
Software Installation On Large Systems

by Paul Anderson

Department of Computer Science
University of Edinburgh, U.K.
<paul@dcs.ed.ac.uk>

Introduction

The *Software Installation Workshop* at the 1992 LISA conference¹ was organised to discuss some of the problems of installing and configuring third-party software in large heterogeneous installations. Most current third-party software packages provide their own installation procedures which often seem to be designed primarily for use on small, single-vendor systems. In large installations, these procedures can be totally inappropriate, causing considerable difficulties for system administrators; in extreme cases, it may even be impossible for a particular site to configure a package in any useful way at all, limiting the choice of available software and affecting purchasing decisions.

The workshop attempted to identify the main source of these difficulties and give some hints and recommendations for improving software installations so that they are more compatible with large systems.

Software Installation and Configuration

The Posix draft P1003.7.2 proposes a standard software distribution layout and describes an installation process consisting of several stages: These stages cover the identification of files to be installed (*selection*), verification of certain prerequisites (*analysis*), and copying of the files onto the system (*load*). The proposal provides many welcome features, such as the ability to de-install a package and perform remote installations.

During the *load* phase, provision is made for vendor-supplied scripts to be executed with the intention that these scripts should configure the software package itself for the environment. Changes to the actual environment (such as

modifications to system files) are expected to occur in a further stage (*configuration*), although details of this are explicitly excluded from the standard. These configuration issues which are not covered by the Posix standard are precisely those areas that frequently cause difficulties when installing software packages onto large heterogeneous systems. The following sections describe some of the issues which were identified as significant problems at the workshop.

Location of files

The single largest cause of difficulties with software installation, is probably due to a lack of flexibility in configuring the pathnames used by the package. Different vendors use significantly different pathnames in their basic operating systems, and most large multi-vendor installations have their own standard locations for various types of files.

Current installation procedures typically specify some fixed absolute pathnames for some, or all, of the files used by the package. At best, this can involve extra work and confusion because the supplied pathnames do not conform to the normal site conventions. At worst, one or more of the pathnames will conflict with some other package or will be on an inappropriate filesystem - for example, `/usr/local/lib` is often mounted read-only, in which case it is not a suitable place for log files or other writable data. The Posix standard provides an option for a package to be *locatable* - ie. to specify an alternative root directory for the installation. This obviously avoids the worst of these problems, but it is only optional, and it is barely sufficient for large installations where it is most useful if different pathnames can be specified for different categories of files. For example:

- In a heterogeneous network, it is very useful to separate out those files which are architecture-dependant and those files which can be shared between architectures.
- Files which need to be writable must be separated from read-only files which are frequently replicated and stored on read-only

¹ Usenix: 6th Large Installations Systems Administration Conference, October 1992, Long Beach, California.

filesystems.

- Files which need to be private to a particular workstation should be separated from files which would be common across the whole network.
- The directory from which the package will finally run is sometimes different from the directory into which it is initially installed. This is the case, for example, if a package is installed into `/home/package` but subsequently distributed to a different location on multiple servers (typically using `rdist`).

Installation programs which use `pwd` to determine the current directory and subsequently configure this into the software are particularly troublesome because they usually do not work in conjunction with the automounters which are common in large NFS installations.

File ownership and security

Large installations are generally more sensitive to file ownership and permission issues, due to the increased scale, and the importance of security. This means that it is important that files are not installed with inappropriate usernames or protections (for example, as a result of assumptions about the default `umask` at installation time). At runtime, packages should also use `setuid` or `setgid` to some normal user in preference to `setuid root` wherever possible.

As with the choice of pathnames, a large installation is also more likely to suffer from conflicts between user and group names if a package insists on being installed as a particular user or group (the same is true of user and group IDs). It is usually quite reasonable for a package to be installed with its own username, but it must be possible to override the default name (and ID) if required.

During the installation itself, system administrators will be very reluctant to run processes as `root` unless this is absolutely essential and their consequences are well understood. A dummy installation option, which simply shows the actions that a real installation would perform, was suggested as a useful way of checking the consequences of such a procedure. In most cases, however, the bulk of the installation process should be capable of running under normal user permissions - this is sufficient, even for installation of files in public areas (such as `/usr/local/bin`) with careful use of group access permissions.

Changes to the system configuration

In some cases, configuration of a software package involves modification of critical system files, such as:

- Changes to kernels and device drivers.
- Changes to `inetd.conf`.
- Changes to password and host databases.
- Changes to `rc` files.

On a small system, it is often possible for the configuration process to make the necessary modifications automatically, but this is rarely successful in a large system, due to conflicts with existing customisations, or the use of completely different procedures (eg. NIS instead of a local password file). The Posix standard does not attempt to address this issue at all and there seems to be no good multi-vendor solution to the problem of making safe changes to the system configuration. At present, most system administrators would probably prefer to supervise such critical operations manually, even though this is time consuming and frequently requires special solutions, when many different machines are involved.

Some common difficulties include:

- Distributed authentication schemes such as NIS and Kerberos which mean that the local password file, if any, may not contain the expected information. Attempts to read (or, even worse, edit) the password file are unlikely to be successful on a large system.
- Similarly, the use of DNS or NIS means that there may be no valid local host file.
- Automounters are virtually standard in large NFS installations and may affect the apparent contents of directories and the pathnames returned by `pwd`. Even simple symbolic links in place of directories have been known to cause problems with some installation software.

Some other problems

A number of other problems that have been encountered during software installation are due to a lack of "network awareness" by the installation procedure. For example:

- The installation may well be performed remotely and might not have the terminal type, or window system that would be expected on a standalone machine of the appropriate type.

- Node-locked software licensing is usually inappropriate in big networks and the mechanisms used to implement some schemes simply do not work in a network context. Even network-based license managers do not address the needs of large networks, such as robustness against the failure of a single license server.

Conclusions

The Posix 1003.7.2 standard goes some way towards alleviating the problems of installing software onto large distributed systems. However, there are many configuration issues that are not addressed by the standard and an awareness of the special needs of large systems is necessary when designing software installation and configuration procedures.

Further information

The mailing list `soft-managers@nas.nasa.gov` was formed at the LISA workshop for continuing discussion of the above issues (mailing list subscription requests to `soft-managers-request@nas.nasa.gov`).

Draft copies of the Posix 1003.7.2 standard are available for anonymous ftp from `dcd-mjw.fnal.gov` and the mailing list `7bsm@ui.org` is used for detailed discussions of the drafts (mailing list subscription requests to `7bsm-request@ui.org`).

Acknowledgements

Thanks to all the participants at the LISA workshop and those who joined in subsequent discussions on the mailing list. In particular, thanks to John Stewart for managing the list, and Matthew Wicks for clarifying the Posix standard.