Xi Software

GNUspool Release 1

System Reference Manual

Table of Contents

1 Introduction

GNUspool is a fully functioned, high performance Print Spooler and Management System for a wide range of machines running a Unix-style Operating System.

The product consists of a "core product" or "basic product" which contains the scheduling software, command-line and character-based interfaces. Additional options provide for:

- An interface to receive incoming jobs from LPD-style clients
- An interface to send outgoing print jobs to LPD-style clients
- An X-Windows Motif Toolkit Interface (not supported on GNU)
- An X-Windows GTK+ interface.
- An API for use with C and C++
- An Interface for MS-Windows (this needs rewriting as it currently uses non-free software)
- An API for use with MS-Windows
- Browser Interfaces

The basic manuals cover the "basic product" and the X-Windows interfaces. Additional supplements cover the other interfaces.

The basic manuals are:

- User Guide - a quick introduction and "cookbook" for use of GNUspool
- Reference manual - this manual. A complete description of all components of the basic product.
- Administrator Guide. Information about installation and customisation of the software.

Also available are:

- API reference manual for Unix and MS-Windows API
- MS Windows Interface Manual
- Browser Interface Manual

1.1 Documentation Standards and Program Usage

This manuals uses various character fonts to indicate different types of information as follows:

`File names and quotations within the text`
`Examples and user script`
`Generic data (where you should put a value appropriate to your own environment)`
Program names, whether for GNUspool or standard Unix facilities

Warnings and important advice

1.2 Command Line Program Options

Almost all of the programs that make up GNUspool can take (or require) options and arguments supplied on the command line. As much flexibility as possible is allowed in the specification of these options and arguments; White space may be inserted into flag arguments as in

```
gspl-pr -c 3 -f invoices
```

or it may be left out as in

```
gspl-pr -c3 -finvoices
```

Single character options may be strung together with one minus sign:

```
gspl-pr -mwsv
```

or separated, as in

```
gspl-pr -m -w -s -v
```

If mutually contradictory arguments are permitted, the rightmost (or rather the most recently specified) applies.

The ability to redefine option letters has been provided, together with the `+keyword` or `--keyword` style of option. Such options should be given completely surrounded by spaces or tabs to separate them from each other and their arguments, for example

```
gspl-pr +copies 3 --formtype invoices
```

In addition, all the commands have an option `-?` or `+explain` (or `--explain`) whose function is to list all the other options and exit.

There is a mechanism for picking up options from environment variables or `.gnuspool` files containing the relevant keyword.

2 Licence

GNU Free Documentation License

Version 1.2, November 2002

Copyright (C) 2000,2001,2002 Free Software Foundation, Inc. 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent.

An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit.

When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free

Software Foundation.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

> Copyright (c) YEAR YOUR NAME.
>
> Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU   Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

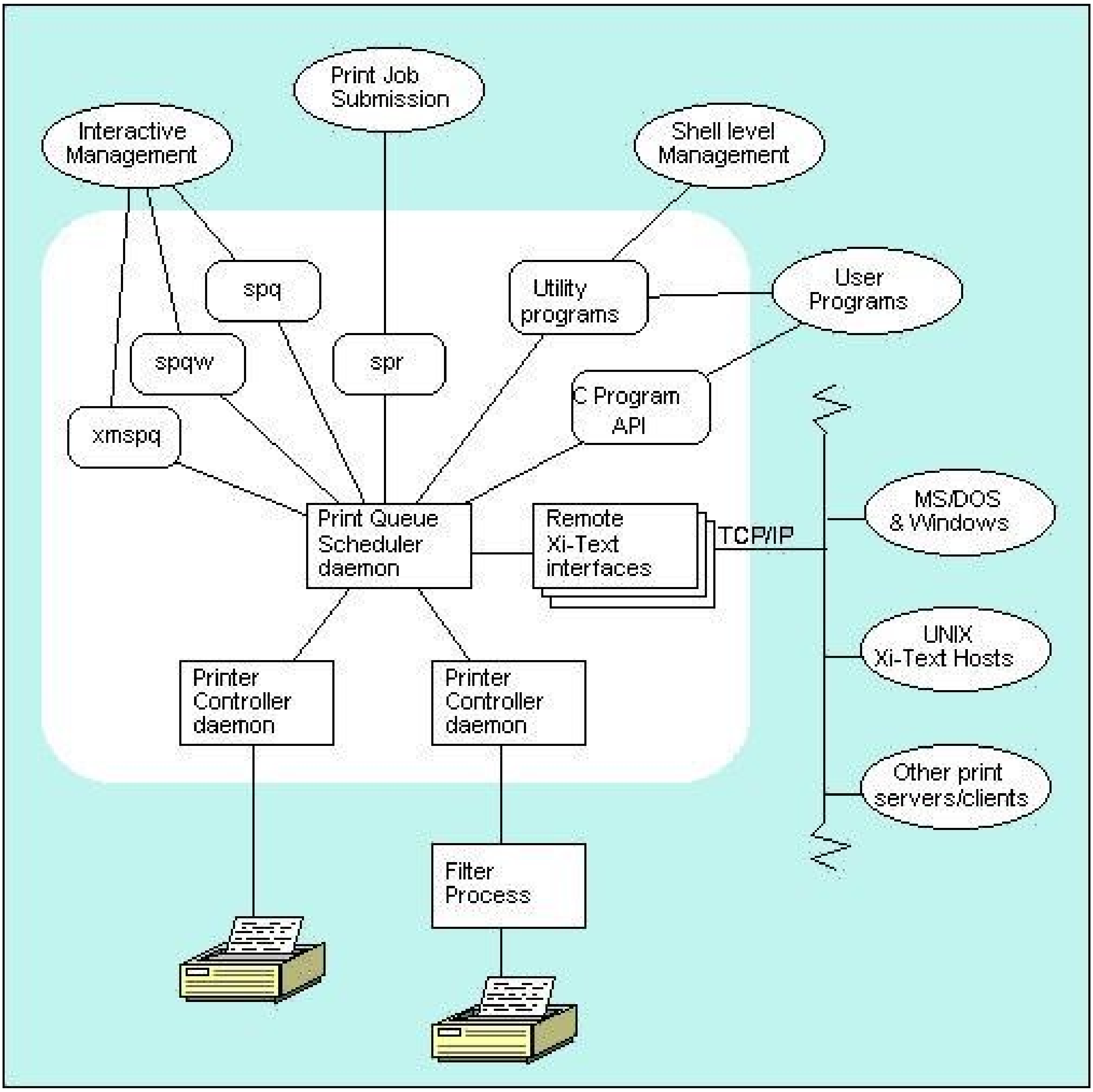
> with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains non-trivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

3 Overview

GNUspool can run on a single Unix host or several co-operating machines. The central white area of the diagram shows the possible components of GNUspool on a Unix machine. The shaded area indicates the entities, outside of that machine's GNUspool system boundary, which use or provide services to it.



At the heart of GNUspool is the print queue scheduler daemon spshed, which manages the print jobs. There is one instance of spshed running on a stand alone system, or two if co-operating with other GNUspool hosts. Co-operating GNUspool hosts require a network providing TCP/IP services.

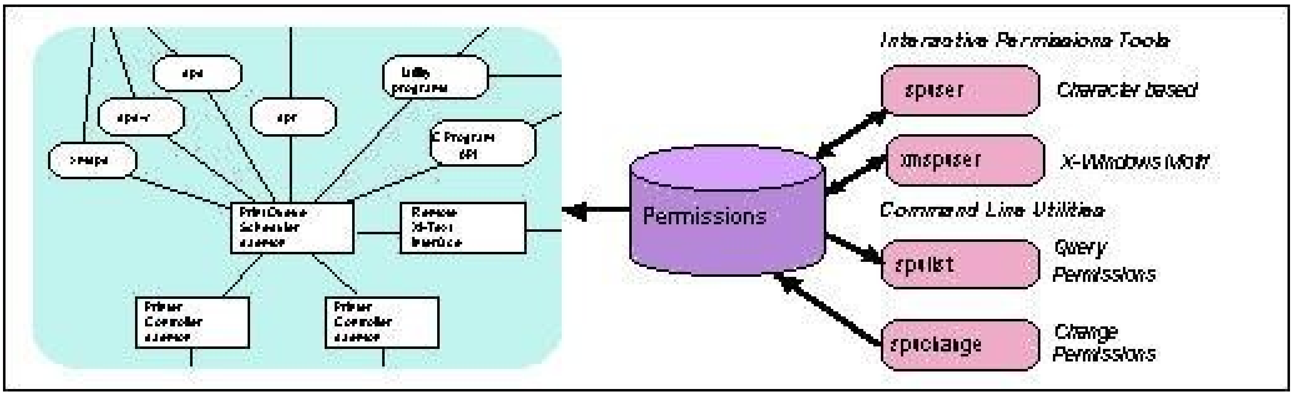
Each printer that is currently enabled to print is managed by an instance of the printer controller or "despooler" daemon spd. This daemon may communicate directly with a device driver or some intermediate program, as required. Such intermediate programs, or filter processes, may be used for special post-processing or interfacing to network connected printers.

Jobs can be submitted to the print queue for printing using the programs gspl-pr and gspl-lpcover (our version of the standard lp command). Windows PCs and machines that send jobs to remote lp servers can also submit jobs to GNUspool. These facilities are available via our Windows Client PC Option, described in a separate manual, and the LPD interface, described in chapter 13.

There are three mechanisms for permitted users or system administrators to query and manage the scheduler. All of these communicate with spshed via signals and the Unix IPC services:

1. Interactive screen based tools for Character Terminals (gspl-pq), X-Windows Motif (gspl-xmpq) and PC Windows clients (spqw).
2. A suite of utility programs for use from the command line or within shell scripts.
3. The web browser interface support (CGI) programs, sqcgi, spcgi etc.
4. A 'C' Programmer's API.

There is a sophisticated system of permissions to control what each user is allowed to do using these mechanisms. These are administered using the following choice of interactive and command line programs, which were not shown in the previous diagram.



3.1 Directory and File Structure

The files which comprise GNUspool are held in various directories depending upon their nature. With the exception of global configuration files the installation can be tailored to suit local practices and standards.

- Global configuration files are always held in the `/usr/local/etc` directory.
- User programs can be placed in any directory which is on the GNUspool users' `PATH` variable.
- Internal programs and data are held in three or sometimes four separate

directories.

3.1.1 Internal Directories

GNUspool uses various directories to hold the internal programs and data.

These directories may be relocated from the values set when GNUspool is built by assignment to environment variables. These environment variable assignments may (and probably should) be placed in the master configuration file, `/usr/local/etc/gnuspool.conf`, to ensure consistency. The directories and corresponding environment variables are as follows:

Directory	**Environment Variable**	**Function**
`libexec/gnuspool`	`SPROGDIR`	Internal programs
`share/gnuspool`	`SDATADIR`	Data and control files
`share/gnuspool/help`	`SPHELPDIR`	Help files for programs
`share/gnuspool/ptrs`	`SPOOLPT`	Description of printers and form types
`var/gnuspool`	`SPOOLDIR`	Spool directory for pending jobs
`var/gnuspool_alt`	`ALTSPOOL`	Alternate spool directory

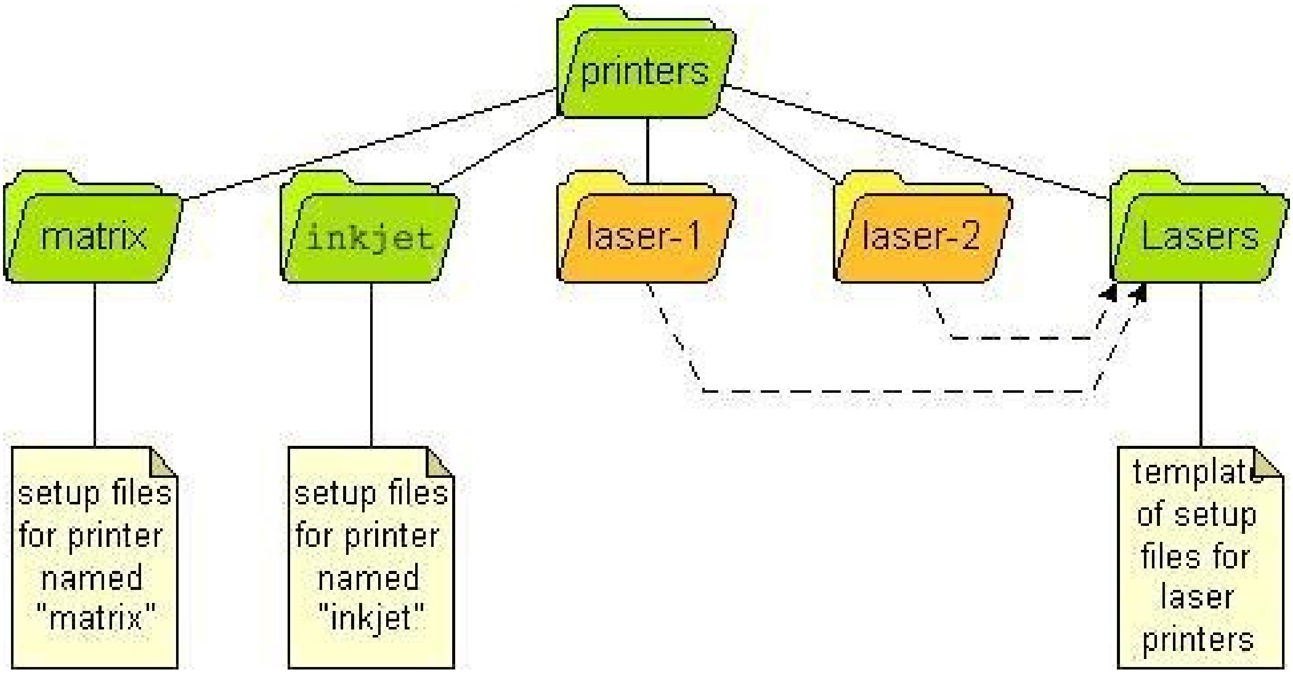
Take care not to assign values to these environment variables arbitrarily; very strange things will happen if one part of **GNUspool** is using one set of directories and some other part is using another!

Do not confuse these environment variables with those passed to filter processes, which are set up during printing.

3.1.2 Printer Definitions and Setup Files

The printers directory contains the print control and formatting information for all of the printers that are logically attached to GNUspool on that Unix host. Within this directory there is a sub-directory for each printer. Each sub-directory holds one or more setup files for the printer, which make up the printer definition.

To maintain identical printer definitions for two or more identical printers a master directory can be set up as a template. Each of the printer directories can then be replaced by a symbolic link to the template directory.


printers
matrix
inkjet
laser-1
laser-2
Lasers
setup files for printer named "matrix"
setup files for printer named "inkjet"
template of setup files for laser printers

Both "hard" and symbolic links can also be used inside printer directories where just certain components are identical across two or more printers.

Setting up printers and their definitions is described in chapter 5 of this manual.

3.1.3 Help and message files

GNUspool reads all of its messages (apart from the "help I cannot find the help file" messages) from a set of text files which may be modified to tailor the command interface, help and error messages. This can provide anything from national language support to minor changes in nomenclature to match corporate standards. These are *system-wide* message files, it is common practice to set up customised versions for individual users or applications.

The files are:

File	Function
`spq.help`	Message file used by gspl-pq.
`xmspq.help`	Message file used by gspl-xpq and gspl-xmpq.
`spuser.help`	Message file used by gspl-user.
`xmspuser.help`	Message file used by gspl-xuser and gspl-xmuser.
`splpd.help`	Message file used by LPD interface programs
`rest.help`	Message file used by all other user programs including the web browser interface.
`int-config`	Message file used by internal programs.

By default the files are held in the directory `/usr/local/share/gnuspool/help`, however a different location may be chosen if desired.

3.1.4 Spool directory files

The following files are held in the spool directory, `/usr/local/var/gnuspool` by default.

File	Function
`pwdump0`	Optional saved password map file (see gspl-uchange).
`spufile0`	File of user permissions (with suffix of the major release of GNUspool).
`spcharges0`	File of user charges (with suffix of the major release of GNUspool).
`sputmp`	Temporary file used when reformatting `spufile23`
`spshed_reps`	Log file recording system startup. shutdown and errors
`spshed_jfile`	File used for recording the job queue
`spshed_pfile`	File used for recording the printer list
`SP`*nnnnnnn*	File holding the text of a spool job
`PF`*nnnnnnn*	Holds page offsets where the page delimiter is not a form feed.
`ER`*nnnnnnn*	Used to hold information to be mailed back to a user.
`SPDE`*nnnnnnn*	Used to hold the standard error output of filter commands.

Unused copies of the last four kinds of file may safely be deleted, or requeued and tidied up with the gspl-gspl-setspdir utility. The *nnnnnnnn* component of the file name is derived from the print job number.

The gspl-setspdir utility may also be used to divide up the spool directory into component subdirectories, possibly on different file systems, in order to make the spool directory manageable when it contains a large number of jobs. Such subdirectories are given 3-digit numeric names `000` upwards.

3.1.5 Internal programs

These include the scheduler daemon, printer management daemon, and the utilities used by them. They are held in the internal programs directory. With certain exceptions it is not intended that users should ever invoke these programs.

The file structure of the internal programs is flat within their directory.

All of the internal programs are described in Chapter 6 of this manual.

3.1.6 User Programs

These include the printer and user administration programs. The GNUspool permissions system ensures that only authorised users may use sensitive programs and options.

All of these programs will be owned by `gnuspool`, the GNUspool user. They will also have the Set UID file privilege set. In addition to the usual read and execute bits this gives them a mode of `4755`. Running `ls -l` on gspl-pq should look similar to this:

```
-rwsr-xr-x 1 gnuspool  bin  167619 Mar 23 2000  gspl-pq
```

The user programs are described in detail in Chapter 7 User Programs.

3.1.7 Configuration files held in usr/local/etc

3.1.7.1 GNUspool Hosts File

The file `/usr/local/etc/gnuspool.hosts` is used to denote details of the remote hosts and clients to which connection is to be made.

Each line in the file other than blank lines or comment lines (introduced with a `#` sign) consists of up to 4 fields. These are as follows:

1. The *hostname* to attach to or an internet IP address such as `197.3.9.1`. For DHCP clients, this gives the Windows user name to be recognised (case insensitive).
2. An *alias name* by which the remote host is to be referred to within GNUspool. The user can give either the host name or the alias name in commands such as gspl-conn but displays (as in gspl-pq or gspl-plist) will always use the alias. For DHCP clients, this gives the Unix user name (if different) corresponding to the given Windows user name.

