/usr/local - Architecture

Version 1.2

2/17/2002

DAS Document #12229

Modification History

The table below documents all versions and revisions to the document, most recent at the top.

Version	Date	Author	Description
1.2	2/17/2002	Forrest Girouard	Change location of platform dependent files. Update configuration files.
1.1	7/18/2002	Forrest Girouard	Added one new requirement. Added appendix A. Enhanced Figure 1.
1.0	7/16/2002	Forrest Girouard	Initial version

Reviewers

These were the required reviewers of the most recent version of the document:

Mark Abbott
Maurice Meyer

Table of Contents

1 Introduction

1.1 Scope and Goals

This document describes an architecture for a stable, well behaved and controlled `/usr/local` file system. This description covers the organization of the files under `/usr/local`.

It does not specify how /usr/local is made available on any particular host (for example, it might be NFS mounted or a copy may be available on a local disk).

This scheme is not intended to handle very large or active software packages like Oracle or InterMail.

Furthermore, this document will describe some mechanisms, tools, and conventions that should be used with this scheme.

1.2 Requirements

- All installed files in `/usr/local` **must** belong to a versioned package. In other words, every file in /usr/local and it's subdirectories must be part of a known package and the actual files must be stored below a subdirectory whose name contains the version.
- Platform dependent files **must** be handled in a convenient and obvious way. The scheme **must** support multiple platforms.
- It **must** be possible to access a complete `/usr/local` without specifying any packages or versions. In other words, there must be a default version of every package available.
- It **must** be possible to specify versioned packages explicitly.
- It **must** be possible to have multiple versions of any given package installed and usable simultaneously.
- Platform independent files **must** be shared between platforms.
- It **should** be possible to migrate an existing `/usr/local` into the new organization without affecting the people or systems currently using it with the exception of some small amount of downtime.
- Since every file installed in `/usr/local` must belong to a package tools **should** be provided to create simple packages.

1.3 References

See the online man pages for descriptions of commands like `uname`.

1.4 Glossary

LD_LIBRARY_PATH	Special environment variable containing a colon separated list of directories, which is searched when looking for shared objects.
MANPATH	Special environment variable containing a colon separated list of directories that is searched when looking for man pages.
NFS	Network File System
OS	Operating System. The value of OS is defined as the value of the following command: `uname –s | tr ‘[:upper:]’ ‘[:lower:]’`
OS Release	Operating System Release. The value of OS Release is defined as the value of the following command: `uname –r`
PATH	Special environment variable containing a colon separated list of directories that are searched when executing commands.
platform	A platform is defined as a valid pair of OS and OS Release.
shar	A shell archive created using the `bundle` command available from the `misc` package.

2 Usage

For most users when `/usr/local` is made available on any given host they can put `/usr/local/bin` on their **PATH**, `/usr/local/lib` on their shared object path (for example, on `sunos` this would be the **LD_LIBRARY_PATH** environment variable), and `/usr/local/man` on their **MANPATH**. These directories will contain the default set of packages.

For building products or components only the explicitly versioned directories should be used. Furthermore, it is possible to create and maintain custom subsets of `/usr/local` for particular components or products.

3 Architecture

The most important directory in the overall `/usr/local` organization is `/usr/local/lib`. Each subdirectory in `/usr/local/lib` consists of two hyphen separated tokens. The first is the common or standard name of the package (for example, `misc`). In some cases this may be shortened in some obvious way. The second token is the version of the package (for example, `1.13`). If the currently available archive of the package does not contain a version then one is created (for example, `0.1`). So, for the `1.13` version of the `misc` package the following directory would exist and contain the installed files: `/usr/local/lib/misc-1.13`.

A symbolic link is created in `/usr/local/lib` that points to the default version of the package. For example, `/usr/local/lib/misc -> /usr/local/lib/misc-1.13`.

Each of the files under the versioned package directory is linked into the corresponding platform specific directory under `/usr/local`. For example, the `misc` package contains the following subdirectories: `bin`, `include`, `lib`, `man and whatever platform specific directories have been built`. Each of these directories contains platform independent files except for the platform specific subdirectories which contain the platform specific files. For the `misc` package the `bin` directory contains the following executables: `ltee`, `stail`, `pslabel`, and `bundle`. The platform specific subdirectories contain a bin directory with the platform specific executables. For example, `/usr/local/lib/misc-1.13/ sunos/5.8/bin` contains the following executables: `catport`, `slock`, `cleanenv`, `printenv`, `appendpaths`, `gmtime`, `timegm`, `locallog`, `purgepath`, and `ttyraw`. Finally, under `/usr/local` platform specific directories are created which contain links to both the platform independent and platform specific files in the versioned package directories under `/usr/local/lib` as appropriate (see the Figure 1).

