Chapter 1

General Considerations

1.1 Overview

Before rolling the tape onto a disk, there are some general issues which it is useful to consider. If you are planning to install PP on more than one machine, you should read the latter parts of this chapter carefully.

1.2 General Requirements

1.2.1 Systems

PP is designed to be portable to a wide range of UNIX[1] systems. Configurations are given for the following systems:

- Sun 3 using SunOS 3.X.
- Sun 3 using SunOS 4.X.
- Sun 4 using SunOS 4.X.

It is also known to run on the following systems:

- Vax Ultrix.
- Mips Ultrix.
- CCUR 6000 RTU 6.0
- ICL DRS/6000
- NeXT
- Olivetti LSX-3020
- Pyramid 9800 and MIS

[1] UNIX is a trademark of AT&T Bell Laboratories.

1.2.2 Disk space

A fair amount of space must be made available for the system. The figures given below are only a guide.

The basic distribution tape contains sources amounting to around 7Mb.

You will need approximately 40Mb of headroom extra to compile everything, although you can do it in stages if you do not have that much space.

The installed binaries will take up somewhere around 20-30Mb.

The routing table files plus local tables may take up about 1Mb if you have a lot of connectivity.

The *dbm* database built from these tables, perhaps another 1-4Mb.

Of course you must also allow plenty of space for the mail passing through!

This suggests that you should allow about 50Mb of space for building everything, and when built, about 25Mb to run the system with whatever room you feel reasonable for spooling of messages (phew!).

This figure is less if your version of ISODE can make use of shared libraries (e.g., SunOS).

1.2.3 Communications Requirements

PP uses OSI/ROS for its internal communication, using the ISODE implementation of ROS and the lower layers[Ros89]. It is assumed that ISODE is installed and available on the system in question. ISODE *ASN.1* tools are also required to build parts of PP. This version of PP assumes ISODE version 7.0.

1.2.4 X Window System

The system has a very useful monitoring tool which is written using the X Window System[2] which is used for managing the MTA as a whole. It is possible to run an MTA without the console, but not so easily. The console depends on X11 release 4 (or greater) and the Athena Widget set. It will not work under earlier releases of X11.

There are also present some small programs to do X-based alert of new mail. These programs have the same X requirements.

All binaries should work against X11 release 3 (or greater) servers.

1.2.5 QUIPU

This release requires the *libdsap.a* library, but has no requirements on the QUIPU X.500 directory service agent. There is a list channel that utilises the directory but this is optional. Also the X based alert utility has the ability to fetch photographs of the user sending you mail from the directory. This feature can be disabled though. Future releases of PP will require a QUIPU DSA.

[2]The X Window System is a trademark of the Massachusettes Institute of Technology

1.3 Specific Channel Requirements

Some of the channels in PP have external dependencies.

SMTP: The Simple Mail Transfer Protocol channel requires the use of TCP/IP. If the domain name server is to be used, then the BIND[Dun] resolver library must be provided (see Table 2.1 on page 6).

JNT Mail: This is integrated with the UNIX-NIFTP package, available from Cambridge (UK) at the following address:

```
Piete Brooks (pb@cl.cam.ac.uk)
Cambridge University Computer Laboratory,
Cambridge
ENGLAND
```

X.25 access is usually required for this channel.

X.400: This uses the ISODE lower layers (both 1984 and 1988 variants). For most services X.25, CONS, or CLNP is required to run this channel, although it can be run over TCP/IP using RFC-1006[RC87].

UUCP: This channel requires a working UUCP installed on the given machine.

DECNET: This requires the Sunlink DNI (version 7.0 or later) package.

1.4 User Agents

This version of PP does not support any sophisticated user agents. It provides a mail-sending interface of the same approximate functionality of */bin/mail*. It is strongly recommended that the MH[Ros86] user agent is considered for use with PP. MH should be configured with the SMTP transfer option which will connect to PP though the SMTP protocol (configuration option `mts send/smtp`). This also allows PP to be accessed from hosts not running PP directly but with SMTP connectivity.