   An alias or Unix user name can be omitted by just putting a single "`-`" sign.

   An alias *must* be supplied if the host name is given as an internet address.
3. Flags, which are further described below.
4. A numeric time-out value in seconds. The default if this is omitted is 1000. This is most important for Windows clients, as it also denotes a time after which the connection becomes "stale" and must be refreshed, possibly by re-entering the password.

The *`flags`* field is one or more of the following separated by commas.

`probe`	denotes that the scheduler should check that the specified host is active before attempting a connection.
`manual`	denotes that no connection is attempted until the operator invokes one with gspl-conn.
`external`	denotes that the named host is some external system, possibly running LPD protocol, for the benefit, in particular, of incoming requests from xtlpd using the `-o` (originating host) option to gspl-pr.
`dos(`*`user`*`)`	for a Microsoft Windows client PC. Requests are allocated by default to the username given.
`client(`*`user`*`)`	is a synonym for `dos(user)`.
`clientuser`	denotes that the first and second fields are user, not machine names, for DHCP clients. As a special case, if the first field is default and the second field is a user name on the Unix host, then a default user name is thereby supplied for all unknown Windows users.
`clientuser(`*`machine`*`)`	as `clientuser`, but denotes that the default client machine is given, otherwise a password is required.

`trusted`	exchange information with this Unix host about Windows client users.
`pwchk`	demand Unix password (or an interface password as set by pgspasswd from Windows clients in all cases.

For example:

```
mach19         red      probe
mach20         green    probe,manual
192.112.238.7  yellow   probe
WS21           blue     dos(jmc)         30
john           jmc      clientuser,pwchk
default        guest    clientuser
```

This provides for 4 machines, where host names are `mach19`, `mach20`, `WS21` an IP address and also user names for DHCP clients. These are given aliases of `red`, `green`, `yellow` and `blue`.

In the first to third case any connection will be tested first before continuing.

In the `green` case no connection is attempted until the user types,

```
gspl-conn green
```

or

```
gspl-conn mach20
```

Next the `blue` machine is a Microsoft Windows workstation. Requests will be assumed to come from user `jmc`. Time-outs of 30 seconds apply to requests.

Next the Windows user name of `john` on any Windows PC is translated to a Unix user name of `jmc`, after checking the password. (The Unix and Windows user names may be either way round, but one name must be a valid name on the Unix host).

Finally, any unrecognised Windows user name is treated as the Unix user name of guest.

The utility program gspl-hostedit may be used to create or edit this file with appropriate checks.

3.1.7.1.1 Multiple IP addresses

If the host on which GNUspool is to be installed has multiple IP addresses, and one is to be selected for use when communicating with other hosts, possibly different from that returned by system utilities to get the local IP, then the correct IP address may be selected using as a "local address".

This is specified thus:

```
localaddress 193.112.238.250
```

The localaddress statement must be the first item (other than comments or blank lines) in the host file.

3.1.7.2 GNUspool Master Configuration File

In order to work properly, the scheduler process and all the other programs must be started with the same environment variables. For convenience, the environment may be initialised for each program by creating a *master configuration file* `/usr/local/etc/gnuspool.conf`.

This is particularly important where some of the main directories, such as the spool directory, are to be relocated.

This file contains a list of environment variable assignments. Any environment variables not defined on entry to any of the programs are initialised from this file. (They do not have to be just the ones listed above).

For example:

```
SPOOLDIR=/usr1/spool/spd
SPROGDIR=/usr1/spool/bin
MAILER=/usr/lib/sendmail
SPOOLOPTS=-v
```

Please note that the text to the right of the `=` sign is taken literally; there is no recursive expansion of *`$name`* constructs except for the message file names `SPQCONF`, `SPUSERCONF` and `SPRESTCONF`.

3.2 Impact of different Unix Security features.

The security and accounting mechanisms built into GNUspool are explained in Chapter Error: Reference source not found, User Administration. Here we describe the security implications of GNUspool itself.

Gspl-spq processes are warned by a signal from spshed to reread the job queue. Printer processes are likewise sent restart job and abort job messages. Therefore spshed needs to be able to send signals to all gspl-pq processes. On some versions of Unix, processes can only send signals to processes whose real UID matches their effective UID. If your version of Unix is like this, spshed must be made set-UID to root, to ensure that signals can be sent.

If, however, on your system, processes can send signals to any process whose real or effective UID matches their real or effective UID, spshed will work satisfactorily with set-UID `gnuspool` rather than set-UID `root` (we do provide utilities to work this out).

To see if this works on your system, check for the correct functioning of queue refresh functions in gspl-pq, together with abort and restart of the current job on the printers.

3.3 Standard Exit Codes

It is often convenient to run gspl-pr or gspl-start from other programs or shell scripts. In order to allow convenient error diagnosis, there are a standard set of exit statuses which are used by all the GNUspool programs.

Code	Description
2	Bad class specification - would give zero

3	Bad command arguments e.g. unknown option
5	Cannot find spooler directory (usually `/usr/local/var/gnuspool`)
6	Scheduler not running or crashed
7	Illegal priority specified
8	Illegal form type specified
9	Illegal number of copies specified
10	Bad printer name specified
11	gspl-start - printer is being shut down
12	gspl-start - printer is in `a/w oper` state
13	Unknown job (gspl-qdel or gspl-qchange)
16	Insufficient privilege for requested operation
40	(Internal to spdinit) bad setup file
100	Bad or non-existent configuration file
101	I/O error on terminal input
150	(Internal to spjobdump) file not found.
151	(Internal to spjobdump) cannot change directory.
152	(Internal to spjobdump) cannot create output file.
153	(Internal to spjobdump) job not found.
200	Unexpected signal received
201	Cannot create pipe
202	Cannot fork
203	Job queue non-existent (internal error)
204	Print queue non-existent (internal error)
230	gspl-pr - cannot create spool file - is disk full?
240	Scheduler internal error
250	GNUspool badly installed; setup error
254	Run out of memory
255	Missing message file

The command gspl-pstat returns 0 (shell TRUE) if the specified printer is in the specified state, 1 (shell FALSE) if it is not in the specified state, or one of the above error codes if an error is detected.

Error numbers less than 100 are for "normal" errors, such as invalid options or privilege violations. Error numbers >100 are "internal" or system generated errors, i.e. they are outside the control of GNUspool.

In the cases of gspl-start (and its links gspl-phalt, gspl-ok, gspl-nok, gspl-pinter, gspl-padd and gspl-pdel), gspl-qdel and gspl-qchange exit codes less than 100 will not cause the program to halt; the program will do as much as it can, reporting appropriate error messages and terminating with an exit code corresponding to the last error message reported.

The exit codes for spjobdump and spdinit are recorded here for completeness; in fact they are not expected to be run except from gspl-pq and spd respectively, which will interpret the error codes and report appropriate error messages.

4 User Administration

GNUspool maintains a list of users which is generated from the password system (whether using the `passwd` file or NIS). Hence, each user who has a Unix account will also have a GNUspool account.

To add a new user to GNUspool, they must first have a Unix login created. Once the Unix account is set up, run the interactive user administration program gspl-user with the command `gspl-user -i`. Alternatively to create their GNUspool account non-interactively run the command `gspl-uchange -R`. This will create the user with the machine's current default user parameters.

There are 4 aspects to the GNUspool user account:

Privileges	control access to usage and administration functions of the system. For example, the privilege to delete jobs from the queue.
Classcodes	effectively subdivide the print resources between separate groups of users.
Charges	provide an accounting mechanism for each users print output.
Priorities	position print jobs in the queue relative to the other jobs. The higher the priority the closer to the front of the queue a job is placed. Facilities exist to specify what priorities each GNUspool user may put on their print jobs.

The privileges and classcodes provide two different dimensions of security. The privileges control what users may do. The classcodes control what jobs and printers users may exercise their privileges on.

4.1 Privileges

The following aspects of **GNUspool** are subject to control by privileges.

Privilege Name	**Meaning**
Edit admin file.	Assign other users' privileges
Override class.	Increase class code (set bits on)
Stop spooler.	Run gspl-stop
Use other forms.	Select other than pre-defined form type
Use other printers	Select other than pre-defined printer(s). On by default.
Change priority on queue.	May change priority of job on queue
Edit other users' jobs.	May inspect or change other users' jobs
Select printer list.	May move to printer list in gspl-pq
Halt/restart printers.	May start or stop printers
Add/delete printers.	May add or delete printers on list
Set any priority on queue.	May set any priority not just in range
Change own default prio/ form.	May run `gspl-user -c`

Unqueue jobs.	May take jobs off the queue, copying to a file for editing
View other users' jobs	May inspect but not change other users' jobs
Edit remote jobs	May edit jobs on remote machines
Edit remote printers	May change printers on remote machines
Access attributes	May access attributes of jobs not displayed by default on gspl-pq. On by default.
Freeze attributes	May save changes to default display. On by default.

In addition there are configurable limits and per-user values for:

- The number of copies
- Default, minimum and maximum priority
- Default form type and permissible form types (a pattern).
- Default printer and permissible printer types (a pattern).

Some privileges imply others; for example adding and deleting printers implies starting and stopping them as well, which in turn implies selecting the printer list.

The privileges on one machine, with GNUspool networked, apply to that machine and to operations performed via that machine. These operations may affect jobs or printers on other machines which the user would not be permitted to access. For example it is possible to be set up to stop a printer on a remote machine which you could not stop if you logged in to that machine. The administrator should try to avoid this situation if possible.

When installed, the users `gnuspool` and `root` (the super-user) have all the privileges and other users none. The privileges and defaults should be set up for each user and the defaults for new users, using gspl-user or gspl-xuser. Current privileges are displayed by typing:

```
gspl-user
```

This will produce output something like (for user `gnuspool`):

```
Default form standard types standard
Maximum Copies 10
Priority: Min 10 Max 100 Def 50 Current charge is 321
Default printer '' types ''
Class Code A-Pa-p
Privileges:
You may Edit admin file.
You may Override class.
You may Stop spooler.
You may Use other forms.
You may Use other printers.
You may Change priority on Q.
You may Edit other users' jobs.
You may Select printer list.
You may Halt/restart printers.
You may Add/delete printers.
You may Set any priority on Q.
```

```
  You may Change own default prio/form.
  You may Unqueue jobs.
  You may view other users' jobs.
  You may Edit remote jobs.
  You may Edit remote printers.
  You may access queue options.
  You may save new default options.
```

The first line gives the default form type, in this case `standard`, and the permissible form types. Then is the maximum number of copies. The next 3 figures give the priorities; the lower and upper bounds, and the default if none is given.

The charge is the total charge so far accumulated by this user; see later for details.

When a new user is added, GNUspool will give them an initial set of values for the minimum, maximum and default priorities, form type, privileges, class code and maximum copies.

These initial values may themselves be altered by the administrator to apply to subsequently created users, or explicitly copied to existing users. When GNUspool is first set up, these initial values are set as follows:

1. The minimum, maximum and default priorities are set to 100, 200 and 150.
2. The initial default form type is set to standard.
3. The privileges are set to all zero, except for the super-user and `gnuspool`.
4. The class code is set to `ABCDEFGHIJKLMNOPabcdefghijklmnop` (all ones). This is usually represented as `A-Pa-p` to save space.
5. The maximum number of copies is set to 10.

Priorities and maximum copies to be set may range from 1 to 255.

4.2 Priorities

A spool job may have a priority of 1 to 255. Users will usually be restricted to a smaller range between their individual minimum and maximum priorities, but which are normally the system defaults, initially 100 to 200. A default priority for each user may be set; again there is a system default, initially 150.

Increasing the priority of a job increases its chances of getting printed sooner, as follows.

- When a job is added to the queue, a copy of the priority is made. This is the working priority.
- The bottom item on the queue is considered. If the priority of this item is less than the working priority of the new job, then the working priority of the new job is decremented and the process repeated with the next item.
- This process is continued until the job reaches the top of the queue or a job is encountered with a priority the same as or greater than the working priority.
- The working priority is saved in case the job's priority is subsequently changed. The change to the job's priority is reflected in the working priority,

and the job moved up or down the queue according to the above rule.

When a job is queued using gspl-pr, it is given the user's default priority unless overridden. It is possible to set a users minimum, maximum and default priorities to apparently useless values, but, in fact, these combinations provide possibly useful restrictions:

Combination	**Meaning**
min <= default <= max	Normal usage; User may set priority anywhere between *min* and *max*.
min = default = max	Special case; User can only use *default* priority.
min > max	"Impossible"; User may not use **GNUspool**, gspl-pr etc will complain.
default > max or *default < min*	Default priority is effectively undefined; user **must** explicitly give a priority to gspl-pr etc, or the job will be rejected with an error message.

Jobs belonging to remote machines may appear in different places on the queue than on their machines when they initially come on line, but this situation, which is harmless, should in any case rapidly adjust itself.

4.3 Charging

At the end of each job the cost of printing that job is added to the user's stored charge. The cost of the job is calculated using the algorithm:

$$\left\lfloor \frac{\left\lfloor \frac{ncp+999}{1000} \right\rfloor \times cpt}{1000000} \right\rfloor \times \frac{jp^2}{sp^2}$$

Where:

ncp is the number of characters printed
cpt is the charge per 1000 characters printed for the printer as defined in the setup file.
jp is the job's priority and
sp is hardwired to the installation default priority of 150 units.

If a job, submitted on one machine, is printed on another over the network, then the charge for the printer on the machine where it was printed, and the system default priority on the machine on which the job was submitted are used. The result is used to update the users' account on the latter machine.

In simple terms; the user is charged for the number of characters printed multiplied by the charge per character for that printer (see printer charging), scaled by an amount dependent on how much the priority of the job had been raised above the system default when it was printed.

Using the square of the priority is to penalise users in greater amounts the more they "jump the queue".

It is proposed to remove charging from subsequent releases of GNUspool. When it was first written, people were conscious of such things but now no one appears to use it.

4.4 Class codes

Class codes allow printers and jobs to be "visible" to one group of users, but not to another. For example it might be desirable for the accounts department to have their printers and jobs concealed from users who otherwise have quite general access to spool files and printers. Likewise it may be useful for the accounts department to be disallowed from accidentally stopping or restarting printers unconnected with their work.

To achieve this, every user and every printer has a class code (displayed in the output of gspl-user). This is a string of 32 bits, represented everywhere by the upper case alphabetic characters `A` through to `P` then the lower case character `a` through to p.

The interpretation of the bits is entirely at the disposal of the system administrator, but a possibility is to give each "group" of printers a single bit, for example

Code	Printer "Group"
`A`	General Printers
`B`	Accounts dept printers
`C`	Laser printer 1
`D`	Laser printer 2
`E`	Plotter 1
`F`	Plotter 2

In the examples in the following sections, these assignments will be assumed, to simplify the discussion. However we do want to stress that the usage is entirely up to the administrator in any given situation.

4.4.1 Treatment of class codes

GNUspool assigns a class code to each user, each printer, and each job. These class codes are *matched* by taking the bitwise *and* of two class codes. If the result is non-zero, then the class codes *do* match, otherwise they *do not* match. Put another way, if one or more bits are common, then they match, otherwise they do not.

A class code to be assigned to a user, printer or job is never permitted to be zero as that would mean that printers or jobs would be invisible as would any created by the user.

For example, suppose that the user has a class code of `ACDEF`. Then this will match (using the example in the previous section), the general printers, the 2 laser printers and the 2 plotters, but not the accounts dept. printers. On the other hand, members of the accounts dept. might have class codes of, for example,

`B`	covering the accounts dept. printers, but nothing else,
`BCD`	allowing access to the accounts dept. printers plus the 2 laser printers,
`ABCDE`	allowing access to the general printers, the accounts dept. printers, the 2 laser printers and the first plotter.

Any job submitted by a user has the class code of that user assigned to it, unless that user provides otherwise. Similarly, by default any printer added to the system is assigned the class code of the user who added it. These can be varied, by turning off bits, within the limits of the user's class code, unless the user has the override class privilege, enabling any class code to be set.

1. A job is assigned to a printer only if its class code matches (i.e. has at least one bit in common with) that of the printer. This means that a job will not get printed on an "invisible" printer.
2. Only jobs whose class matches the user's class code will appear on the screen when the user is using gspl-pq.
3. Only printers whose class code matches the user's class code will appear on the screen when the user is using gspl-pq.

The effect of all this is that if users' classes are made completely disjoint, then they won't even be aware of the existence of jobs submitted by, or printers assigned to, other users.

If some users have class codes which are supersets of those of other users, it is possible that (for example) "unknown" (to a "subset" user) printers might "pop up" on the job queue section of gspl-pq, or "unknown" jobs on the printer section because the class codes were such that the job was "visible", but not the printer, or vice versa. This is, however, the only situation where a user might notice the existence of jobs or printers not covered by their class code.

By default the class codes are set to all ones (`A-Pa-p`) which has the effect of disabling class codes, making everything accessible to everyone.

4.4.2 User access to class codes.

In general a user may vary the class code assigned to him or her by specifying an option to one of the user programs. The class code can, in general, only be made more restrictive, temporarily narrowing the user's interest to a subset of the jobs or printers normally displayed or processed.

Continuing from the earlier example, if you were only interested in using laser printer 1, which has a class code of `C`, then you could specify the argument `-C C` to various programs such as gspl-pq and gspl-pr. This would guarantee that jobs only went to printers having class `C`, and when using gspl-pq, only jobs for that class are visible. (You can save typing by creating configuration files or environment variables to hold these options; see Chapter 11 for more details).

A special privilege, "Override Class", may be specified, if desired, for a given user or set of users enabling them to widen their field of view by turning bits on as well as off. If a user has this privilege then the class code given by the `-C` option is used throughout the operation of the specified program. Otherwise the bitwise "and" is silently taken between the user's class code and the class code specified, yielding a value for the class code consisting of only those bits in common between the class bits specified and those in the user's class code.

The exception to the "silently" in the above sentence is that the class code is never permitted to be zero and if the bitwise "and" yields zero an error message will be given.

5 Printer Definitions

A Printer Definition consists of a directory containing one or more Printer Setup files. The Printer Setup files contain the parameters which are used for:

- Initialisation when printing starts on a printer.
- Resetting when printing is finished on that printer.
- Setting up to handle different paper and suffix types.

