf) program.

Library

The 'Library' directory contains most of the functional code of the GNUstep Development Environment.

The primary reason for the structure of folders within Library is to keep a complimentary structure throughout all domains. This allows easier development, by keeping a standard directory layout, providing developers with a relatively common hierarchy to work within.

ApplicationSupport

This directory contains bundles and other resources that are provided for an application, but that are not specifically distributed with that application. For instance, these may be third-party resources for an application.

For example, modules for the Preferences application may be stored here, in a 'Preferences' subdirectory.

Bundles

This directory contains bundles. Bundles are collections of executable code and associated resources that may be loaded at runtime by an application or tool. Note: this directory is depreciated. Use ApplicationSupport to install bundles that can be used by an application.

ColorPickers

This directory contains bundles that are used by the color picking system. They may include code that implements picking colors from a color wheel, a custom defined list of colors, etc.

Colors

This directory contains files that define specific color mappings for use within libraries or applications that require color definitions.

Defaults

This directory contains files that store defaults for applications, e.g. user preferences. This directory only exists in the User domain.

DTDs

This directory contains any Document Type Definitions required for document parsing.

DocTemplates

This directory contains text templates for automatic documentation, as generated by autodoc. Any additional documentation template types must be placed in this directory, as a central location for documentation template types. Any templates installed within this directory must have an extension indicating what type of documentation system it is referenced by (ie. .gsdoc for the GNUstep implementation of autodoc).

Documentation

This directory contains documentation for libraries, applications, etc.

Fonts

This directory contains fonts and files for organizing font information.

Frameworks

This directory contains frameworks. Frameworks are a type of bundle, which include, within their directory structure, a shared library providing a specific functionality (or group of related functionalities), and all resources required by that shared library.

All frameworks must have the extension '`framework`', to indicate their usage.

Use of frameworks is generally discouraged, as it is difficult to support them in a clean way on multiple platforms. Bundles are a better method of organizing shared collections of resources and code.

Headers

This directory contains header files associated with a library located in the Libraries directory.

Images

KeyBindings

Libraries

This directory contains libraries. (NOTE: The Libraries directory being in Library may sound somewhat redundant, however, it could be read as "a Library of shared libraries".)

Makefiles

This directory contains the different types of makefiles used by the GNUstep development environment to build applications, libraries, bundles, etc. These makefiles are usually included in a project specific GNUmakefile, which is used to build a project under the GNUstep development environment.

This directory only exists in the System domain.

PostScript

This directory contains directories for specific PostScript document types and definitions, allowing applications written using the GNUstep development environment to display PostScript documents, or communicate with printers using PostScript.

Services

This directory contains bundles that are specifically built to provide functionality between different programs (for example, spell checking, creation of a note from text within an email application). Services that are installed on the system must an extension of ".service".

Sounds

This directory contains sound files.

Configuration

The locations of the various domains within the filesystem hierarchy on your machine are determined by the GNUstep configuration file (or if that is not present, by default values built into the GNUstep make and base packages when they were configured and built).

The location of the GNUstep configuration file is built in to the make and base packages when they are configured using the –with-config-file option to the configure script. The path specified must be an absolute one for the make package, but may also be a path relative to the location of the base library itsself (as dynamically linked into applications) for the base package.

However, the location of the configuration file may also be specified using the GNUSTEP_CONFIG_FILE environment variable, overriding the value built in to the package, at any time when using the make package to build or install software. Support for the environment variable may also be enabled for the make package when its configure script is run.

Please do

```
configure --help
```

to get all the options of how the package configuration scripts can be told to set up the default path to and values in the configuration file.

File Format

By default, the configuration file is called GNUstep.conf and exists in /etc/GNUstep on a Unix-like system or C:\GNUstep on an ms-windows

system. In either case this file is in a format suitable for being 'sourced' by the standard unix (Bourne) shell, consisting of lines of the form key=value, comments (everything on a line from the first hash (#) onwards), or blank lines.

This is very convenient on unix-like systems, but needs care for windows users. If a value contains whitespace or backslash characters (or the hash which would start a comment) it needs to be quoted by enclosing the whole value in single or double quotes. An alternative for values containing back-slashes (the norm for a windows path) is to double up each backslash in an unquoted value.

Windows (MINGW)

On ms-windows, for software development, you are likely to want to have an extra configuration file. This is because of the limitations of the make program (used to build and install software).

Basically the issue is that the make package doesn't really like the colons and backslashes in windows paths (using them is error prone) and can't tolerate whitespace in file names ... so you need to set up a config file which uses unix-style paths as used by MSYS (ie of the form '/c/...' rather than 'C:\...') for building and installing software.

On the other hand, the base library (and all applications since they are built using it) wants to work with native windows paths so that applications behave naturally as far as the end users are concerned, and therefore needs a configuration file containing windows-style paths rather than unix-like ones.

The simplest way to achieve this is to use different values for the –with-config-file= option when configuring the make and base packages.

For example, configure the make package like this -

```
./configure --with-config-file=/c/GNUstep/GNUstep.conf-dev
```

and the base library like this -

```
./configure --with-config-file=C:\\GNUstep\\GNUstep.conf
```

Then you need to edit the two config files to make sure they contain values of the correct format.

File Values

The key=value pairs permitted in the configuration file are limited to the keys described here.

GNUSTEP_SYSTEM_ROOT

The value for this key is the path for the System domain.

It must be either an absolute path or a path beginning './' in which case the leading './' is replaced by the path to the directory in which the configuration file is located.

This is normally -

    GNUSTEP_SYSTEM_ROOT=/usr/GNUstep/System

GNUSTEP_LOCAL_ROOT

The value for this key is the path for the Local domain.

It must be either an absolute path or a path beginning './' in which case the leading './' is replaced by the path to the directory in which the configuration file is located.

This is normally -

    GNUSTEP_LOCAL_ROOT=/usr/GNUstep/Local

GNUSTEP_NETWORK_ROOT

The value for this key is the path for the Network domain.

It must be either an absolute path or a path beginning './' in which case the leading './' is replaced by the path to the directory in which the configuration file is located.

This is normally the same as the value configured for GNUSTEP_LOCAL_ROOT -

    GNUSTEP_NETWORK_ROOT=/usr/GNUstep/Local

GNUSTEP_USER_CONFIG_FILE

This is the name of the user specific configuration file (a path relative to the home directory of the user). This configuration file is loaded after the main GNUstep configuration file, and the key=value pairs in this file will override those in the main file (all except the GNUSTEP_USER_CONFIG_FILE obviously).

The default (unless otherwise configured) is

    GNUSTEP_USER_CONFIG_FILE=.GNUstep.conf

If the value of this key is set to an empty styring, the system will not attempt to load a user specific configuration file, thus ensuring that the individual user cannot override the values specified by the system administrator.

GNUSTEP_USER_DIR

This is used to define the User domain ... it is a path relative to the home directory of the user in which the User domain and hence all resources in the User domain are to be found.

The default (unless otherwise configured) is:

```
GNUSTEP_USER_DIR=GNUstep
```

but setting an empty string for this value will cause the User domain to be the home directory of the user.

GNUSTEP_USER_DEFAULTS_DIR

This is used to define the location of the defaults database for the user ... it is a path relative to the home directory of the user.

The default (unless otherwise configured) is:

```
GNUSTEP_USER_DEFAULTS_DIR=GNUstep/Defaults
```