Warning: MH should be compiled without the `BERK` option.

1.5 Compatibility with Older MTAs

PP is designed as a replacement for *sendmail*(8) and MMDF. As such it is capable of the majority of functions of both (and a great deal more). However, there are some differences.

1.5.1 MMDF

PP does not support the MMDF syntax that allows users to give an address as `user=string@host.domain`.

1.5.2 Sendmail

PP does not deliver to sendmail style mailboxes by default. This can be changed however (see Section 4.12).

Some programs that send mail have compiled into them the *sendmail* program. To deal with these, a fake sendmail is provided that emulates most of the common sendmail options.

As PP does not have sendmail's general rewriting rules, it can not in general be programmed to "fix up" strange addresses. This applies particularly to address formats such as DECnet and Xerox-style addresses. To handle this sort of address an inbound channel should be written that can parse these addresses and convert them to RFC-822 addresses.

1.6 MTA Layout

Usually PP will support sites with many hosts. It is possible, but not desirable, to run a copy of PP on each host. An MTA is defined by the existence of a queue (spool area). Each queue should be managed (in human terms) in order to guarantee good service.

For this reason, the number of MTAs should be kept small, and other mechanisms used for delivery to local hosts. Possibilities include:

- NFS. This simply consists of NFS-mounting (possible through use of an automounter) the necessary files systems for delivery.

- Remote access protocols. Use of services such as POP[Ros88] may be an alternative.

- Delivery to a central place which is NFS-mounted by clients.

For these reasons, some of the PP utility programs may be accessed remotely using *rsh*(1).

A site with 1000 users and 100 hosts might consider installing between one and four MTAs. Reasons for choosing MTAs might include:

- External communication links.

- Large host with many users.

- Fileservers used substantially for mail.

- Separate administration domains.

1.7 Performance

PP is running on a Sun 4/330 at the nsfnet-relay.ac.uk mail hub, moving around 15,000 messages a day between the UK and the US. Experience from this and other sites has been fed back into the development process, with the result that PP 6.0 performs better than its predecessors. However, as is the case with any application, PP could still benefit from more fine tuning.

Chapter 2

Installing PP

This is a description of things to do when installing PP. The basic information is described in *pp-gen*(8), but this section gives the full details. It is recommended that you read this section before compiling or installing the software.

2.1 Compile Time Configuration

Most of the definitions that have to be set before compiling PP are given in *Make.defs* in the top level of the source tree, and *config.h* in the `h/` directory.

2.1.1 Make.defs

A number of *make* variables can be set. These variables come in two classes. The first are variables that are truly site specific. These are listed in Table 2.1. The second class are variables that you can probably leave as the defaults. These are listed in Table 2.2. For each entry a reasonable default is given.

A number of these variables are important, so care must be taken when configuring these variables. A description of each of these variables is now given, however Tables 2.1 and 2.2 give a good overview of the variables.

`CMDDIR`: This is the directory into which most of the PP core commands are placed. This is usually the directory that contains subdirectories for channel and formatting programs.

`CHANDIR`: This directory is normally a subdirectory of `CMDDIR`. It contains all the channel programs that PP knows about.

`FORMDIR`: This directory contains the formatting programs that are run by *filtercontrol*. These can be either binary executables or shell scripts.

`TOOLDIR`: This directory is where a number of miscellaneous tools are placed. These are useful programs that aid in the day-to-day running of PP but are not usually used directly by the PP system.