The structure of the printer definitions is described below.

One of the most powerful features of GNUspool is the ability to flexibly specify the strings to be sent and commands to be run to set up and control the printer.

Given the correct parameters in the setup files, you can arrange for any given form type to be supported on any printer without the user having to worry about which printer it is. For example; at Xi Software, we use the paper type of `a4` on four completely different printers. Suffixes such as `.ps` denote postscript output, with conversion, perhaps to PCL, where appropriate, depending on the printer.

Before going any further, here is a list of what you can specify using setup files.

- Baud rate and other line parameters.
- Network commands for terminal server type devices.
- Flags to turn off banners and specify manual feeding (single-job mode).
- Strings to be output or commands to be executed in various circumstances.
- Alignment files, for test output onto pre-printed stationery before printing.
- "Filter" programs to post-process output before printing or to handle non-standard printer protocols.
- Optional per-printer accounting, with a separate named accounting file for a particular printer type.
- Control of banner (header) pages.
- Control of all these according to the form suffix.

5.1 Form Types

The concept of a "form type" in GNUspool is simple, but easily misunderstood. It encapsulates a number of concepts which we have striven to make accord with users' own understandings of printing operations in general. In particular, we have tried to avoid complexities which add nothing to the printing process, or the user's understanding of it.

What we think of in terms of a form type is that it represents

1. a paper type and
2. how to print on it

The 2 pieces of information are distinct; the "form type" is the combination of both.

The point about a paper type is that someone has to physically intervene in some way to change it. For example to change from plain paper to invoice or letterhead paper someone has to physically touch the printer and we think that the operation of GNUspool should reflect this. Accordingly when the paper type is changed, the intended mode of operation is to

1. Tell the printer to halt, or halt at the end of the current job.
2. When printing has ceased, to physically change the paper.
3. Tell GNUspool about the new paper type.
4. Restart printing with the new paper type.

Nothing about this affects other printers and jobs working with the new or old paper type until the printer is ready to accept jobs for the new paper type, when jobs can start being assigned to it.

On the other hand details about printing on a given paper type can vary from job to job and should, if possible, require no intervention by anyone if the software itself can process the job in some way to achieve the desired effect. Examples of the kind of thing covered by this might be:

- Switching to or from "landscape" or "portrait" modes.
- Automatically selecting fonts or character pitches.
- Selecting input or output bins.

provided that the printer allows these options to be selected under program control.

In addition various post-processing "filters" may be selected to perform translations and conversions as the job is printed, for example automatically invoking troff conversion as the document is printed.

To implement this model of form types we provide for the form type to be split into 2 parts:

1. The paper type, which is everything up to the first "`.`" or "`-`" which denotes the physical paper type.
2. The suffix, which is everything after the first "`.`" or "`-`" if any.

For each printer a file is usually created whose name corresponds to the paper type. Within this file, known as the *setup file*, the interpretation of suffixes is given.

It is entirely up to each installation how setup files and suffixes are defined. We recommend that setup files are created so that the same form type, including sets of suffixes, are interpreted on each printer to give similar effects.

5.2 Location and naming of setup files

The setup files for a printer are located in a directory with the same name as the printer type in the directory

```
/usr/local/share/gnuspool/ptrs
```

So for example if you had decided to call your printer `ljet`, then the setup files would be found in the directory:

```
/usr/local/share/gnuspool/ptrs/ljet
```

and so on. The standard installation procedure offers to create setup files suitable for various printers.

The name of the setup file is that of the paper type, without any suffixes. Thus examples of full path names of setup files might be

```
/usr/local/share/gnuspool/ptrs/ljet/default
/usr/local/share/gnuspool/ptrs/epson/standard
/usr/local/share/gnuspool/ptrs/laser/letterhead
/usr/local/share/gnuspool/ptrs/laser/a4
```

The name `default` is special. If GNUspool cannot find a setup file with the same name as the paper type, it will use the file `default` instead. Please note that it must exist (this is a change from previous releases of GNUspool, which assumed some default parameters).

You can change the name of the file used as the default setup file from `default` to some other name by putting an assignment to the environment variable `DEFAULT_FORM` in the master configuration file, `/usr/local/etc/gnuspool.conf`, thus:

```
DEFAULT_FORM=ifnone
```

Alternatively if you want to ban default setup files and insist on a setup file for each possible paper type, you can specify this as null, thus,

```
DEFAULT_FORM=
```

You might want to include an extra file `-Help` summarising the options of paper types and suffixes you provide. This will be displayed to anyone asking for help whilst displaying the form types to load onto a printer, otherwise a list of paper types will be displayed based upon the file names in the corresponding directory.

Because the setup file represents the papertype supported, its name should not contain "`.`" or "`-`". Conversely if you want to put any other information in the directory, it can be made "invisible" by prefixing its name with either of these characters.

5.2.1 The .device file

To specify common material to be included in all setup files, the file `.device` may be specified. This file is prepended to any setup file to be read.

Although this is not enforced, it is most strongly recommended that this file is only used to hold device-specific information such as baud rates, parity, network commands etc, leaving the setup file to deal with document details and suffixes.

The use of a `.device` file does not relax the requirement for having an appropriate setup or default setup file for the paper type.

5.3 Reading of setup files

The setup file is read whenever

- The printer is started
- The form type suffix changes.

However certain hardware related parameters such as baud rate, character size, number of stop bits, parity etc are only dealt with when the printer is started, except where different handling is requested for the banner page.

Thus if you create a setup file, or change one, you should halt and restart the printer to be sure of resetting any parameters which have been changed.

Syntax and other errors in the setup file are reported by appending lines to the file `spshed_reps`, in the spool directory. This file should always be examined, using the X key in gspl-pq if a printer appears to be misbehaving (the state of the printer will in such cases appear on the gspl-pq display as `error`).

5.4 Syntax of setup files.

The setup file contains some or all of the following:

Commands	are setup strings that are sent to the printer at specific points during the printing cycle. These strings contain printer commands issued for activities like: initialisation, pagination and resetting. All setup strings are introduced by the relevant **keyword**.
keywords	select standard options (e.g. baud rate, buffer size), specify parameters (e.g. filters, logfiles) and introduce **commands**.
names	or **symbolic names**, which give names to character strings. This allows a complex or difficult to read string to be specified once and given an easy to understand name. Each time the string is needed in the setup file the **symbolic name** can be used instead.
conditions	which cause any of the others (or nested conditions) to be selected according to the value of the suffix and
comments	which will (hopefully) make setup files more readable.

In the following sections these are explained in full. However in order to show the "flavour" of these files, here is a simple example:

```
baud 2400
ixon
setup '\e[\x42t'

# initialise the printer
```

This file:

- sets the baud rate to 2400,
- turns on x-on/x-off flow control,
- instructs GNUspool to output a specified string when the printer is started up.

In this example there is no conditional treatment of suffixes, therefore this printer ignores all suffixes.

Although the syntax of setup files may seem incomprehensible at first, a few rules should make things clearer.

1. Comments are allowed anywhere white space is allowed. A "#", possibly preceded by white space, starts a comment; everything up to and including the end of the line is ignored.
2. Keywords must be separated by white space (one or more spaces, tabs and/or newlines).
3. A keyword may be followed by a string, a number or nothing.
4. If a keyword has a string following it, there must be an "`=`" after it, and the string starts immediately after the "`=`". (So don't leave any spaces after the "`=`"!).
5. If a keyword is followed by a number, there will be no "`=`", but any amount of white space may precede the number (there must, however, be at least one space or tab!).
6. File names are always specified according to a standard rule. A file name can either be *absolute* starting with a "`/`", in which case it has its usual meaning, or *relative* with no leading "`/`", in which case it is relative to the directory containing the setup file.

   Also, if a file name in the *printername* directory starts with a "`.`" or with a "`-`", then it will not be among the form types displayed by gspl-pq and similar programs when interrogating possible form types, and thus will be "hidden" from gspl-pq.

5.5 Setup file keywords.

The following sections explain the different keywords which control the treatment of the form type. There are keywords which apply to all printers, just those on certain Unix devices, variations for System V and BSD versions of Unix.

The Complete set of Keywords is:

Keyword	**Function**
`abort`	Command or actions on abort print
`align`	Set alignment page
`banner`	Select special options for head banner pages
`bannprog`	Set alternative banner program
`baud` *n*	Set baud rate to *n* (default is 9600).
`canhang`	Warn that output device may not respond to close attempts.
`cbreak`	CBREAK output mode
`charge` *n*	Set charge rate to *n* (default is 1000)
`clocal`	Ignore status of carrier detect line
`close`	Specify number of seconds to wait for I/O completion
`closekill`	Specify signal number to kill terminal server if I/O not completed
`crmod`	Map CR to LF on input, LF to CR+LF on output

`cs5`	Character size 5 bits
`cs6`	Character size 6 bits
`cs7`	Character size 7 bits
`cs8`	Character size 8 bits
`delimiter`	Set page delimiter
`docend`	Command sent to printer at end of document
`docstart`	Command sent to printer at start of document
`even`	Even parity generation & checking
`extabs`	Expand tabs
`exec`	Force execution of command string as a shell script
`execalign`	Set alignment program
`exit`	Specify network filter exit handling
`fberror`	Feed back error, capture standard error for display
`filter`	Set filter command
`forcehdr`	Force header page even if user turns it off with the suppress header page option
`halt`	Command sent to printer when halted
`hdrpercopy`	Prefix each copy of multiple copies with the header page
`ibaud` *n*	Set input baud rate to *n* (default is 9600).
`inclpage1`	Always print the first page (which may perhaps contain formatting/font information) when a range of pages other than the whole job is to be output.
`ixany`	Allow any character to restart output
`ixon`	Enable X-ON/X-OFF flow control
`lcase`	Map upper case to lower case on input, lower to upper on output
`logerror`	Copy diagnostics from terminal server processes to system logfile
`logfile`	Set text log file
`network`	Set network filter
`nohdr`	Turn off banner pages
`noranges`	Treat jobs as single unit, ignoring page range requests
`obaud` *n*	Set output baud rate to *n* (default is 9600).
`ocrnl`	Map carriage return to linefeed
`odd`	Odd parity generation & checking
`offline`	Set off-line timeout to *n* (default is 30)
`olcuc`	Convert lower to upper case
`onlret`	Linefeed does carriage return
`onecopy`	end only one copy to printer, even if more requested
`onlcr`	Map linefeed to carriage return linefeed
`onocr`	Don't do carriage return at start of line
`open` *n*	Set open timeout to *n* (default is 30)
`outbuffer` *n*	Set output buffer size to *n* (default is 1024)

`parodd`	Odd parity generation
`pageend`	command sent to printer at end of each page
`pagestart`	command sent to printer at start of each page
`parenb`	Even parity generation
`portsetup`	Command to run prior to opening device
`postclose`	Specify seconds to wait at end of each job
`raw`	Raw mode output
`record`	Set binary record file
`reopen`	Close and reopen device (or network filter) between jobs
`restart`	Command or actions on restart print
`retain`	Force jobs to be retained on queue
`seterror`	modifier for exit and signal keywords
`setoffline`	modifier for exit and signal keywords
`setup`	command sent to printer at start up
`signal`	Specify network filter signal handling
`single`	Select single job mode
`stdhdr`	Reset header options to standard
`stty`	Invoke command after opening device
`sufend`	Command sent to printer at end of suffix
`sufstart`	Command sent to printer at start of suffix
`tabs`	Same as extabs
`tandem`	Set input flow control
`twostop`	Two stop bits instead of 1
`width`	Set page width to *n* (default is 80)
`windback`	Wind back *n* pages on interruption

5.5.1 Line (or device) parameters.

The following keywords set the baud rate and other hardware and line parameters. The keywords used are the same as the corresponding parameters to the stty command. They must be delimited by white space (space, tab or newline).

These parameters are all ignored for printers run via the network command.

If none of these keywords are used, the line will be set to a default state:

- Assumed serial
- 9600 baud
- 8 bits/character
- no parity or other options set.

We strongly do not recommend that this be relied on, and for this reason we no longer support printers without setup files.

5.5.1.1 Parameters for GNU/Linux, Unix SVR4 or SVR3 machines

Systems based on the System V Unix or GNU/Linux use different parameters to those based on BSD Unix. The table shows those for System V.

Keyword	**Function**
`baud` *n*	Set baud rate to *n* (default is 9600).
`parenb`	Even parity generation
`parodd`	Odd parity generation
`cs5`	Character size 5 bits
`cs6`	Character size 6 bits
`cs7`	Character size 7 bits
`cs8`	Character size 8 bits
`twostop`	Two stop bits instead of 1
`olcuc`	Convert lower to upper case
`onlcr`	Map linefeed to carriage return linefeed
`ocrnl`	Map carriage return to linefeed
`onocr`	Don't do carriage return at start of line
`onlret`	Linefeed does carriage return
`ixon`	Enable X-ON/X-OFF flow control
`ixany`	Allow any character to restart output
`clocal`	Ignore status of carrier detect line
`extabs`	Expand tabs
`tabs`	Synonym for extabs

Options set earlier on in a file may be turned off by preceding them with a minus sign where there is only an "on" or "off" state (i.e. all keywords except baud and the character size keywords).

The `OPOST` flag is turned on wherever necessary for the handling of options beginning with o. If `clocal` is specified, the device is first opened in *no-delaymode* prior to setting the parameters.

5.5.1.2 Parameters for BSD Unix machines

If you are using a version of Unix based on BSD (Berkeley Software Distribution) Unix, your set of line parameter keywords will be different, since the *ioctl* interface available is different:

Keyword	**Function**
`baud` *n*	Set input and output baud rate to *n* (default is 9600).
`ibaud` *n*	Set input baud rate to *n* (default is 9600).
`obaud` *n*	Set output baud rate to *n* (default is 9600).
`even`	Even parity generation & checking
`odd`	Odd parity generation & checking
`raw`	Raw mode output

`crmod`	Map CR to LF on input, LF to CR+LF on output
`lcase`	Map upper case to lower case on input, lower to upper on output
`cbreak`	CBREAK output mode
`tandem`	Set input flow control
`extabs`	Expand tabs
`Tabs`	Synonym for extabs

As with the System V parameters the purely "Boolean" options may be turned off by prefixing with a minus sign.

5.5.2 Terminal server printers.

The above baud rate etc commands are applicable to serial lines, and on many systems work adequately (or are ignored) on parallel ports. Output may optionally be sent to FIFO (named pipe or socket) type files and in general these commands will be ignored.

For other types of output device, particularly terminal servers, or where the output needs special treatment, such as invocation of a network command, there is provided the *network filter* command. This may need a little more insight as to the requirements of your particular machine.

The term *network filter* may be something of a misnomer; indeed it may give rise to confusion with networked machines. However most of the time it is used for this purpose, and has been kept to this for historical reasons.

To set up a networked printer, you must create the entry:

```
network=<command>
```

Additionally you must put an "address" as the "device name" of your printer in gspl-pq, by enclosing it in angle brackets `<>`.

It is a mistake not to do both of these things, if you try to start the printer without setting both of these up then the printer will revert to `error` state, and an appropriate message will be written to `spshed_reps` (you can view this with the `X` command in gspl-pq).

The command given is one which will accept on standard input a data stream and pass it to the remote machine. This will be used in place of any device. The address entered in `<>`s in gspl-pq will be passed in the environment variable `SPOOLDEV`, which could, of course be on the command line supplied. For example, if the command is called `sendtonet` (this is a completely fictitious name) which takes a host or port address as its argument the `network=` line would look like this:

```
network=sendtonet $SPOOLDEV
```

The command should produce an exit code of zero. Other exit codes will cause the printer to revert to error state with an appropriate message being written to `spshed_reps`. Don't forget that this command can be a quite arbitrary shell script.

As far as all other aspects of the setup file go, the network command is treated as an output device, with `docstart` etc strings being sent to it. Commands to set the baud rate etc will be ignored (except that they will be syntax checked).

Typically the value of `SPOOLDEV`, which the user inserts in angle brackets in the gspl-pq screen, will be a host name and/or internet port number. The exact syntax and use of the commands is beyond the scope of this manual, as it varies according to the details of the network and terminal server system which you are using. Please refer to your system manuals for details of how to send data to your server.

Please note that we do provide a variety of interface programs for common terminal server types.

5.5.2.1 Trapping of exit codes, signals and reports

When a network server type printer is in use, by default an exit code of 0 means normal exit, and any other exit code reflects an error state, and this is displayed against the printer with state halted.

Likewise, again by default, termination with a signal of `SIGHUP` (hang-up) is taken as an off-line indication, and any other signal is treated as an error.

This behaviour may be adjusted by specifying alternative range of signals and/or exit codes as follows:

```
exit setoffline 2,10-14,200-255
exit seterror 5-9,15-199
signal setoffline 1-3
```

Exit codes not "covered" are treated as normal, signals not "covered" are treated as error.

In addition, it is helpful, in cases where the interface is "clever" enough to send information back, to be able to display what is going wrong on the gspl-pq screen, perhaps in the form

error:Paper Jam
offline:Out of toner

If this can be detected, the network filter process should output this on standard error. The first 40 characters of the final line so output are captured and displayed. To signify that this should be done, include a line holding just the keyword

```
fberror
```

for "feedback error" in the setup file.

The keyword `logerror` is similar, but causes the standard error to be written to the system error log file `spshed_reps`. This may be used as well as or instead of `fberror`.

5.5.3 General Keywords.

The following are those for all machines, whether GNU/Linux, System V or BSD based. Purely "Boolean" options may be "turned off" by preceding the keyword with a minus sign.

5.5.3.1 Reopening the printer.

On some system, printer and/or interface combinations, the only way to "flush" the printer buffer at the end of a job is to close then re-open the port at the end of every job.

If your printer behaves like this, use the keyword

```
reopen
```

This causes GNUspool to close and re-open the port after every job has been printed.

If used in conjunction with the `network` keyword, then this will cause a new command to be issued for each print job. You will almost invariably want to include this in such cases.

5.5.3.2 Open timeout.

Depending on the model of printer and how it is connected, it may be possible for your machine to detect whether or not the printer has been switched on and is operable. If your system "hangs" when trying to open a printer which is inoperable, it may be useful to tell GNUspool to "time out" the open if the printer does not respond within a specified number of seconds; this is done with the open keyword:

```
open 5
```

means set the printer state `offline` if the open does not complete within 5 seconds. If no open keyword is used, the default value is 30 seconds. You may turn this off altogether with a value of 0.