To accomplish this organization the following conventions should be used. The downloaded package archive is copied into `/usr/local/src/dist`. It is unpacked into `/usr/local/src/`*platform* if the downloaded archive unpacks everything into a single directory and, if not, a directory is created and the archive unpacked into it. For example, `/usr/local/src/dist/misc1.13.shar.gz` does create a directory so one is created, `/usr/local/src/sunos-5.8/misc-1.13,` and the shar bundle is unpacked there. A few files are maintained per package in order to capture the build environment and to create the necessary directories and symbolic links. A package specific configuration file is created (nominally called *`package-version`*`.conf`) for each version of each package where settings common across all packages are saved in a common configuration file (nominally called `usrlocal.conf`). A second package specific file is created (nominally called *`package-version`*`.ln`) to create all the symbolic links. Optionally, a third package specific file is created (nominally called *package-version*.make) which is used to configure and build the package when the *gnu-default.make* is insufficient. All these configuration files should be kept in `/usr/local/src/etc`. So using the `misc` package as an example the following files are created: `/usr/local/src/etc/usrlocal.conf`, `/usr/local/src/etc/misc-1.13.conf`, `/usr/local/src/etc/misc-1.13.make, usr/local/src/etc/misc-1.13.ln` (see the appendixes for the contents of these files).

A partial layout of the `misc` package according to this scheme looks as follows:

/usr/local - Architecture

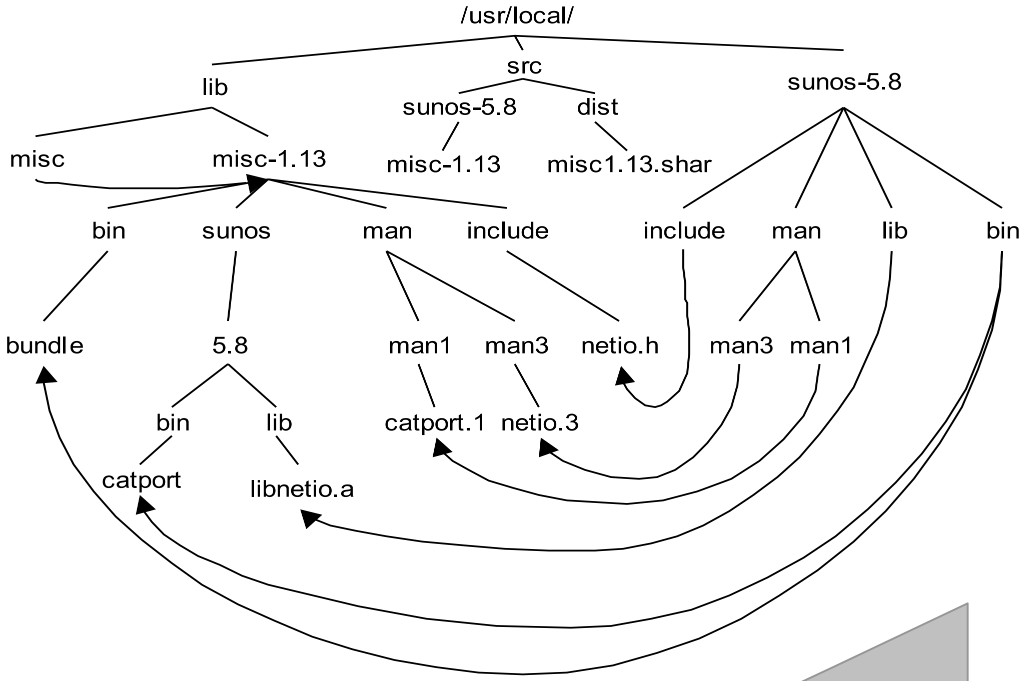

Figure 1

The `/usr/local/sunos-5.8` directory will be made available on a SunOS 5.8 host as `/usr/local`.

4 Assumptions & Limitations

If two packages contain a file with the same name then only one package can be completely installed.

5 Open Issues

Should provide a tool that can easily switch between versions of packages and which understands the platform issue.

Should provide tools to create and manage the configuration files.

The ability to support the platform specific directories is dependent on whether the package itself manages platform distinctions and, if so, whether or not it exposes a means of specifying them when the package is configured and/or built. Typically, GNU packages support distinct *prefix* and *exec-prefix* options to their configure tools.

6 Summary

A similar scheme has been in partial use on the `sv-devsun` hosts for the last several years and prior to that this scheme was successfully used in a 24x7 satellite operations center.

More recently this specific scheme has been successfully implemented and used by a number of developers across several different products.

7 Appendix A: Sample configuration files

These example configuration files are considered non-normative.