Variable	Example	Description
TAILOR	/usr/lib/pp/tailor	Location of the pp tailor file
CMDDIR	/usr/lib/pp	Directory for PP basic commands
CHANDIR	/usr/lib/pp/chans	Directory for channel programs
FORMDIR	/usr/lib/pp/format	Directory for the simple formatting channels
TOOLDIR	/usr/lib/pp/tools	Directory for miscellaneous shell scripts and debugging tools
LOGDIR	/usr/lib/pp/logs	Directory for PP logs
QUEDIR	/usr/spool/pp	Location of the PP queue directory
TBLDIR	/usr/lib/pp/tables	The directory containing the table files
LIBSYS	-ldsap -lisode	External libraries required - must include ISODE and DSAP
LIBRESOLV	-lresolv	Resolver library for bind (empty if nameserver not required)
USRBINDIR	/usr/local/bin	Directory for user binaries
OPTIONALCHANS	list 822-local smtp	List of optional channels you require
OPTIONALFILTERS	ascii2fax	List of optional filters
MANDIR	/usr/local/man	Directory to place manual pages under
MANOPTS	-bsd42	Manual page installation options
NIFTPSRC	/usr/src/niftp	Directory containing the source to unix-niftp package. (Only required if you wish to run grey book mail).
NIFTPINTERFACE	sun	Niftp interface being used (restrictions as above)

Table 2.1: Make Variables That Should Be Set

Variable	Default	Description
CC	cc	The *C* compiler to use
CCOPTIONS	-O	*C* compiler options
LIBCCOPTIONS	-O	*C* compiler options specific to library files
OLDCC	cc	Other cc if gcc is used
LDOPTIONS	-s	Loaded options
PEPY	pepy	The *pepy* ASN.1 parser/compiler
POSY	posy	The *posy* stub generator
ROSY	rosy	The *rosy* remote operations stub generator
PEPSY	pepsy	The *pepsy* program
X11	true	Does the system have X11?
LIBX	-lXaw -lXmu -lXt -lX11	The X11 libraries - may need -lXext for X11R4
LINT	lint	The *lint* (1) command
LINTFLAGS	-hbuz	Lint flags
PGMPROT	755	Program protection mode
PPUSER	pp	The user id PP will run as
ROOTUSER	root	The user id for privileged channels
CHOWN	chown	The *chown*(8) command
CHMOD	chmod	The *chmod*(1) command
BACKUP	cp	A command to back up the old binary
INSTALL	cp	A command to install a new binary
LIBPHOTO	-lphoto	The quipu photo library
FAXDRIVERS	dexNet200	The fax modem to compile

Table 2.2: Make Variables That Can Be Defaulted

`LOGDIR`: This directory is where the logging files are written to by default. These logs are used for a variety of purposes, including debugging and statistics gathering. You should allow for a reasonable amount of space.

`TAILOR`: This specifies the location of the PP tailor file. It is important you get this right. Almost every PP variable can be changed in the PP tailor file, but obviously the location of the PP tailor cannot. If you get this wrong, you will need to relink all the PP binaries against an updated *libpp.a*.

`QUEDIR`: This is the directory under which the PP queue is maintained. As all messages sent are queued here, you should allow enough room to handle all the mail you are likely to receive. A busy site with just a day or two's network outage can gather several megabytes of mail.

`TBLDIR`: This is the directory in which all the PP routing tables and similar configuration files are kept.

`LIBSYS`: This variable contains all the external libraries that you may require when building most of the PP binaries. It should contain at least the `ISODE` librarys `-lisode` and `-ldsap` and possibly other libraries such as the *dbm* (3) library and any machine specific libraries you may need.

`LIBRESOLV`: This variable contains the library for the bind domain name server (or equivalent). It is only required if you are making use of the DNS (see Section 2.1.2). Leave it empty otherwise.

`USRBINDIR`: This is a directory where binaries that are used by users are placed. This includes a simple mail interface, and some delivery time processors.

`OPTIONALCHANS`: PP requires a certain core set of channels to operate at all, plus a number which are used in all normal configurations. However, there are a number of protocol and experimental channels which can be included if required. This variable specifies which of the optional channels and protocols you wish to support. Currently this consists of a list made from the following:

`822-local`: Basic RFC 822 delivery.

`lists`: Support for distribution lists.

`grey`: Support for JNT mail protocol (Grey book).

`shell`: Support for delivery to programs.