It is advisable to err on the generous side when setting this parameter; allow a fair margin for error to deal with heavily loaded systems and borderline cases. If this parameter is set too small, there may be panic among operators who think a printer is "dead" when in fact the whole system is just too overloaded to respond in time.

5.5.3.3 Close timeout

This is only applicable to terminal servers. After sending the last of the data to the terminal server process, the despooler spd closes the "pipe" to the process performing the output to the server, and then waits for the process to exit.

It may take some time for the process to exit. In some cases, it may not exit until the whole job is printed. In other cases it may return quickly and be ready for further printing, and any delay indicates some kind of error.

To give the user control over this the `close` timeout is provided. The `close` keyword gives a number of seconds for the spd process to wait before automatically killing the process with `SIGINT` (interrupt signal), and then waiting a further such period before killing with SIGKILL. Thus:

```
close 3600
```

would leave a 1-hour delay.

As a further refinement, the initial kill signal number may be specified, so as to be different from `SIGINT`, by using the `closekill` keyword. Thus:

```
closekill 15
```

will change the initial signal to 15 (`SIGTERM`).

To turn off the timeout, and wait indefinitely, use a zero parameter to `close`.

5.5.3.4 Post-close wait

Even after the terminal server process has finished, especially on larger jobs, it may be some time before the terminal server will permit another connection to print another job. GNUspool is often too fast for terminal servers to cope with. If this is a problem, then insert a "sleep" immediately after printing each job with:

```
postclose 5
```

The number gives the number of seconds to "sleep" for.

Tune this parameter so that the sleep is not too long, but you do not get "connection refused" messages.

Many of the terminal server drivers provided by Xi Software include options to make several retries at a given time apart which obviate the need for this and are much quicker where it is not needed.

5.5.3.5 Off-line timeout.

To cope with situations where it is impossible to detect a printer going off-line, GNUspool will "time out" any output to the printer, and if no output is possible within the specified time limit, the printer state will be set to offline. The format is:

```
offline 10
```

which sets the printer state to offline if no output is possible within 10 seconds. The default value is 30 seconds, which should be adequate for all normal circumstances. As with the `open` parameter, generosity is urged here. The bigger the buffer on a printer the greater the offline timeout may need to be.

5.5.3.6 Buffer size.

To allow for varying buffer sizes and also to assist in off-line detection, it is possible to set the output buffer size used by spd. Do not confuse the spd buffer with the printer buffer. The default size is 1024. If there are problems with off-line detection, it may be useful to decrease the buffer size. If you are worried about efficiency and speed of printing, increase it. The keyword to use is outbuffer, e.g.:

```
outbuffer 4096
```

will set the buffer size to 4 Kbytes, which may be useful for fast printers.

If output filters are being used, the buffer size should probably be set to a multiple of 1024 bytes, as most programs read and write pipes in multiples of this size.

The output buffer size will affect the rate at which the `Posn` display in the main gspl-pq screen is updated.

5.5.3.7 Windback pages.

There is a problem with buffers on the printer in that the printer may decide that it is off-line some time after it has ceased printing properly. GNUspool will normally restart printing at the page it thought was interrupted. However you can automatically "wind back" a given number of pages by specifying this command, thus for example:

```
windback 3
```

will move back automatically 3 pages if the printer is halted in the middle of a document, or to the beginning of the document if the current page was 3 or less. (N.B. you can specifically reset the "halted at" page with gspl-pq).

If you always want the document to be restarted from the beginning, specify a huge number such as 100000.

5.5.3.8 Inclpage1 keyword

Sometimes the first page of a document contains formatting, macro or font information and it is important to include this even if a required range of pages does not include the first page.

The `inclpage1` keyword indicates that the first page should always be printed.

5.5.3.9 Canhang keyword.

With some devices it is impossible to complete a close(2) system call if there has been an error. This may mean that if, for example, a printer goes off-line, then the spd despooler process may "hang" in a state where a "kill" command has no effect. This can have potentially disastrous effects if the scheduler process spshed then believes that the spd process is still running.

The `canhang` keyword attempts to overcome this by making sure of the fact that it is the *last* process to close the device (or to exit or be killed which both imply a close) which is affected.

This keyword causes spd to fork off a new process and then to close the device before the forked-off process attempts to. The forked-off process closes all other files etc before attempting to close the device.

This may also be used with terminal server processes in the event that they hang "immortally" until whatever has caused the problem is resolved or the system is rebooted.

5.5.3.10 Noranges keyword

Sometimes page range handling must be switched off, for example where graphics data is being sent and the page breaks are only apparent. In such cases the job must be restarted from the beginning and any start, end or "halted at" page ignored.

To achieve this, include the `noranges` keyword in the setup file.

5.5.3.10.1 Banner pages

Several keywords and options control how banner pages are generated.

5.5.3.10.2 Turning off banner pages and forcing them on.

A banner page is normally printed at the start of each job, but only prior to the first copy of multiple copies printed together. The user may opt to suppress the banner page with the `-s` option to gspl-pr, or by changing this option within gspl-pq or gspl-

`xmpq` prior to printing.

Various keywords and options change this behaviour, for example it is undesirable to have banners printed on labels or expensive stationery.

- The keyword `nohdr` turns off banner pages for the specified form type, regardless of any options specified by users.
- The keyword `forcehdr` turns on banner pages for the specified form type, regardless of any options specified by users.
- The keyword `hdrpercopy` causes a banner page, if it is being printed, to be put in front of each copy, even if the copies are being printed together.
- The keyword `stdhdr` restores the default handling of banner pages where this is required for a particular form type suffix. For backward compatibility with previous versions of GNUspool, this is also achieved by the sequence "`-nohdr`".

5.5.3.10.3 Changing the format of banner pages

If you do want banner pages, but don't like our way of printing them, then you can specify a shell command to do it. This command is given by using the construct

```
bannprog=<shell command>
```

The command is invoked with standard output and standard error sent to the output device or network command, and with the following parameters:

1. Job number as decimal digits.
2. User name of submitting user.
3. User name to be posted to.
4. Size of job in bytes.
5. Time submitted as decimal seconds since 1/1/70 (this is easier to parse in a shell script or C program than dates and avoids arguments about date formats and time zones).
6. Priority of job as decimal digits.
7. Organisation name licensed to.
8. Serial number.

Don't forget that the program will have other parameters such as the form type and job title available in the environment variables `SPOOLFORM`, `SPOOLHDR` etc. These are further documented below.

If a *filter command* is specified, then this will not be invoked in addition to the banner program. You should include this in your program and/or script. However if `docstart` and/or `docend` strings are given, or strings specific to the banner as described in the next section, then these will be sent or executed before and after the banner respectively.

Note that for the purpose of keeping a log of the number of pages sent, 1 page is assumed to be generated by the banner program. Also note that with Release 23 there are subtle variations in the handling of the `bannprog` keyword, which are discussed later.

5.5.3.11 Banner parameters.

There are 2 additional `docstart` and `docend` strings available specific to printing the banner. By default these are set to be identical to the "main" strings, and they are output before and after the banner page. (Remember that you will get a formfeed if no `docend` is specified).

If you want to specify a different set for the banner, then you should bracket appropriate commands between the lines

```
banner
```

and

```
-banner
```

For example:

```
banner
docstart'......'
docend '.....'
-banner

# "Main" docstart etc

docstart '.....'
```

If all you want to do is to "turn off" the strings that you specified for the "main" document, then you should just use an empty set of commands thus:

```
banner
-banner
```

Note that this feature only applies to the `docstart` and `docend` strings. The kind of thing this is useful for is where you want the banner printed in portrait format and the rest of the document printed in landscape for a given suffix (this is how we print documents in landscape format).

5.5.3.12 Banner line parameters.

The line parameter settings (baud rate etc) are by default copied and used in the banner. However, if you put line parameters such as `onlcr` into the set of commands delimited by `banner` and `-banner` then these will apply to the banner only.

Line parameters given up to the point where you start to specify them after the `banner` keyword are first copied, so you should put first any parameters unlikely to change, such as the baud rate.

For example:

```
baud 19200

banner
```

```
onlcr
-banner

# No onlcr in main document
```

Remember that you will have to re-specify the `docstart` and `docend` strings if you do this, as described in the previous section.

5.5.3.13 Single job mode.

The keyword `single` causes the single-job mode of operation to be selected. In this mode each job causes GNUspool to halt the printer in `a/w oper` mode, for the operator to restart it using the y command against the printer line in gspl-pq.

It is intended

1. for devices, like plotters, where new paper has to be loaded before each job,
2. for form types where the first page is different in some way from subsequent pages and requires manual intervention, e.g. loading the first sheet manually, as in a series of letters where the continuation paper is loaded, and a front sheet at the start of each letter.

5.5.3.14 Retention.

The keyword `retain` causes all jobs to be retained on the spool queue until explicitly removed. The number of copies is set to 0. This is equivalent to the `-q` option of gspl-pr, but it cannot be overridden by the user.

The two major uses of this feature are:

- critical output, such as cheques, can be validated before final deletion from the spool queue
- jobs which may require re-printing can be left on the queue, to be re-printed using gspl-pq rather than having to re-submit another job. Additional copies can be printed by setting the number of copies parameter to a non-zero value.

5.5.3.15 Onecopy

The `onecopy` keyword restricts GNUspool to send only one copy to a printer, regardless of how many copies were requested. It is provided for the following requirements:

1. To take advantage of multiple copy printing where a printer, or printer controller is smart enough to handle requests for multiple copies. This may reduce machine utilisation or network traffic. Normally the print job would be sent to the printer, by GNUspool, as many times as there are copies requested.
2. When specifying multiple copies to a remote print server, such as an lp emulator.
3. To stop multiple copies being easily printed on expensive or commercially

sensitive media, such as high quality colour pictures or bank cheques.

The environment variable `SPOOLCPS` holds the number of copies actually requested. For use with a smart printer include the onecopy keyword in the printer setup file. Then pass `SPOOLCPS` to the printer or controller to specify how many copies to print.

5.5.3.16 Page width.

The keyword `width` *n* sets the page width to *n* characters (the default is 80). This is used to control the size of the banner pages.

Note that GNUspool does not truncate or fold lines; this parameter only affects the banner page.

5.5.3.17 Page length delimiter.

The page range option of gspl-pr is by default interpreted by noting form-feeds in the spool file. Thus if pages 3 to 20 are to be printed, then the file is ignored up to and including the 2nd form-feed and after the 19th.

To use some other string as the page delimiter, or to count delimiters, specify the delimiter keyword followed by an optional number and a string, for example:

```
delimiter "****"
```

which means pages are delimited by four asterisks, or:

```
delimiter 66 '\n'
```

which means that the page is delimited by counting multiples of 66 linefeed characters. The string can also be specified as a symbolic name, or explicitly as characters in double or single quotes, or angle brackets `<>`.

If a delimiter specification was given to gspl-pr using the `-d` and `-D` options, then the specification in the setup file this will override and replace the delimiter, and any page offset file regenerated. This may cause the display of the details of the job on the gspl-pq screen to change.

5.5.3.18 Print charges.

You may change the charging rate for the printer and form type by specifying the `charge` keyword, e.g.:

```
charge 2500
```

The charging rate is divided by 1000, the default, to give the rate at which output is charged, thus the above example charges output at 2.5 the default rate.

This can be used for special stationery and/or printers, to reflect the true cost of printing.

The charging information can be inspected using gspl-charge.

5.5.3.19 Record of output.

It is sometimes useful to record the total number of characters output on a printer.

This can be useful for estimating lifetimes of ribbons or toner cartridges.

To enable recording of character counts, use the record keyword, giving GNUspool the name of a file in which to record the number, and the byte offset within the file at which to record it, e.g.:

```
record 0:.totals
record 100:/usr/spool/record
```

The part of the argument before the colon, is the byte offset, the name following is the pathname of the file.

This causes GNUspool to update the file after every job by reading a `long int` (to use C terminology) from the file at the given offset, adding the number of characters printed to it and rewriting it at the same offset.

GNUspool contains no facilities for printing or further processing this data; it is intended that administrators requiring this facility will provide their own tools for this purpose.

A caveat; there is no interlocking performed on the record file; if it is required to zero the record, you should be sure that GNUspool is not running while this is done.

5.5.3.20 Log files.

To allow more detailed accounting than is available using the standard GNUspool charging mechanisms, the keyword `logfile` may be specified to cause a record of each job printed to be appended to a given file, for instance:

```
logfile=.usage
```

The format of the log file is single lines of text of the form:

```
09/06|11:52:18|11:56:53|0:05:13|jmc|CS|a|947|tty1|laser|150|3|14408
```

The fields are respectively; date (`dd/mm` or `mm/dd`, the latter if the current timezone is more than 4 hours West), time started, time finished, elapsed time (since job enqueued), user, title, form type, characters printed in bytes, device, printer and priority, the number of pages including 1 for the banner. The last field is the print job number or host name and job number if the job did not originate on the current machine.

It is assumed that utilities outside of GNUspool will process this file. (E.g. grep or sed).

As an aid to future compatibility, it should be noted that all the fields are separated by "`|`"s; this format will be maintained through future releases of GNUspool, as will the contents of the current set of fields defined above. New fields may, however, be added to the end of the line, again separated by "`|`"s. Thus, utilities which process log files should ignore any trailing fields to avoid future compatibility problems.

5.5.3.21 Alignment pages.

Two alternative features provide for the printing of alignment or test pages. The keyword align as in:

```
align=.setpage
```

provides for the contents of the file `.setpage` to be output to the printer before the first job with the relevant form type is printed. Typically it would contain blocks of characters to print in boxes on pre-printed stationery or labels. The file is taken from the same directory as the setup file if it is a relative pathname (i.e. not starting with a `/` character).

The `docstart` and `docend` strings precede and follow the file as it is sent to the printer, and the file is passed through any specified `filter` command.

An alternative to specifying an alignment file is to specify a command to run by using the execalign keyword, as in:

```
execalign=myalign
```

The job number of the next job to be printed will be given by the environment variable `SPOOLJOB` environment variable and the environment variable `SPOOLHOST` will be set to the hostname if the job is a remote one, and otherwise null. The current directory will be the spool directory. Any filter command is not applied but the `docstart` and `docend` strings are.

The action of processing an alignment page is to process the alignment file or command and to put the printer into `a/w oper` state. The alignment process is repeated every time the user disapproves the alignment page on the printer list of gspl-pq or by running the gspl-nok command and until the printer is either halted or the alignment page is approved.

If the file specified in the `align` keyword does not exist, is inaccessible to user gnuspool, or the command given in the execalign statement does not run successfully, then alignment processing will be silently bypassed.

5.5.3.22 Port Set Up.

Sometimes you cannot open the device or network until you have run some command. Several terminal servers require the user to map a `/dev/ttynn` entry to a network address and *then* to open it.

The `portsetup` command allows you to run an arbitrary shell command the first time the printer is started and before any attempt is made to open the device. Note that the device name is available, as `$SPOOLDEV`, the printer name as `$SPOOLPTR` and the form type as `$SPOOLFORM`. Other environment variables (job title etc) will not be defined at this stage.

Note that the command is run without regard to error messages or codes; so do test the command first. Also note that the command will run under effective and real user id `gnuspool`; possibly only `root` will be able to run some commands. You can run GNUspool using user id `root`, but most people dislike the security implications of this; it is better to run a set-user to root C program to invoke the desired command given as arguments.

Here is a suitable program:

```
#include <pwd.h>

main(int argc, char **argv)
{
    struct  passwd *pw;

    if(argc < 2)
        return 1;

    if (!(pw = getpwnam("gnuspool")))
        return 2;

    if  (pw->pw_uid != getuid())
        return 3;
    setuid(0);
    execv(argv[1], &argv[1]);
    return 4;
}
```

5.5.3.23 Filter commands.

Filter commands allow you to invoke any form of post-processing of spool job files. Examples of what you can do (and we do at Xi Software) are:

- Download fonts onto laser printers dynamically and transparently without the user having to know which are resident and which are not (apart from the downloading time making this obvious, perhaps).
- Invoke processors such as *troff*, ghostscript etc. with the correct options.
- Produce letters, wrapping a graphics logo and letterhead around each page.
- Extract the address from a letter and run gspl-pr in parallel to produce a label at the same time.
- Automatically format and insert the date into a document, and the name of the author from the `/etc/passwd` file.
- Translate certain characters, i.e. currency signs, to handle variations in printer character sets.
- Insert something at the start of each line, for instance line numbering, or indentation for awkward printers.

This is done by means of the `filter` keyword. For instance, if your printer only has an upper case character set, this will translate all output to upper case:

```
filter=tr '[a-z]' '[A-Z]'
```

Effectively what happens is that the shell is used to execute

```
(command) <spoolfile >printer 2>mailfile
```

The command can be a shell script or a pipeline of commands, for example to print out *MM macro* files on a *HP LaserJet* compatible printer, we use:

```
filter=troff -mm -Thp|dhp|hpfinsert
```

where `dhp` is our *troff* back end for these printers, and `hpfinsert` is a filter which interprets font selections and downloads soft fonts.

If the filter command generates any output on its standard error output, it will be mailed back to the user. Likewise if the filter command terminates with an exit code other than zero, then a note about this will be mailed back to the user. (Beware of the fact that some Unix commands generate strange exit codes. For example, some versions of tr generate a random exit code despite correct usage).

Eleven environment variables are passed to the filter command:

Name	**Contains**
`SPOOLFORM`	The entire form type (both main part and suffix), e.g. `a4.troff`
`SPOOLUSER`	The user ID of the originator expressed as a decimal digit string e.g. `SPOOLUSER=192`.
`SPOOLHDR`	The job title. Note that this may not be the same as the file name being printed. This may include spaces.
`SPOOLFLAGS`	Any flags explicitly passed to the filter from gspl-pr by the user using the `B` option of gspl-pr.
`SPOOLPTR`	The printer name
`SPOOLDEV`	The device name including any leading `/dev`. Or the network address given in `<>`s for terminal servers.
`SPOOLJUNAME`	The user name (characters) of the user who submitted the job.
`SPOOLPUNAME`	The user name (characters) of the user to whom the job is to be posted (as specified by the `-u` option to gspl-pr).
`SPOOLJOB`	The job number (not zero-filled) of the job being printed.
`SPOOLHOST`	Null if the job originated on the current machine, otherwise the host name of the machine from which the job did originate.
`SPOOLPAGE`	A numeric page number; however this will be an undefined value when the filter is started.
`SPOOLRANGE`	Null if all pages have been requested. A page range is specified in the form `m-n`. The range `-n` indicates all pages to `n`, similarly `m-` indicates all pages from `m` to the end.
`SPOOLCPS`	Number of copies requested. Decremented for each copy.
`SPOOLOE`	Set to 1 if printing only odd pages, 2 if only printing even pages or otherwise set to Null.