Here is a sample of a usrlocal.conf file:

```
LOCALROOT=/net/re-build/build4/local
USRLOCAL=/usr/local
DISTDIR=${USRLOCAL}/src/dist
ETCDIR=${USRLOCAL}/src/etc
MAKEFLAGS=e
LOCAL=/usr/local
OSNAME=`/usr/bin/uname -s | /usr/bin/tr '[:upper:]' '[:lower:]'`

. ${ETCDIR}/${OSNAME}.conf

CONFPATH=${CONFPATH-${USRLOCAL}/src/${OSPLATFORM}/etc:${ETCDIR}}
CONFSHELL="${USRLOCAL}/${PLATFORM}/bin/cleanenv -/bin/sh +getenv USRLOCAL
+getenv DISTDIR +getenv ETCDIR +getenv OSNAME +getenv OSVERSION +getenv
PLATFORM +getenv OSPLATFORM +getenv SRCPLATFORM"
```

Here is a sample of a sunos.conf file:

```
OSVERSION=`uname -r`
PLATFORM="${OSNAME}/${OSVERSION}"
OSPLATFORM="${OSNAME}-${OSVERSION}"
SRCPLATFORM=$USRLOCAL/src/$OSPLATFORM

AR=/usr/ccs/bin/ar
INST=/usr/ucb/install
#CC=/nfs/local/SUNWspro62/SUNWspro/bin/cc
STATIC=-Bstatic

mkdir -p $USRLOCAL/src/$OSPLATFORM
mkdir -p $USRLOCAL/$PLATFORM
( cd $USRLOCAL/$PLATFORM ; \
  mkdir -p bin etc include info lib/X11/app-defaults libexec sbin share \
        man/man1 man/man3 man/man4 man/man5 man/man7 man/man8 man/manl \
        man/mann )
```

Here is a sample of the misc-1.13.conf file:

```
 PACKAGE=misc
VERSION=1.13
INSTALLDIR=$USRLOCAL/lib/$PACKAGE-$VERSION
LINKDIR=$USRLOCAL/lib/$PACKAGE
SRCDIR=$SRCPLATFORM/$PACKAGE-$VERSION
DIST=$DISTDIR/$PACKAGE$VERSION.shar.gz
STCSRC='hp_dysize.c hp_timegm.c strptime.c'
LIBNETIO='-lnetio -lsocket -lnsl'
LIBTTYRAW='$(STATIC) -L/usr/ucblib -lucb'
SUNOSRELEASE=`uname -a | awk '"SunOS" == $1 {print $3}' | tr -d '.' | cut -
c-2`
SUNOS=-DSUNOS=$SUNOSRELEASE
```

Here is a sample of the misc-1.13.make file:

```
DIFF='280a
$(INSTALL)/$(PLATFORM)/lib/% : %
        $(INST) -m 644 $< $(INSTALL)/$(PLATFORM)/lib

$(INSTALL)/$(PLATFORM)/bin/% : %
        $(INST) -m 755 $< $(INSTALL)/$(PLATFORM)/bin

.
275,277d
221a
        $(PLATFORMBIN:%=$(INSTALL)/$(PLATFORM)/bin/%) \
.
219c
        $(PLATFORMLIB:%=$(INSTALL)/$(PLATFORM)/lib/%) \
.
196,198c
LOCALBIN = $(SCRIPTS)
PLATFORMBIN = $(PROGS)
PLATFORMLIB = $(LIBS)
LOCALMAN1 = $(LOCALBIN:%=%.1) $(PLATFORMBIN:%=%.1)
.
'
rm -rf $INSTALLDIR/$PLATFORM $PACKAGE-$VERSION.out
if [ -d $SRCDIR -a ! -f $SRCDIR/Makefile.diff ]
```

```
then
    ( cd $SRCDIR
      chmod +w Makefile
      echo "${DIFF}" > Makefile.diff
      patch -i Makefile.diff Makefile
      mv Makefile Makefile.gnu
      sed -e 's/^\([A-Z][A-Z]*\)[       ]*+= /\1 = /' \
      < Makefile.gnu \
      > Makefile
      echo "catport.o : netio.h" >> Makefile
      echo "timegm.o : strptime.h" >> Makefile
    ) 2>&1 | tee $PACKAGE-$VERSION.out
fi
if cd $SRCDIR
then
    if [ -f ../$PACKAGE-$VERSION.out ]
    then
        mv ../$PACKAGE-$VERSION.out .
    fi
    ( INSTALL=$INSTALLDIR ; export INSTALL
      mkdir -p $INSTALLDIR/bin
      mkdir -p $INSTALLDIR/$PLATFORM/bin
      mkdir -p $INSTALLDIR/$PLATFORM/lib
      mkdir -p $INSTALLDIR/include
      mkdir -p $INSTALLDIR/man/man1
      mkdir -p $INSTALLDIR/man/man3
      PATH=${PATH}:/usr/ccs/bin ; export PATH
      make
      make install
    ) 2>&1 | tee -a $PACKAGE-$VERSION.out
fi
```

Here is a sample of the misc-1.13.ln file:

```
rmlns
rm -f $LINKDIR
ln -f -s $INSTALLDIR $LINKDIR
ln -f -s $LINKDIR/include/* $PLATFORMLOCAL/include
ln -f -s $LINKDIR/bin/* $PLATFORMLOCAL/bin
ln -f -s $LINKDIR/bin/$PLATFORM/* $PLATFORMLOCAL/bin
ln -f -s $LINKDIR/lib/$PLATFORM/* $PLATFORMLOCAL/lib
ln -f -s $LINKDIR/man/man1/* $PLATFORMLOCAL/man/man1
ln -f -s $LINKDIR/man/man3/* $PLATFORMLOCAL/man/man3
```