`slocal`: Support for structured local delivery (experimental).

`smtp`: Support for the SMTP protocol.

`uucp`: Support for the UUCP protocols (primitive).

`x40084`: Support for the X.400 P1(1984) protocol.

x40088: Support for the X.400 P1(1988) protocol.

fax: Experimental fax delivery.

decnet: Contributed DECnet[1] MAIL11 channel.

OPTIONALFILTERS: PP has a number of filters that can be used. Some are built by default; others are only useful if you plan to support the given style of content. Leave this empty initially if you are unsure; the defaults are usually sufficient.

FAXDIRVERS: The PP fax channel can driver a number of FAX modems. This variable should be configured if you are using the fax channel to specify the particular driver(s) to use.

FAXPAGESIZE: This macro alters the size of page the fax reformatter filters produce. If not defined the filters will produce US size pages, 216mm by 273mm. If equal to `-DA4` they will produce A4 size pages, 210mm by 297mm.

MANDIR: This is the directory under which manual pages are placed. The installation process understands about most layouts of manual pages, so this is usually a base directory.

MANOPTS: This variable configures the style of manual pages to be installed. It can take the following values:

Option	**Format**
`-bsd42`	man<N>/file
`-bsd44`	cat<N>/file.O
`-ros`	man<N>/file – using /etc/install
`-sys5`	<a|p|u>_man/man<N>/file
`-local`	manl/file.l
`-l`	man<N>/file.<N>l
`-hpux`	HP manual style

NIFTPSRC/NIFTPINTERFACE: These two variables control the PP interfacing with the unix-NIFTP code. These variables are ignored unless grey book mail is being installed. The variables should be set to the source directory of the unix-NIFTP package, and the particular interface the grey book channel will use. Suitable interface settings include:

- sun (this is the only version currently in use.)
- inet.
- ubc.
- dexpand.
- x25b.
- yorkbox.

[1] DECnet is a trademark of Digital Equipment Corporation.

`CC`: The *C* compiler to use. The PP system has been compiled with a number of *C* compilers.

Warning: If you wish to use the *gcc* compiler on `sparc`-based system, you must compile files making use of *dbm*-style routines with *cc*. (The compiler will normally warn about these files.) See `OLDCC` below.

`OLDCC`: The *C* compiler to use on dbm-related files. This is to work around a problem with *gcc*; it should usually be set to *cc*.

`CCOPTIONS`: This contains global *C* compiler options. If your include files for `ISODE` are in a non standard place, a "`-I`" flag or two may be necessary here.

`LIBCCOPTIONS`: This contains global *C* compiler options to use when compiling files to place in libraries. In normal use this will be the same as `CCOPTIONS`.

`LDOPTIONS`: This variable contains global loader flags. It is normally empty or just contains the "`-s`" flag.

`PEPY/POSY/ROSY/PEPSY`: These variables are the *pepy*, *posy*, *rosy* and *pepsy* programs. Normally they just contain those names, but may be full pathnames if they are not in the standard search path.

`X11`: If you have X11 then set this variable to *true*. Leave it empty and no X-based programs will be compiled.

`LIBX`: The X11 libraries. This contains a list of the X libraries necessary to compile the few X-based programs in the distribution. This distribution is aligned to the X11R4 release; it will *not* run under earlier versions of X.

`APPDEFAULTS`: The application default directory. This contains the directory in which to install the application defaults for the various X-based programs.

`LIBPHOTO`: The `ISODE` photo library. This is required for the *xalert* program. Alternatively you can set `-DNOPHOTOS` in the *config.h* file to compile this without photograph support.

`LINT`: The *lint* (1) command. This is not normally used unless you are doing development work.

`LINTFLAGS`: A list of global lint flags. This variable is also not normally used unless development work is being done.

`PGMPROT`: This is the default mode to install programs as. Some programs need special modes, but these are handled separately.

`PPUSER`: This is the username that most PP programs are installed under. It is usually `pp`, but need not be.