Do not confuse these with the environment variables set up in `/etc/gnuspool.conf`.

If any other environment variables should be passed, such as timezone (`TZ`), or search path (`PATH`), these should be specified in the startup of spshed.

5.5.3.24 Advanced filtering.

Since the filter is connected directly to the printer, it may be more than just a simple filter.

Possibilities include

- Handling strange communication protocols, such as ETX/ACK,
- Detecting off-line or similar errors in ways unknown to GNUspool.

If a filter detects an error on the printer, it can set the printer status to `offline`. This is done by sending a hang-up signal (SIGHUP) to its process group. This works because spd (and all its children including the filter process) exist in their own process group; spd catches `SIGHUP` and assumes it means the printer has gone off-line, or is in some other way unusable.

In C, the code to send the signal is:

```
signal(SIGHUP, SIG_IGN); /* ignore it ourselves */
kill(-getpgrp(), SIGHUP);
```

If it is allowed for the filter to die too (which it often is), you can omit the signal call, and the filter will die along with everything else.

If you specify a `filter` command on a networked printer, then data will be passed from GNUspool to the filter, and from the filter command directly to the networked printer.

5.6 Setup file strings

To define strings of characters, most commonly used to send character sequences to the printer, a standard string syntax is provided.

Strings consist of a sequence of printing characters, with certain special sequences to denote non-printing characters detailed below. Tabs are not allowed within strings except by means of an escape sequence, and have the effect of terminating the string (be careful not to let your editor blindly expand or insert TAB characters).

All strings which are part of commands, such as setup or docend, should be enclosed in quotes `'` or `"`, or angle brackets `<>`.To use the closing character within the string it should be preceded by a `\`, for example:

```
<string with a \> in it>
```

Strings used in definitions are different; see the next section.

The following escape sequences are used to produce non-printing characters. They are ineffective where the `exec` keyword is used to denote that the string is to be treated as part of a shell command, as described below under commands, where they are considered inappropriate.

Escape sequences are all introduced by the characters \ and ^.

`\` followed by `^`, `\`, `b`, `e`, `s`, `t`, `v`, `r`, `n` or `f` may be used to denote `^`, `\`, *backspace*, *escape*, *space*, *tab*, *vertical tab*, *carriage return*, *linefeed* or *form-feed* respectively. `\` followed by 0 and up to 3 octal digits or by x and up to 2 hexadecimal digits may be used to generate a character with that value.

To make long strings more comprehensible, `\` followed immediately by the end of line is ignored, allowing long strings to be continued over multiple lines.

`^` followed by `@` and `A` to `Z` and `[` to `_` may be used to denote the corresponding control character, and `^^` is used to denote a single `^`.

Upper or lower case letters may be used in sequences introduced by `\` or `^` characters.

For example, on one line printer, to set the page length to 66, the binary value had to be output, preceded by escape `[`, and followed by `t`; a suitable string would be

```
setup '\e[\x42t'
```

(Hex 42 is 66 decimal). Alternatively, if you prefer octal to hex, this is equivalent:

```
setup '\033[\0102t'
```

5.7 Symbolic Names.

Using the same escape sequence in several places is error-prone and hard to understand. To avoid this an arbitrary symbolic name can be given to each string by using assignments such as

```
SET_PAGE_LENGTH=\e[\x42t
```

The name given must not clash with a keyword. The name is followed by an equals sign and then immediately by the string. The string may include spaces, but tab characters must be represented by \t (or by ^I if you prefer). A newline or a tab character itself terminates the string. Note that no quotes are required, in fact, if they are used, they will be included in the string.

Names may be redefined any number of times within the setup file. To "undefine" a variable, set it to null, e.g.:

```
SET_PAGE_LENGTH=
```

The last preceding definition is used when the name is used in a command. It is an error to use an undefined identifier in a command.

The following are additional examples of valid definitions:

```
EXPAND=^N
TABSET=\e(009,017,025,033,041,049,057,065,073)
```

Definitions do not in themselves cause anything to be output, but they enable the identifiers thereby defined to operate in commands as a "shorthand" replacement for a string in quotes.

Note that `\` and `^` sequences are expanded when the definition is read, not when it is used, so care is needed when definitions are being used in complex shell command lines.

5.8 Commands

The setup file can define 8 strings, or *commands*, to be sent to the printer or executed as shell commands at certain specific points during the print cycle, and two further ones for exceptional cases. In each case the *command* can either send a string to the printer, or generate a shell command, but not both. Shell commands are executed and their output sent to the selected device.

The *commands* are like "layers" in the printing process onto the beginning or end of which to "hang" various operations. The *commands* are:

Command / layer	**When sent or executed**

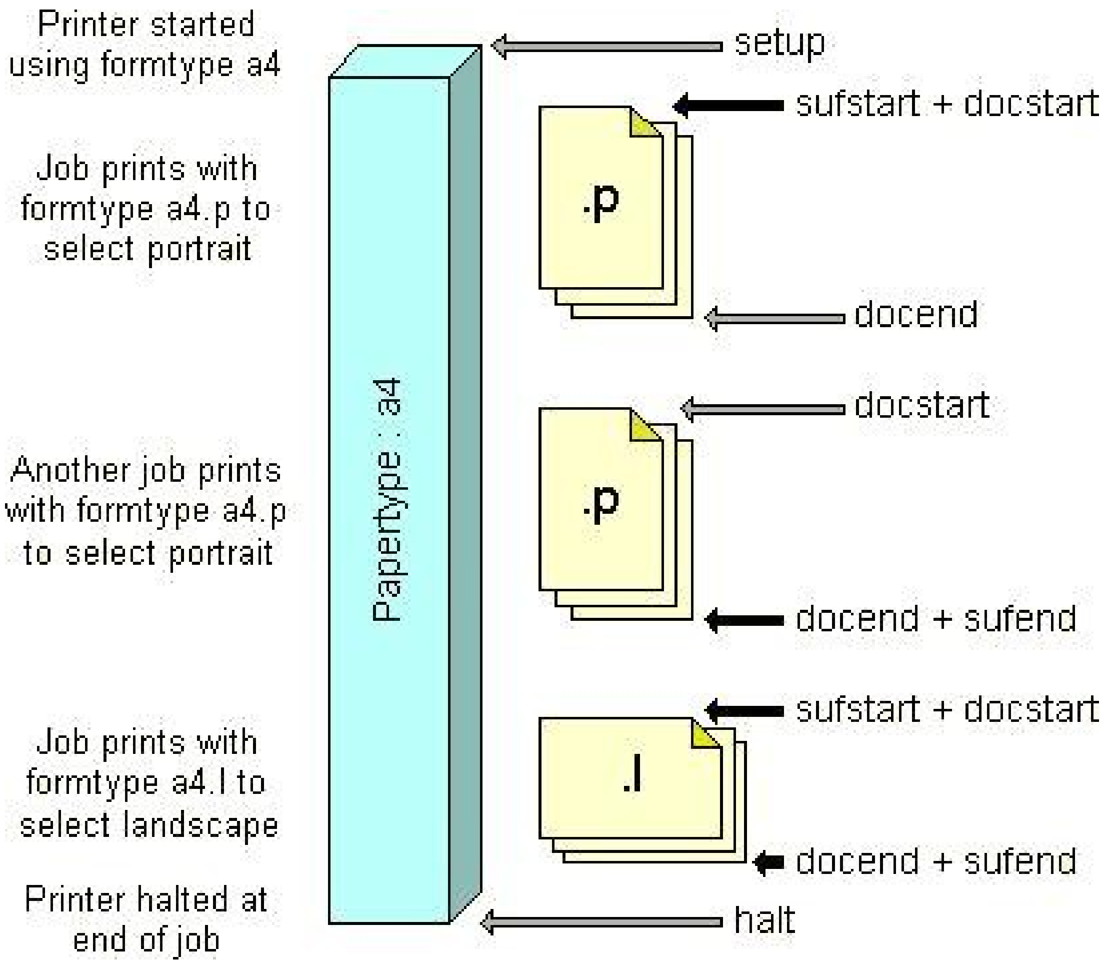
`setup`	Printer Startup
`. sufstart`	Suffix Selected (i.e. this job has a different suffix to the last)
`. . docstart`	Start of every document
`. . . pagestart`	Start of each page
`. . . pageend`	End of each page
`. . docend`	End of every document
`. sufend`	Suffix de-selected (i.e. next job has a different suffix)
`halt`	Printer Shutdown
`abort`	Action or string to be output when job aborted or deleted during printing
`restart`	Action on job restarted from beginning.

Each of the operations represented by the commands is purely optional. By default they are set to null, with the exception of the `docend`, `abort` and `restart` keywords. The `docend` string defaults to a single formfeed if not specified, unless a `filter` command is also specified.

In the case of the abort and restart keywords, the default action is to output the strings `***ABORTED***` or `***RESTARTED***` respectively (these strings are read from the message file, by default `/usr/local/lib/gnuspool/help/int-config`, which may be edited if necessary), followed by enough delimiters to end the page.

The keyword `exec` is used to specify that a string is to be executed as a shell command, not sent as a string. Although normally put immediately following the command keyword, `exec` can appear anywhere in the string. The value of this is explained later.

To illustrate how the commands work consider the following simple example. There are 3 print jobs to be printed on `a4` paper. The first two jobs require portrait printing and the third will be in landscape format. The printer is initially halted and is set to the `a4` formtype. The printer will be started, print each of the jobs and then be halted.

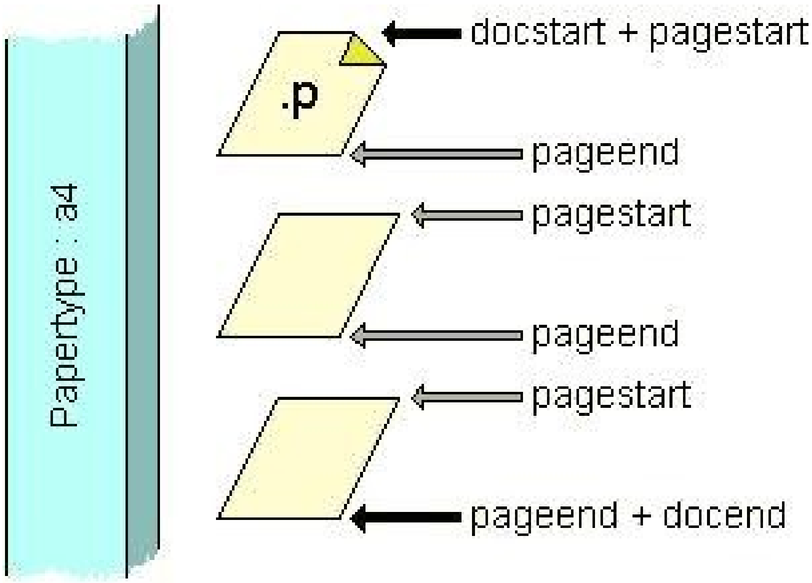

Printer started using formtype a4
Job prints with formtype a4.p to select portrait
Another job prints with formtype a4.p to select portrait
Job prints with formtype a4.l to select landscape
Printer halted at end of job
Papertype : a4
setup
sufstart + docstart
.p
docend
docstart
.p
docend + sufend
sufstart + docstart
.l
docend + sufend
halt

The diagram shows the sequence of commands, excluding `pagestart` and `pageend`, that are sent to the printer. The dark arrows show where more than one command is sent consecutively. The print jobs have the same papertype with different suffices, `a4.p` for portrait and `a4.l` for landscape.

This processing may well cover many of the cases of pitch, orientation and font selection options for your printer.

If the printer is `halted`, and there are outstanding `docend`, `sufend` or `halt` commands, these are transmitted/executed in that order.

The `pagestart` and `pageend` commands are similar to the `docstart` and `docend` commands, but operate upon pages. The diagram below is an exploded view of the middle job in our example to show the relationship of these commands.



As described previously, there are optional special copies of the `docstart` and `docend` strings appropriate for the banner pages. This may be appropriate where you want, for example, to specify landscape orientation but keep the banner constant.

5.8.1 Build-up of strings.

The two key points to remember about the commands are:

1. Each string is either sent to the printer or executed as a shell but not both.
2. Each string may be built up in several stages.

The command string in question is "selected" by specifying the command keyword, e.g. `setup`, `halt`, etc. This causes every string which follows to be considered to be part of, and appended to, the last such keyword read. The ASCII characters to be appended to the string then follow after the keyword. This allows the author of the setup file to group together parts of strings to be added according to the value of various suffixes. Here is an example:

```
setup 'Abc'
# We are now in a mode where all new strings are added to
# the "setup" string.
'def'
halt '01234'
# We are now in a mode where all new strings are added to
# the "halt" string.
pageend
# We are now in a mode where all new strings are added to
# the "pageend" string.
'xyz'
halt '5678'
```

This results in a `setup` string consisting of the characters *'Abcdef'*, a `halt` string consisting of *'012345678'* and a pageend string of *'xyz'*.

The command keywords may appear in any order and be repeated many times. There is always a "current" string, as selected by the last command keyword encountered.

Any character literals which subsequently appear are appended to the current string. The current string at the start of the file is defaulted to be the `setup` command.

5.8.2 The exec keyword.

The function of the `exec` keyword is to turn the current string into a shell command. In principle this could appear in some arbitrary and misleading part of the file, but it is much easier to understand if you insert it straight after the relevant keyword, as in

```
sufstart exec '/usr/fonts/download helvetica10'
```

The `exec` keyword causes the entire string to be treated as a shell command instead of a string to be sent. (This has the side effect of turning off the subsequent interpretation of escape sequences for that keyword). The shell command is invoked with standard output and standard error sent to the printer. The environment variables are set up and available for the shell command as for filter commands, described above.

The `SPOOLPAGE` environment variable is only set to anything useful with the `pagestart` and `pageend` keywords.

Any amount of white space may separate component strings of the command to be concatenated together, but only spaces or newlines explicitly specified by the user within the strings or definitions are inserted into the result.

Here is a simple example where `setup` and `halt` strings are specified:

```
setup '\e[\x42t^N\e(009,017,025,033,041,049,057,065,073'
halt exec 'echo finished using $SPOOLFORM paper|write fred'
```

Note:

1. The use of the `halt` command to do something not directly related to the printer.
2. The use of the `$SPOOLFORM` environment variable which is passed to the shell command.

This might seem easier to read in the following form with identical effect:

```
setup '\e[\x42t' '^N' '\e(009,017,025,033,041,049,057,065,073'
halt exec 'echo finished using $SPOOLFORM paper|write fred'
```

or definitions might be used thus:

```
LINES_66=\e[\x42t
EXPAND=^N
TABSET=\e(009,017,025,033,041,049,057,065,073
BOSS=fred
MESSAGE=Finished using $SPOOLFORM paper
setup LINES_66 EXPAND TABSET
halt exec 'echo ' MESSAGE '|write ' BOSS
```

5.9 Order of commands and keywords.

Commands and keywords can occur in any order; a command is terminated either by the end of the setup file, or by the next occurrence of a command keyword. Thus, it is possible to do something like:

```
setup "\e[\x42t" record 0:.totals extabs "^N"
```

which is equivalent to:

```
setup "\e[\x42t^N"
record 0:.totals
extabs
```

although, obviously, less readable! Basically, the rule is that when a string is seen, it is appended to the string whose keyword was last seen.

As another (possibly useful) side effect of this rule, it is possible to do slightly strange things with the `exec` keyword. A command string is either a string to be sent to the printer or a command to execute (*not* both). If it contains the `exec` keyword, it is a command to be executed, in which case escape sequences will not be expanded, otherwise they will be expanded. However, the `exec` keyword only affects strings appearing after it in the command. Thus, the 3 lines:

```
setup exec "echo \e[\x42t"
setup "echo " exec "\e[\x42t"
setup "echo \e[\x42t" exec
```

are all run as commands with their output sent to the printer, but in the third case, since the `exec` is after the "`\e[\x42`", GNUspool, rather than the shell and echo interprets it. This "trick" can be used to get unusual escape sequences into shell commands. Another way of doing the same thing is to use variable definitions; escape sequences are always recognised in variable definitions, whether used in an `exec` command or a string.

You can turn off the effect of an `exec` keyword for the currently active string if you really think you know what you are doing, by preceding it with a minus sign.

5.9.1 Refinements to handling of network etc commands

In recent releases of Xi-Text reflected in GNUspool, certain refinements have been made to the handling of shell commands previously introduced by `network=` etc, in particular:

`network=`	network output
`filter=`	post-processing filters
`align=`	alignment files
`execalign=`	alignment programs
`bannprog=`	banner program
`portsetup=`	initialise port prior to open
`stty=`	initialise after open

As before, the shell command to be executed may be specified using the standard sequences, for example:

```
network=xtelnet -H $SPOOLDEV
```

However they can also be specified using the same "construct string" technique used for the `setup` etc keywords, only `exec` is assumed to be turned on (i.e. escape sequences such as `\n` are not interpreted).

For example:

```
network
# Current string is now 'network'
'xtelnet '
'-H $SPOOLDEV'
```

Would have a similar effect, except in the case where the setup file was expecting to add subsequent strings to some other keyword, such as `docend` or `pagestart`.

Similarly all the other strings which were previously, and still may be, specified with *`keyword=`* such as `filter=` may also be specified in this way.

If the two styles are mixed, then the following actions are taken.

- The version with ***`keyword=`*** re-initialises the value. In principle, this could be used with an empty string to reset to the empty string.
- The version with ***`keyword=`*** does not affect which string is currently selected for the purpose of appending strings in the setup file. This is to preserve compatibility as far as possible with previous setup files.

As a (perverse) example, consider the following:

```
network 'command1'
'command2'
setup 'abc'
network=command3
'def'
network 'command4'
'command5'
```

This would leave the network command set to "`command3command4command5`" and the setup string set to "`abcdef`", as the "`network=`" command in between the `abc` and the `def` did not alter which string was being appended to, and also caused the preceding strings to be discarded.

5.9.2 Effect of exec with network etc

When network etc commands are specified using the "currently selected string" option rather than as `network=`, then `exec` may be specified as well. Unlike with `setup` etc, this does not mean that the string is to be run as a command, as this is already done.

The effect of the `exec` keyword in such instances is to cause the command to be directly executed, rather than via the shell. `PATH` and other environment variables are interpreted and expanded.

The reason for doing this is increased speed, as some shells take a long time to start up, and also in the event that closer control of signal handling is required, as some shells "catch" signals or program halts and then exit with an inappropriate exit code. Also an additional process is frequently created by having the shell running as well as the terminal server program or filter program.

5.10 Interaction of filter and setup strings.

This section clarifies how the `docstart` and `docend` strings interact with the `filter` command, if any. For the purpose of the following description suppose that there is no page delimiter, and the default of *form-feed* applies.

1. At the start of each document the string `docstart`, if any, is sent or executed.
2. The document is then sent to the printer, via the filter, if any.
3. At the end of the document a *form-feed* is sent if there is no `docend` string, no `filter` command *and* the last character actually sent was not a *form-feed*.
4. Finally the document end string `docend` is transmitted or executed.

The third rule above means that nothing extraneous to the document is sent to the printer if you opt for close control over the characters output. It also means that if you introduce a filter, and there is no `docend` keyword, you may wonder why the *form-feed* disappears. To overcome this, add the following to the setup file:

```
docend '\f'
```

Beware that some printers require a carriage return after the *form-feed*:

```
docend '\f\r'
```

If the page delimiter for the job is not *form-feed* then in step 3 above then the balance of the delimiter is output; for example if the delimiter consists of 66 *line-feeds* and the final page has 10 on it, 56 are output to complete the page. If the page is thought to have just ended then this is avoided.

5.10.1 Avoiding a terminating formfeed

As the default value of `docend` is a single formfeed, there are some cases where the extra formfeed is spurious and should be suppressed. This can be done by specifying the null string as a formfeed, thus:

```
docend ''
```

5.11 Conditions.

Conditions are the way of testing the value of the suffix, and selecting keywords, definitions, commands, and maybe further conditions, according to its value.

The syntax is as follows:

```
{  (suff1) items1
   (suff2) items2
    ...
}
```

If the suffix matches `suff1`, then `items1` are "obeyed", otherwise if it matches `suff2`, then `items2` are "obeyed", and so on until the list is exhausted, or until a match is found, whereupon subsequent matches are ignored.

For example a fragment of a setup file `a4` might be

```
DRAFT=\em0
CORRESP=\em1
LQ=\em2

sufstart
{
     (dr)      DRAFT
     (co)      CORRESP
                   record 0:.totals
     (inv)      CORRESP
            align=.invoices
            record 4:.totals
     (lq)      LQ
            nohdr
            single
            record 8:.totals
}
```

This would define modes to be selected if the form types were any of `a4.dr`, `a4.co`, `a4.inv` or `a4.lq` respectively. In this example the number of characters printed for each form type is counted at a different offset in the file .totals.

Note that:

1. This setup file uses (as do many) the fact that a command, such as `sufstart` accumulates all the strings until the next string-selecting keyword. Thus all the uses of definitions (of `DRAFT` etc) in the example are actually part of the `sufstart` command, however do not think of the `{}` as part of the sufstart command, this just selects the string being appended to.
2. Unknown suffixes are ignored; see below for a more sensible approach to unknown suffixes.

A simple "wildcard" notation, just like the Bourne shell's, may be used:

`*`	Matches any number of characters (including none at all).
`?`	Matches one character
`@`	matches the "null string", i.e. no suffix is specified
`[xc-p]`	matches `x` or `c` to `p` inclusive,
`[!s-z] or [^s-z]`	matches anything other than `s` to `z`.

It would probably be sensible to replace this example by:

```
DRAFT=\em0
CORRESP=\em1
LQ=\em2

sufstart
{
    (co)  CORRESP record 4:.totals
    (inv) CORRESP
          align=.invoices
          record 4:.totals

    (lq)  LQ
          nohdr
          single
          record 8:.totals
    (*)   DRAFT record 0:.totals
}
```

To treat all undefined suffix types as draft.

A portion of the pattern may be "remembered" by enclosing it in braces `{}`, and subsequently inserted into a string using the keyword `_` (underscore). For example, to select a point size on a laser printer based on the end of the suffix name the following scheme was adopted:

```
Default_point=10
{
     (?{[0-9][0-9]})      _
     (*)                 Default_point
}
```

This would extract the final two characters if they were both digits and insert them into the setup string, and otherwise insert 10.

5.12 Real setup files.

The following is a setup file used on a laser at Xi Software.

```
charge 6000
ixon nohdr
DOCSEP=\\$E*
NOJUST=\\$L0
DISPLAY=\\$Z.MESSAGE \\"$SPOOLHDR\\"
NEWL=\\\\n
ECHO=echo
FONTSHELL=/usr/llib/laser/bin/listfonts

docstart
        exec ECHO DOCSEP NEWL NOJUST NEWL DISPLAY
docend '$E*\n'
{
     (s*)
           filter=/usr/llib/laser/bin/side
           sufstart exec FONTSHELL 'start'
           sufend exec FONTSHELL ' end'
     (lp*)
           filter=/usr/llib/laser/bin/list
           sufstart exec FONTSHELL 'start'
           sufend exec FONTSHELL ' end'
     (*)
           filter=/usr/llib/laser/bin/finsert
}
```

Notes:

1. This provides for 3 types of suffixes:
    1. `scw10`, to denote landscape program listing, together with font type and point size,
    2. `lpcw12`, for portrait program listing in the same manner, and
    3. a default, that includes a filter to detect and download fonts as used by the document.
2. In the program listing mode, the shell script `/usr/llib/laser/bin/listfonts` either downloads (with argument `start`), or clears (with argument `end`) the required font(s). listfonts determines the appropriate font from the form type, which is passed in the `SPOOLFORM` environment variable.
3. The laser printer requires documents to begin and end with the string `$E*`. In the case of the `docstart` string, this is written `\\$E*`, because the shell requires a `\` to be inserted before the `$` if the `$` is not to be interpreted by the shell, and the syntax of the setup files requires two backslashes `\\` to be inserted to generate a single `\` as part of a string. Similar complications occur with the definition of `DISPLAY` used in the `docstart` string.
4. The elaborate `docstart` command puts up the job title on the front panel display of the laser printer.

The next example is that of a setup file for an Epson-compatible dot matrix printer.

```
NLQMODE=\ex1
DRAFTMODE=\ex0
ELITE=\eM
PICA=\eP
CONDENSED=\e\x0f
CANCCOND=\x12
PROP=\ep1
CANCPROP=\ep0
ixon extabs
width 136
{
     (*lq)
           sufstart NLQMODE
           sufend DRAFTMODE
             charge 2000
}
{
     (g*)
           nohdr
     (e*)
           sufstart ELITE
           sufend PICA
           width 163
           onlcr
           sufstart ELITE CONDENSED
```

```
            sufend PICA CANCCOND
            width 272
            onlcr
      (c*)
            sufstart CONDENSED
            sufend CANCCOND
            width 233
            onlcr
      (p*)
            sufstart PROP
            sufend CANCPROP
            width 200
            onlcr
      (nr*)
            sufstart PICA
      (*)
            onlcr
}
{
      (*uk*)
            sufstart '\eR\x03'
            sufend '\eR\x00'
}
docend '\f\r'
```

This provides for suffixes of the form `dr` and `lq` for pica draft and near-letter quality mode respectively, `edr` and `elq` for elite equivalents and `cdr`, `pdr` etc. for compressed mode etc. respectively. As a final touch, the language set can be adjusted using a `uk` in the middle, as in `cuklq`.

The `docend` string was added because the printer buffer didn't get flushed until it had seen a carriage return.

Note that form type suffixes beginning with `g`, such as `gdr` etc. (for graphics) has no `onlcr` in so that graphics output does not have a carriage return put in front of everything looking like a linefeed.

As a final example, here is a setup file we used for a Brother Laser Printer with LaserJet-compatible commands on which this document was originally printed.

```
# Keep a record of jobs printed in a file. This file will be
# interpreted relative to the current directory and will be
# located in the main printers directory,
# /usr/local/share/gnuspool/ptrs
# (Note that several printers may share the same record
# file if desired).

logfile=../Record

# Set the offline timeout to 3000 seconds

offline 3000

# We arrange to interpret CR characters as terminating
# a line. We give symbolic names to the commands to
# enable and disable this.

LINE_TERM=\e&k3G
OFF_LINE_TERM=\e&k0G

# Give symbolic names to the commands to select top
# and bottom input feed. Top feed is for single sheets
# and requires the operator to insert the page by hand.

TOPFEED=\e&l2H
BOTTOMFEED=\e&l1H

# Give symbolic names to the commands to select portrait
# and landscape orientations

PORTRAIT=\e&l0O
LANDSCAPE=\e&l1O

# Give symbolic name to the reset command.

RESET=\eE

# Give symbolic names to the built-in fonts

ANELIA=\e(s1p129T
BROUGHAM=\e(s0p128T
BRT=\e(s1p153T
HV=\e(s1p152T

# Give symbolic names to the pitches available

P8=\e(s8.3H
P12=\e(s12H
P16=\e(s16.6H
P10=\e(s10H

# Normally insert carriage returns before every line
# feed.

onlcr

# Now follows the first test of a suffix.
# We have a special case of the suffix "lh"
# for letterhead which we use a wordprocessor for
# l<anything-other-than-h> for landscape
# p<anything> for portrait
# anything else means more or less "straight through"

{
# First case is our letterhead.
# Note that for this the suffix must be "lh"
# NOT "lh10" or anything like that

      (lh)

# If we have a header page, we do want to interpret
# carriage returns whilst we are printing it
# On our printer this combination (before the main
# document) causes the header to come out on the bottom
# try so that the letterhead pages are not used for
# the headers. You might need to include

BOTTOMFEED

# if your printer behaves differently

            banner
            docstart LINE_TERM
            docend OFF_LINE_TERM '\f'
                       -banner

# We seem to have to reset feed from the top tray at the start
# of each document. Make sure we reset to bottom feed when we've
# finished

            docstart TOPFEED
            docend '\r'
            sufend BOTTOMFEED

# Next we provide for landscape modes.
# The [!h] means "anything other than h".
# In fact this is not necessary as we have already
# dealt with the "lh" case.
# The "*" is essential to make it accept any number
# (including none) of trailing characters

      (l[!h]*)
            sufstart LINE_TERM

# A refinement to the handling of landscape -
# make the header pages come out in portrait
# (this is the default, and reset at the end
# of the document with the docend command below)

            banner
            -banner
            docstart LANDSCAPE
            docend PORTRAIT
            sufend OFF_LINE_TERM
# Portrait mode - rely on this being the default
# setting, and upon landscape mode being reset
# Note that we must have the "*" as we may want
# to interpret further characters in the suffix
```

```
      (p*)
            sufstart LINE_TERM
            sufend OFF_LINE_TERM

# Any other combination and none, handle banner
# pages as text and treat as sent by the software
# we use just "a4" for wordprocessor output (apart
# from to a4.lh) and other software
# NB for graphics software we might want to turn
# "onlcr" off as well.

      (*)
            banner
            docstart LINE_TERM
            docend OFF_LINE_TERM '\f'
            -banner
            docend '\r'
}

# Next we provide for font selection.
# We do this by looking for strings somewhere in the
# suffix, thus ltm10 will match the second option
# below and select the "BRT" font - in landscape mode

{
      (*an*)
            sufstart ANELIA
      (*tm*)
            sufstart BRT
      (*hv*)
            sufstart HV
      (*)
            sufstart BROUGHAM
}

# Finally we take any trailing number on the end of the
# suffix to be a pitch. If we cannot find a pitch setting,
# or we don't know it, use pitch 10

{
      (*12)
            sufstart P12
      (*8)
            sufstart P8
      (*16)
            sufstart P16
      (p*)
            sufstart P10
      (l[!h]*)
            sufstart P10
}
```

```
# Undo everything we did when the printer is halted.
halt  RESET
```

With a PJL printer, it might be helpful to provide for an abort string, to reset the printer in the event that it was aborted in the middle of a graphics sequence, thus:

```
abort '\e%-12345X@PJL EOJ'
```

6 Internal Programs and files

The internal programs are listed below, arranged by function. With a few exceptions these programs are not intended be invoked or accessed by a user, or administrator.

The internal programs are all held in the same directory, `/usr/local/libexec` by default. A different directory may be selected by assigning to the environment variable `SPROGDIR`, usually in the *Master Configuration File* `/usr/local/etc/gnuspool.conf`.

6.1 Core Internal Programs

6.1.1 Spshed

Spshed is the spool scheduler process for the **GNUspool** spooler and document management system.

To start **GNUspool**, spshed should be started. This should normally be done using gspl-start, which restarts it and passes the appropriate options. This may be put within the Operating System startup scripts.

Likewise to halt it, gspl-stop should be invoked. This should be put, using the `-y` option, in any system shutdown routines.

If you do need to kill spshed for any reason, first try kill without any -9 option. This will give spshed the opportunity to attempt to tidy up any IPC facilities before shutting down.

Information, either in respect of other machines to connect to, or pre-existing jobs and printers on the current machine, are read from the files `/usr/local/etc/gnuspool.hosts` and the directory `/usr/local/var/gnuspool` respectively.

If a networked version of **GNUspool** is being run, then a "slave" spshed process is spawned to monitor and process incoming network messages. Incoming remotely-submitted jobs and API interfaces are handled via a separate process xtnetserv, which is also invoked as appropriate by gspl-start.

Appropriate log messages are written by spshed and other system processes to a log file, `/usr/local/var/gnuspool/spshed_reps`. Be sure to check this file for any error messages relevant to any problems you encounter.

6.1.1.1 Files used by spshed (under /usr/local)

`etc/gnuspool.hosts`	host names and descriptions
`etc/gnuspool.conf`	master configuration file
`share/gnuspool/help/int-config`	message file
`var/gnuspool`	spool directory
`var/gnuspool/spshed_jfile`	job file
`var/gnuspool/spshed_pfile`	printer file
`var/gnuspool/spshed_reps`	error log file

`var/gnuspool/spufile0`	user data
`var/gnuspool/charges0`	user charges data
`var/gnuspool/spmm_jobi`	job memory-mapped hash file
`var/gnuspool/spmm_jobd`	job memory-mapped data file
`var/gnuspool/spmm_ptrs`	printers memory-mapped file
`var/gnuspool/spmm_xfer`	communication buffer memory-mapped file

6.1.1.2 Environment variables relevant to spshed

`SPOOLDIR`	alternative location for spool directory, by default `var/spool/gnuspool`.
`SPHELPDIR`	alternative location for help file directory, by default `share/gnuspool/help`.
`SPROGDIR`	alternative location for internal programs directory, by default `libexec/gnuspool`.

6.1.1.3 IPC facilities used by spshed

An IPC message queue, with key `0x58691000` and owned by `gnuspool`, is created by spshed and used to receive messages from user processes, pass instructions to spd, and to pass internal messages from the slave spshed process to the master.

Two shared memory segments are created to hold details of jobs and printers. As the shared memory facility provides no facilities for growth, then additional shared memory segments may be created if the job and printer lists expand sufficiently and the original ones deallocated.

The keys given to the shared memory segments start at `0x58691002` and ascend upwards to `0x58691064` before wrapping around.

A further shared memory segment, with key `0x58692002` is created to hold details of pending jobs before transfer to the main shared memory segment.

Some versions of GNUspool may be compiled to use memory-mapped files rather than shared memory. The files are held in the spool directory, by default `/usr/local/var/gnuspool`, and have the names `spmm_jobi`, `spmm_jobd`, `spmm_ptrs` and `spmm_xfer`.

A set of 5 semaphores, with the key `0x58691001` is created to interlock access to the shared memory segments. Some versions of GNUspool may be compiled to use file locking and use `spjob.lock`, `spptr.lock` and `spxfer.lock`.

The presence or absence of these IPC facilities is used by spshed and other programs to determine whether a previous copy of itself is running. If spshed is abnormally terminated, it will probably be necessary to delete these IPC facilities before spshed can be restarted.

When printers are set running, spshed invokes an instance of spd to control each printer. Mail and attention messages are passed to spmdisp for processing.

6.1.1.4 Internet Ports used by spshed

spshed accepts and sends interconnections from other machines on TCP port, passes the contents of spool files on a further TCP port, and undertakes "probes" on a UDP port.

The port numbers are set up in the `/etc/services` file when **GNUspool** is first installed.

6.1.1.5 Diagnostics from spshed

spshed is invoked from system startup procedures or other programs such as gspl-start. Thereafter it runs as a "daemon process" and diagnostics are not written to any terminal but to the file `/usr/local/var/gnuspool/spshed_reps`.

In the event of any problems this file should be examined.

6.1.2 Xtnetserv

Xtnetserv is the remote server process for the **GNUspool** spooler and document management system. It serves 3 purposes; firstly it accepts jobs from other hosts submitted by gspl-rpr, secondly it accepts jobs and administration requests from DOS and Windows machines, and finally it supports API operations.

It is normally invoked by gspl-start.

Information, in respect of other machines to connect to, is read from the file `/usr/local/etc/gnuspool.hostss`.

6.1.2.1 Files used by xtnetserv under /usr/local

`etc/gnuspool.hosts`	host/client names and descriptions
`etc/gnuspool.conf`	master configuration file
`share/gnuspool/int-config`	message file
`var/gnuspool`	spool directory
`var/gnuspool/spshed_reps`	error log file
`var/gnuspool/spufile`	user data

6.1.2.2 Environment variables relevant to xtnetserv

`SPOOLDIR`	alternative location for spool directory
`SPHELPDIR`	help files directory

6.1.2.3 Internet ports used by xtnetserv

xtnetserv uses 2 ports:

```
gnuspool-netsrv
```

(N.B. no second "e") to accept incoming jobs on TCP from gspl-rpr, and on UDP from the DOS/Windows interface, and 2 further ports

```
gnuspool-api
```

to communicate with the API library.

The port numbers are set up in the `/etc/services` file when **GNUspool** is first installed.

6.1.2.4 Diagnostics from xtnetserv

xtnetserv runs as a "daemon process" and diagnostics, apart from those detected when it is first started, are not usually written to any terminal but to the file `/usr/local/var/gnuspool/spshed_reps`.

In the event of any problems this file should be examined.

6.1.3 Spd

Spd is the spool despooler process for the **GNUspool** spooler and document management system.

It is only ever intended to be invoked by the scheduler process spshed, and the effect of a user invoking this process is not defined.

The arguments are the printer name and the slot number in shared memory which refers to it.

6.1.3.1 Files used by spd under /usr/local

`etc/gnuspool.conf`	master configuration file
`share/gnuspool/help/int-config`	message file
`var/gnuspool`	spool directory
`share/gnuspool/ptrs`	printer setup file directory
`var/gnuspool/spshed_reps`	error log file

6.1.3.2 Environment variables relevant to spd

`SPOOLDIR`	alternative location for spool directory
`SPROGDIR`	alternative location for spool programs
`SPOOLPT`	alternative location for printer descriptions

6.1.3.3 Diagnostics from spd

Spd is usually invoked indirectly, via spshed from system startup procedures or other programs such as gspl-start.

Thereafter it runs as a "daemon process" and diagnostics are not written to any terminal but to the file `/usr/local/var/gnuspool/spshed_reps`.

In the event of any problems this file should be examined.

6.1.4 Spdinit

Spdinit is an internal program used by spd to decode printer setup files for the **GNUspool** spooler and document management system.

It is only ever intended to be invoked by the despooler process spd, and the interface is not in any way guaranteed in future releases.

Output is to standard output, and it may be useful to run strings on the result to see whether setup files have produced the expected result.

6.1.4.1 Files used by spdinit under /usr/local

`etc/gnuspool.conf`	master configuration file
`share/gnuspool/help/int-config`	message file
`var/gnuspool`	spool directory
`share/gnuspool/ptrs`	printer setup file directory
`var/gnuspool/spshed_reps`	error log file

6.1.4.2 Environment variables relevant to spdinit

`SPOOLDIR`	alternative location for spool directory
`SPROGDIR`	alternative location for spool programs
`SPOOLPT`	alternative location for printer descriptions

6.1.4.3 Diagnostics from spdinit

Spdinit is usually invoked indirectly, via spd and spshed from system startup procedures or other programs such as gspl-start.

Diagnostics are not written to any terminal but to the file `/usr/local/var/gnuspool/spshed_reps`.

In the event of any problems (typically syntax errors) this file should be examined.

6.1.5 Spjobdump

Spjobdump is invoked by gspl-pq, gspl-xmpq and gspl-qdel to unqueue jobs when required.

It is not intended for general use and is not documented further.

6.2 Message Handlers

6.2.1 Spmdisp

Spmdisp generates messages as required by spshed in response to the mail or write completion or attention options of gspl-pr and equivalents.

By default, it uses the system basic mailer to dispose of mail options, spwrite to send messages to users' terminals and dosspwrite to send messages to Windows PCs.

The messages are generated by spmdisp from the system message file, which by default is `/usr/local/share/gnuspool/help/int-config`.

The program to be used in each case may be overridden by assignments to the environment variables `MAILER`, `WRITER` and `DOSWRITER`, most conveniently in the master configuration file `/etc/gnuspool.conf`. The program (or shell script) to be run in each case should take data on standard input and the relevant user name as the first argument, and will run under the identity `gnuspool`.

These variables may also be set on a per-user basis by assignment in a `.gnuspool` file located in a user's home directory. The user may also specify an alternative message file by assignment to the variable `SYSMESG`. These programs or scripts will be run under the identity of the user, typically the owner of the job to be run.

The interface (options etc) are internal to **GNUspool** and are not documented here.

Note that spmdisp is identical to **Xi-Batch**'s btmdisp, apart from using different message and configuration files.

6.2.1.1 Files used by spmdisp

`/usr/local/share/gnuspool/help/int-config`	message file
`~/.gnuspool`	user's home directory

6.2.1.2 Environment variables relevant to spmdisp

`DOSWRITER`	alternative write to PC program or script.
`MAILER`	alternative mail results program or script.
`WRITER`	alternative write results program or script.
`SPHELPDIR`	alternative location for help file directory, by default `/usr/local/share/gnuspool/help`.
`SYSMESG`	alternative message file location (relative to user's home directory if need be).

6.2.2 Spwrite

Spwrite sends messages to users' terminals in response to the `-w` and `-A` options of gspl-pr. It is used in preference to the write command as this picks just one (and usually the wrong one!) of the terminals at which the user may be logged in, and does not display a suitable name for the originator of messages.

Spwrite takes a list of one or more users as arguments. It sends the text on standard input to each user's terminal. The message is mailed to users who cannot be reached. This facility is available for use in your own software if you wish.

6.2.2.1 Files used by spwrite under /usr/local

`share/gnuspool/help/int-config`	message file

6.2.2.2 Environment variables relevant to spwrite

`SPHELPDIR`	alternative location for help file directory, by default `/usr/local/share/gnuspool/help`.

6.2.3 Dosspwrite

Dosspwrite sends messages to Windows PCs similar to spwrite does for user's terminals in response to the equivalents of the `-w` and `-A` options of gspl-pr. This is only done for jobs which originated on Windows PCs.

The Windows PC must be running sprserv for this to be effectual.

If the job was submitted by a user working from a client with a DHCP-allocated IP address, a message may be received on all clients currently logged-in with that user name.

6.2.3.1 Files used by dosspwrite under /usr/local

`share/gnuspool/help/int-config`	message file

6.2.3.2 Environment variables relevant to dosspwrite

`SPHELPDIR`	alternative location for help file directory, by default `/usr/local/share/gnuspool/help`.

6.2.4 Spexec

Spexec runs commands for macros under the identity of the invoking user. This is required because gspl-pq and gspl-xmpq are set-user programs (to other than `root`) and there is an inherent security breach in many versions of Unix in that such programs cannot divest themselves of traces of the set-user user id.

This program is only intended for internal use and is not further documented.

6.3 File Formats

6.3.1 Gnuspool.conf

`/usr/local/etc/gnuspool.conf` is an optional file for providing additional environment variables to component programs of the GNUspool spooler and document management system.

This may be useful for relocating standard files and directories, such as `SPOOLDIR`, which defaults to `/usr/local/var/gnuspool`, so that a different spool directory is used.

However completely arbitrary environment variable assignments may be made, in particular so as to pass the resulting environment on to subprocesses invoked by the despooler program spd.

The format of the file consists of environment variable assignments such as:

```
SPOOLDIR=/usr2/spooldir
```

Comment lines may be included, introduced by a `#` sign and blank lines are ignored.

Every component program of **GNUspool** examines this file and resets its environment from this file as the first step of execution.

6.3.1.1 Important notes for gnuspool.conf

1. Make sure that `gnuspool.conf` is world-readable!
2. Ensure that no part of **GNUspool** is running before altering this file.

6.3.2 gnupool.hosts

`/usr/local/etc/gnuspool.hostss` is used to inform the GNUspool spooler and document management system, and in particular spshed and xtnetserv, which other host machines are to be attached.

The host machines should in general be provided for in the standard file `/etc/hosts`.

The file consists of comment lines introduced by the `#` character, and of lines consisting of up to 4 fields, of which only the first is mandatory. These fields are as follows:

6.3.2.1 Host name

This is the name of the host as given in the `/etc/hosts` file or obtainable from DNS.

Alternatively an internet address of the form `193.112.238.10` may be given if necessary and an alias is provided on the next field, but this is not recommended.

6.3.2.2 Alias name

This is the name of an alias to be used in preference to the host name to refer to the machine. To be particularly beneficial, this should be shorter than the host name.

If this field is not required, but subsequent fields are required, then the alias name may be replaced by a single `-` sign.

6.3.2.3 Flags

This is a comma-separated list of markers to denote information about the connection. The following markers may be specified.

`probe`	Indicates that a datagram should be sent, and a reply awaited, from the host, before a full-blown connection is attempted. This is recommended wherever possible, or it is not sure in which order machines are booted.
`manual`	indicates that no connection at all is attempted. To connect to the machine in question, then gspl-conn should be invoked.
`trusted`	indicates that the host is "trusted" by the current machine, which transmits information about Windows clients and their password validations, so the other host need not make such enquiries.

`client(username)`	indicates that no connection is attempted; the current machine is acting as a server for Windows clients. The specified username is to be considered as the owner of any jobs submitted, and the user to whom charges should be applied and to which privileges apply; see gspl-user. If `(username)` is omitted, then the Windows user is assumed, which should correspond to a user name on the host system.
`Clientuser(name)`	Indicates that the whole entry identifies a "roaming user" who might be using one of several Windows clients, possibly with IP addresses assigned via DHCP. The host name in this case is replaced by the Windows user name, and the alias gives the Unix user name if different. If `(name)` is specified, then a password is demanded at the Windows client if the client's IP address does not match that of machine.
`dos(username)`	Is a synonym for `client(username)` kept for historical reasons.
`dosuser(name)`	Is a synonym for `clientuser(name)` kept for historical reasons.
`external`	Is a synonym for client (no username) kept for future extensions.
`pwchk`	Always demand the user's Unix password when first starting up.

6.3.2.4 Timeout

This gives a timeout value in seconds after which the interface is to be considered closed following a connection or alternatively to await a connection after a probe request.

A default of 1000 seconds applies if none is specified.

In the case of Windows clients, the "login" is considered to be dropped after this time, and the user may be asked for a password again.

6.3.2.5 Local Address

On some machines, the "local" host address may be different from that obtained by looking at the result of gethostname(3). To specify a different address for "this" machine, a line of the form:

```
localaddress 193.112.238.112
```

may be specified, but this must precede all other host names in the file.

7 User Programs

Users have a wide variety of Unix programs which may be used to submit and manage print jobs to the spooler. This includes a set of standard command line and interactive programs, plus optional Motif GUI applications and emulators for Unix lp commands.

The following are the user programs available, listed by function, including some intended only for set-up and installation. Some of the descriptions which follow are merged together to save repetition.

More detailed descriptions of the interactive interfaces to gspl-pq, gspl-user, gspl-xpq, gspl-xmpq, gspl-xuser and gspl-xmuser follow in the next two chapters, the descriptions here concentrating on the command line options to these programs.

7.1 Option Syntax

All of the options referred to in the descriptions of the shell-level programs for **GNUspool** below may be supplied in a *configuration file* (q.v.), or in an environment variable whose name is the same as the calling program, except that it is in upper case and non-letters converted to "_" underscore.

This may enable defaults to be supplied according to the application from which the program is invoked. However any options and arguments supplied on the command line usually take priority.

Additionally by editing the appropriate message file (q.v.) it is possible to change the option letters and keywords from those described.

In nearly all cases there are two alternative ways of supplying options:

- Via a traditional Unix-style -letter option, for example as `-z`.
- Via a keyword-style option, such as `+zero-charge`. Keywords are case-insensitive.

The syntax of options is intended to be as flexible as possible. Options which do not take arguments may be grouped together as in

```
-wmv
```

or they may be given separately as in

```
-w -m -v
```

White space is optional in the case of options which do take arguments, thus both

```
-p150
```

and

```
-p 150
```

are acceptable and have the same effect.

If the keyword version of an option is given, then it must be separated from its argument by white space thus

```
+priority 150
```

7.1.1 Configuration files

To save the user from having to specify commonly-used combinations of options, there are mechanisms enabling these to be supplied to each program automatically.

One mechanism is the use of a *configuration file*, `.gnuspool`, in the current or user's home directory. The other is the use of an environment variable.

These files may also be used to specify alternative *message files*.

The format of configuration files is akin to a set of environment variable assignments, with empty lines and lines beginning with `#` being ignored.

The name assigned to is the same as that of the calling program but in upper case and non-alpha converted to underscore, for example that corresponding to gspl-pq is GSPL_`PQ` etc. This is the same as for the corresponding environment variable.

Usually options are taken from the following places in order, so that later-processed ones override earlier ones:

Standard defaults	Each program has a set of standard defaults which are used to initialise the parameters when the program is invoked.
User profile	In some cases, for example default priority, the user's profile as displayed by gspl-user is used to initialise the defaults.
Home directory	The file `~/.gnuspool` is read, and any options specified therein (i.e. with an assignment to the appropriate name) are interpreted.
Environment	Any options specified in the appropriate environment variable (you will almost certainly have to use quotes when setting it via the shell in order to preserve the white space) are read and interpreted.
Current Directory	The file `.gnuspool` is read, and any options specified therein (i.e. with an assignment to the appropriate name) are interpreted. Note that this may mean that the `.gnuspool` file is read twice if the command is run from the current directory. Beware therefore of options which are cumulative, such as arguments and redirections, and use the "cancel existing" options.
Command line	Any options specified on the command line are interpreted last.

Most options have inverses so that it is possible to reset anything which may have been set by previously-read options. Extra care should be taken with cumulative options, so that these are not doubled, especially in the case where the home directory is also the current directory.

7.1.2 Option path

The above description of the order of selection of configuration files, environment and command-line options is the default.

It may be desirable to change the order of selection of options, in to eliminate some alternative locations or to include others.

The environment variable `GS_CONFIGPATH` may be set to a colon-separated list of directories (environment variables and *~user* constructs are appropriately interpreted).

The symbol `!` is used to represent the relevant environment variable, and `-` is used to represent option arguments.

The default value of `GS_CONFIGPATH` is

```
~:!:.:-
```

This provides the interpretation of options in various configuration files and the environment which is documented above.

Note that it is possible to eliminate or override the interpretation of options on the command line by removing or relocating the `-`. This may have very surprising effects especially where configuration files wipe out the effects of options which may have been set on the command line. Where the interpretation of options has been removed altogether, then any options supplied will probably be objected to or misinterpreted as file names or similar.

The commands

```
+freeze-home
```

and

```
+freeze-current
```

and equivalents do not take into account the value of `GS_CONFIGPATH` in any way.

Finally please note that any non-existent or inaccessible directories and files will (usually) be silently ignored. If a configuration file appears to exist but is inaccessible, a diagnostic may be given; however in some cases this may be misleading due to the fact that various versions of Unix are misleading or inconsistent with regard to the error codes reported from an attempt to open a non-existent or inaccessible file in a non-existent or inaccessible directory.

7.1.3 Message files

As well as providing help and error messages, screen key assignments etc, message files also provide the option letters and keyword names used to specify the options.

For each command, there is a default message file. For most of the shell-based commands, this is `/usr/local/share/gnuspool/help/rest.help`. Alternative message files may be specified using an environment variable or configuration file assigning values to a name. For most of the shell-based commands, this is `SPRESTCONF`.

Within the message file itself, the option letters and keywords are set up using sequences of the form

```
A300:?,explain
```

Comments and the context should make clear which commands these options relate to.

These sequences define

A state number	The state number, in the above example 300, which is used internally to denote the argument.
option letters	A single character, often a letter, but in the above example ?, which is the single-character variant of the option, thus -?. Several option letters, each separated by commas may be defined. To define "`,`" itself as an option "letter", use `\,`.
option keywords	A string of alphanumerics, possibly including hyphens and underscores, is used to denote an option keyword, in the above example `+explain`. Several such keywords may be defined, each separated by commas. Note that the case of letters in the keywords is discarded.

7.1.4 Location of message files

It is possible to specify alternative locations for message files so that alternatives are selected according to the application being run etc.

The location may be specified using configuration files in a similar fashion to the search for options, except that the search runs the other way.

The search is in the following order:

Current Directory	If a configuration file in the current directory specifies a location for the message file, by means of an assignment to the relevant variable (for most shell-based commands this is `SPRESTCONF`), then this file is taken. Environment variables in the form `$ABC` and users' home directories in the form `~user` are appropriately expanded. The sequence `$0` is replaced by the name of the program being invoked. (This process may run recursively up to a level of 10)[1].
Environment	If the relevant environment variable (for most shell-based commands this is `SPRESTCONF`) specifies a location, then this is taken.
Home Directory	A configuration file in the home directory may specify a location for the message file.
Default Location	If none of the above specify a replacement message file then the default location is taken.

If a file is specified but does not exist, then this is a fatal error.

However there is an additional step to assist the user to set up some alternative files with a default name.

Should the file not exist, then the search falls back to a name generated by taking the last part of the default file name (for example `rest.conf`) and substituting this for the last part of the file name specified.

For example if the normal message file for the command were

```
/usr/local/share/gnuspool/help/rest.help
```

and the user had specified in a configuration file

```
SPRESTCONF=~/$0.help
```

---

[1] I.e. If the environment variable expansion contains another $name construct, this will be expanded again, but only to 10 levels deep.

then if he or she were to run, say, gspl-pr, then the file

```
~/gspl-pr.help
```

would be searched for. If this did not exist, then a search would be made for

```
~/rest.help
```

This is particularly important for screen-based programs, where a version of the message file suitable for the user's current terminal can be specified, and a default "dumb terminal" version fallen back to if this is unavailable.

7.1.5 Path to locate message files

The above search path may be modified using the environment variable `GS_HELPPATH`. The interpretation is very similar to the description above for `GS_CONFIGPATH`, except that `-` fields are ignored.

The default value of `GS_HELPPATH` is `.:!:~` giving the interpretation described above. Note that this is in the opposite order to `GS_CONFIGPATH`.

7.2 Submitting print jobs

7.2.1 Gspl-pr

```
gspl-pr [ -options ] [ file ... ]
```

gspl-pr queues up one or more print jobs for the **GNUspool** spooler.

If one or more files are specified, each file is treated as a separate spool job, with a title constructed from the file name. If no files are specified gspl-pr reads from the standard input instead. In this case the job is placed on the queue with a blank title unless otherwise specified.

The standard default options are to print one copy, preceded by a banner page, with the priority, printer and form types specified for the user. The standard banner page will contain: the user name and job title (in large characters), the name of the user who submitted the job (if different), the job number, its size and priority.

Other default options may be set up using the environment variable `GSPL_PR` and/or `.gnuspool` files. Any defaults may be overridden by re-specifying that option on the command line.

7.2.1.1 OPTIONS

The environment variable on which options are supplied is `SPOOLOPTS` and the environment variable to specify the help file is `SPRESTCONF`. (Spr is a special case for historical reasons).

Option		Arg	Description
`-?`	`+explain`		causes a summary of the other options to be displayed without taking further action.

Option		Arg	Description
-a	+mail-attention		requests that a message be sent in the mail to the user if a job is awaiting attention, either because it reaches the top of the queue and the required form type is currently unavailable or because the form requires alignment.
-A	+write-attention		requests that a message be sent to every terminal at which the user is logged in, or by mail otherwise if the user is not logged in, if a job is awaiting attention, either because it reaches the top of the queue and the required form type is currently unavailable or because the form requires alignment.
-b	+noatt-message		turns off any job attention messages previously requested.
-c	+copies	*n*	sets the number of copies to be printed to n, which may possibly be zero. The maximum number of copies at a time may be restricted to a number for each user such as 10. See the output from gspl-user for information on the maximum number of copies.
-C	+classcode	*nnnn*	where nnnn is some collection of the letters A through to P inclusive (upper or lower case) with - to indicate ranges, requests that the job be given the specified class code. The class code of a job may restrict which printers the job may be printed on, or which other users can display details of the job. In normal circumstances (except where the user has override class privilege), the specified class code is reduced to the intersection of (i.e. is "anded" with) the specified class code and the user's own class code.
-d	+delimiter-number	*n*	is used in conjunction with the -Dstring option to specify that a page is to be considered complete after reading n delimiter strings. The default value is 1.
-D	+delimiter	*string*	specifies an alternative delimiter to the default of formfeed as being the end of a page. If a number is given using the -dn option, then the page is considered ended after that number of occurrences of the delimiter. The string given may contain standard constructs such as \n \r \tetc to denote linefeed, carriage return etc. Remember that when the job is printed, the printer's setup file may re-specify the delimiter, which will replace the delimiter specified by these options.

Option		Arg	Description
`-E`	`+external-system`	*n*	Used internally to signify originating system type. n may be a number, or the name of a system type given in `/etc/gnuspool.ext`. This option may only be used by `root` or `gnuspool` regardless of other privileges set, as it is intended for use by xtlpd to set parameters on incoming jobs. Any errors in this parameter, including permissions, are silently ignored.
`-f`	`+formtype`	*string*	specifies that the supplied form type (with optional suffix introduced by "`.`" or "`-`") is used. Note that not all users may be allowed to select form types other some pattern, causing the job to be rejected.
`-F`	`+post-proc-flags`	*string*	causes the specified string to be passed as the value of the environment variable `SPOOLFLAGS` to any program invoked at the time the job is printed. This might be as a post-processing filter command, or alternatively some shell command invoked as (for example) a `docstart` string.
`-h`	`+header`	*string*	supplies string as the title for each job, otherwise it is derived from the file name specified. The title appears on the banner page, if this is printed.
`-i`	`+interpolate`		causes any file name suffixes .c, .o etc of the files from which jobs are created, to be appended to the form type, unless it already has a suffix, for example: `gspl-pr -i -f a4 x.c y.ps` would submit jobs using the files `x.c` and `y.ps` with form types `a4.c` and `a4.ps` respectively.
`-I`	`+no-interpolate`		cancels any previously-specified `-i` option.
`-j`	`+job-wait-time`	*n*	is relevant only if input to gspl-pr or gspl-rpr is from a pipeline as in `program|gspl-pr ....` This causes a job to be submitted if no data is received over the pipe for n seconds. gspl-pr will still wait, and may submit any number of subsequent jobs until the pipe is closed by the program generating the output. Note that nothing can be done by gspl-pr to guess at what output might appear (possibly halfway through a line) if the program writing to the pipe does not flush its output buffer after each occasion that it writes something. Specify a parameter of zero to cancel any previous `-j` option, if required.
`-l`	`+local-only`		in a networked environment requests that the job be printed only on the printers local to the machine at which the job is submitted.

Option		Arg	Description
`-L`	`+network-wide`		cancels the request that the job or jobs be printed only on printers local to the machine at which the job is submitted.
`-m`	`+mail-message`		requests a message to be sent in the mail to the user on completion or termination of the job.
`-n`	`+delay-for`	*nn*	specifies that the job should be held on the queue for at least nn minutes from the current time before being printed. The time may alternatively be specified as *hh:mm* or as *hh:mm:ss*, specifying a delay in hours and minutes, or hours, minutes and seconds.
`-N`	`+delay-until`	*time*	is an alternative to the `-n` option to specify the earliest time at which the job is to be printed. The argument may be *hh:mm* or *hh:mm:ss* to give the time of day in 24-hour clock notation. If the time has passed, then tomorrow is assumed. Alternatively a date and a comma may be prefixed to the time in the form *mm/dd* or *dd/mm* depending upon the local convention for date format used. Thus the resulting argument might be `10/11,12:30`
`-o`	`+originating-host`	*host*	Used internally to signify originating host name, in place of the local host. The host name given should appear in the host file `/etc/gnuspool.hostss`, possibly with the external keyword. This option may only be specified by `gnuspool` or `root` users, but is silently ignored in other cases as are all other errors. It is intended for use by xtlpd to set parameters on incoming jobs.
`-O`	`+odd-even-flags`	*flag*	where flag is one of `O`, `E`, `A`, `B` or `-` (the letters may be upper or lower case), cause odd or even-numbered pages to be skipped. or `-` to reset this flag. `O` causes odd-numbered pages not to be printed. `E` causes even-numbered pages not to be printed. `A` and `B` are useful if more than one copy is to be printed. `A` causes even-numbered pages not to be printed on odd-numbered copies, and odd-numbered pages not to be printed on even-numbered copies. `B` is the other way around. If you do not understand this, all you have to do is remember that `gspl-pr -c2 -OA ....` prints all the odd-numbered pages followed by all the even-numbered ones. For this to work properly, page delimiters must be set appropriately.

Option		Arg	Description
`-p`	`+priority`	*n*	specifies the priority of the job, between 1 (lowest) and 255 (highest) or some narrower range to which the user is limited. Increasing the priority of a job increases its chances of being printed earlier than it otherwise would be whilst increasing the charge applied to the user in a non-linear fashion.
`-P`	`+printer`	*name*	specifies that the job is to be sent to a printer with the name given, as opposed to printing it on the first available printer with the given form type. name may be a pattern to select any printer matching the pattern. A user may be limited to a range of printers which must be a superset of name. To "turn off" a printer name previously specified by a preceding `-Pname` option, put a single `-` sign as the printer name.
`-q`	`+retain`		requests that the job or jobs be retained on the queue with copies set to zero after printing, for explicit deletion, or automatically at the expiry of the timeout (as set by the -tn option).
`-Q`	`+host`	*host*	send the job or jobs to the given hostname. Supplied to gspl-pr it will re-invoke gspl-rpr with the same command-line options.
`-r`	`+banner`		restores banner pages previously suppressed using `-s`. Note that some form types may be set up never to use banners regardless of this option.
`-R`	`+page-range`	*m−n*	Specifies that pages `m` through to `n` inclusive are to be printed. This does of course assume that the job has recognisable pages. If `m` or `n` are omitted, then "the beginning" or "the end" respectively is assumed, so `-R 3-5` would print pages 3 to 5, `-7` would print pages 1 to 7 inclusive, and `-R 4-` would print page 4 to the end inclusive. `-R 1-` would turn this option off by selecting 1 to the end.
`-s`	`+no-banner`		suppresses any banner page (large letter user name etc) which is printed before the job itself. Note that some form types may be set up always to print banners regardless of this option.
`-t`	`+printed-timeout`	*n*	
`-T`	`+not-printed-timeout`	*n*	specifies that if held on the queue without being printed, the job will be deleted automatically after n hours. The default value is 168 hours (1 week), and the maximum value is 32767 hours (nearly 4 years).

Option		Arg	Description
`-u`	`+post-user`	*name*	requests that the specified user name be substituted for the submitting user on the banner page optionally printed at the start of the job. The job still remains the responsibility of the submitting user. To "turn off" a user name specified in a previous `-u` option, put a single `-` sign as the user name. The user name must exist on the machine to which the job is queued, but this will be ignored if not.
`-U`	`+originating-user`	*user*	Used internally to signify originating user name. This may only be specified by users `gnuspool` and `root` to have any effect, otherwise the option is silently ignored, as are all errors. It is intended for use by xtlpd to set parameters on incoming jobs.
`-V`	`+toggle-verbose`		alternately with successive uses turns on or off the verbose switch. This causes job number information to be output on standard error when job are submitted.
`+verbose`			turns on the verbose switch. There is no default letter option (however it is possible to create one by editing the message file).
`+no-verbose`			turns off the verbose switch. There is no default letter option (however it is possible to create one by editing the message file).
`-w`	`+write-message`		requests a message to be sent to every terminal at which the user is logged in, or by mail otherwise if the user is not logged in, when the job is completed or terminated.
`-x`	`+no-message`		turns off any job completion messages previously requested with `-m` or `-w`
`-z`	`+no-retain`		cancels a request that the job or jobs be retained on the queue after printing.
`-Z`	`+job-size-limit`	*limit*	limit the size of jobs to limit. limit may be a number, giving a size in bytes, or it may be suffixed with P to indicate a number of pages. If a job exceeds the limit it is truncated with a warning message, but a job is still created. If the limit is prefixed with an `E`, then the warning becomes an error, and no job is created. Supply an argument of a single - to turn off this option.
`+freeze-current`			Save all the current options in a `.gnuspool` file in the current directory. This will supply defaults for further gspl-pr commands invoked subsequently when started from the directory. Note that no job will be expected from standard input if no files are specified after including this option.

Option	Arg	Description
`+freeze-home`		Save all the current options in a `.gnuspool` file in the user's home directory. This will supply defaults for further gspl-pr commands invoked subsequently. Note that no job will be expected from standard input if no files are specified after including this option.

7.2.1.2 Environment variables used by gspl-pr

`GSPL_PR`	space-separated options to override defaults for gspl-pr.
`SPRESTCONF`	location of alternative help file.

7.2.1.3 Messages via terminal and e-mail

You can have a message sent to your terminal or receive mail in two circumstances.

1. When your job has:
    1. Completed normally
    2. Been manually deleted (using gspl-pq etc).
    3. Been automatically deleted after remaining on the queue for a specified time as specified using the `-t` or `-T` options.
    4. Been aborted during printing.
2. When your job reaches the top of the queue and:
    1. It has been selected for printing, but operator attention is required for:
        1. Approval of an alignment page
        2. Confirmation to proceed in single job operation
    2. It has not been selected for printing because no printer is available with the selected form type loaded.

The option `-w` causes a message to be sent to your terminal, and the option `-m` causes you to be sent mail in one of the first set of circumstances. If neither is specified, you should still receive mail if a filter process produces output on standard error or terminates abnormally, or if a job is automatically deleted.

(Do not forget you can override the `MAILER` program to change this behaviour if required as described in Chapter 12.1).

The option `-A` causes a message to be sent to your terminal, and the option `-a` causes you to be sent mail in the second set of circumstances.

If any of these are set in the environment and you don't want them, you may suppress them on the gspl-pr command line using `-x` to turn off both the `-w` and `-m` options, and `-b` to turn off the `-A` and `-a` options.

If your terminal cannot be written to, or you have logged out when a message to your terminal is invoked, it will be diverted to the mail program instead.

7.2.1.4 Queue Timeouts

When the gspl-pr or gspl-rpr command receives its data on standard input from a pipe, or terminal device, the job is not normally submitted until an end-of-file indication is encountered. This is typically caused by the termination of the process writing to the pipe, or by typing the end-of-file character (often ctrl-D) when input is being taken from a terminal.

In some circumstances the process writing to the pipe may never finish, or the terminal being read from may not be being accessed by a human, but the user may wish to proceed anyway with printing.

The `-j` option provides a wait timeout in seconds. If some characters have been received, the timeout is set before reading more characters. If the timeout expires before any more characters are read, then a job is created using the characters received so far and gspl-pr or gspl-rpr restarts, possibly making further jobs if more characters arrive.

This cannot be ideal in the case of pipes without some co-operation from the sending process; this is because if stdio (i.e. the C library functions printf etc, which is also used by many other languages and applications) then the output is usually "buffered" in 1024-byte chunks, and thus up to 1023 bytes of the last part of the output will not be written out to the pipe until the sending process decides to send some more or terminates, and therefore all but the last of the jobs created by use of this option may be "short" by up to 1023 bytes.

If the sending process is a 'C' program or other program which the user has access to, then the user should ensure that the routine fflush is used after every block of output, thus:

```
printf("Totals for......\n";, ....);
fflush(stdout);
```

Alternatively, the routine setbuf should be invoked to reduce the buffering on standard output.

If the user does not have any access to the sending process this option may have undesirable effects at "the seams" of the various jobs due to this buffering. The only successful approach would be to insert a filter process in between gspl-pr and the sending process to ensure that complete pages only were passed through to gspl-pr. It is considered unacceptable to monopolise a printer on speculation that further data may arrive.

A parameter of zero turns off the `-j` option if it has been set in an environment variable or `.gnuspool` file.

7.2.1.5 Delay times

The `-n` and `-N` options provide for the job to be held unprinted on the queue for, or until, a specific time.

The `-n` option provides a time interval to be held for, in minutes, in hours and minutes, or in hours, minutes and seconds. The following examples all specify the same time interval of 1 hour and 30 minutes. If no colon appears, a time period of minutes is assumed:

```
-n 90
-n 1:30
-n 1:30:00
```

The `-N` option provides an explicit time and possibly a date to be held until. The time may be specified as a 24-hour clock time with optional seconds thus

```
-N 16:35
-N 04:28:32
```

In these cases the given time in the next 24 hours is taken to be the required time. If a different date is required this can be put in front of the time in the format yy/mm/dd with a comma thus:

```
-N 91/2/12,12:30
```

The year may be omitted, and the date will be taken as a future date. The date will be taken as *dd/mm* for timezones less than 4 West, otherwise *mm/dd*.

This allows the printing of long jobs to be printed at a quiet time, overnight for example. Alternatively specifying a delay time can provide an opportunity for thought and possible amendment before continuing.

7.2.1.6 Interaction of gspl-pr and gspl-rpr

If the `-Q` option is given to gspl-pr, either on the command line or in the environment or `.gnuspool` files, then gspl-rpr will be invoked with the same command line options (but gspl-rpr will accept its own `gspl-rpr` environment variable or `.gnuspool` file options).

gspl-rpr, however, will not invoke gspl-pr if the `-Q` option is not given; this is an error. Remember that the `-Q` option could be set from a `GSPL_PR` definition in a `.gnuspool` file, but without a `-Q` option on an `gspl-rpr` definition, this would mean that gspl-rpr did not get invoked with a `-Q` option, so this could give an infinite loop.

7.2.2 gspl-rpr

```
gspl-rpr [ -options ] [ file ... ]
```

gspl-rpr queues up one or more print jobs to a remote queue running the **GNUspool** spooler. **GNUspool** does not need to be running on the submitting machine; only the message file and hosts file are required.

If one or more files are specified, each file is treated as a separate spool job, with a title constructed from the file name. If no files are specified gspl-rpr reads from the standard input instead. In this case the job is placed on the queue with a blank title unless otherwise specified.

The standard default options are to print one copy, preceded by a banner page, with the priority, printer and form types specified for the user. The standard banner page will contain: the user name and job title (in large characters), the name of the user who submitted the job (if different), the job number, its size and priority.

Other default options may be set up using the environment variable `gspl-rpr` and/or

`.gnuspool` files. Any defaults may be overridden by re-specifying that option on the command line.

7.2.2.1 OPTIONS

The environment variable on which options are supplied is `gspl-rpr` for GSPL_RPR and the environment variable to specify the help file is `SPRESTCONF`.

As supplied, the options are identical to gspl-pr except that `-E` `-o` and `-U` are not supported.

`-?`	`+explain`		causes a summary of the other options to be displayed without taking further action.
`-a`	`+mail-attention`		requests that a message be sent in the mail to the user if a job is awaiting attention, either because it reaches the top of the queue and the required form type is currently unavailable or because the form requires alignment.
`-A`	`+write-attention`		requests that a message be sent to every terminal at which the user is logged in, or by mail otherwise if the user is not logged in, if a job is awaiting attention, either because it reaches the top of the queue and the required form type is currently unavailable or because the form requires alignment.
`-b`	`+noatt-message`		turns off any job attention messages previously requested.
`-c`	`+copies`	*n*	sets the number of copies to be printed to n, which may possibly be zero. The maximum number of copies at a time may be restricted to a number for each user such as 10. See the output from gspl-user on the appropriate host for information on the maximum number of copies.
`-C`	`+classcode`	*nnnn*	where nnnn is some collection of the letters `A` through to `P` inclusive (upper or lower case) with `-` to indicate ranges, requests that the job be given the specified class code. The class code of a job may restrict which printers the job may be printed on, or which other users can display details of the job. In normal circumstances (except where the user has override class privilege), the specified class code is reduced to the intersection of (i.e. is "anded" with) the specified class code and the user's own class code.
`-d`	`+delimiter-number`		is used in conjunction with the `-Dstring` option to specify that a page is to be considered complete after reading n delimiter strings. The default value is 1.

-D	+delimiter	*string*	specifies an alternative delimiter to the default of formfeed as being the end of a page. If a number is given using the -dn option, then the page is considered ended after that number of occurrences of the delimiter. The string given may contain standard constructs such as `\n \r \t` etc to denote linefeed, carriage return etc. Remember that when the job is printed, the printer's setup file may re-specify the delimiter, which will replace the delimiter specified by these options.
-f	+formtype	*string*	specifies that the supplied form type (with optional suffix introduced by "`.`" or "`-`") is used. Note that not all users may be allowed to select form types other some pattern, causing the job to be rejected.
-F	+post-proc-flags	*string*	causes the specified string to be passed as the value of the environment variable `SPOOLFLAGS` to any program invoked at the time the job is printed. This might be as a post-processing filter command, or alternatively some shell command invoked as (for example) a `docstart` string.
-h	+header	*string*	supplies string as the title for each job, otherwise it is derived from the file name specified. The title appears on the banner page, if this is printed.
-i	+interpolate		causes any file name suffixes `.c`, `.o` etc of the files from which jobs are created, to be appended to the form type, unless it already has a suffix, for example: `gspl-rpr -i -f a4 x.c y.ps` would submit jobs using the files `x.`c and `y.ps` with form types `a4.c` and `a4.ps` respectively.
-I	+no-interpolate		cancels any previously-specified `-i` option.
-j	+job-wait-time	*n*	is relevant only if input to gspl-rpr is from a pipeline as in `program|gspl-rpr ....` This causes a job to be submitted if no data is received over the pipe for n seconds. gspl-rpr will still wait, and may submit any number of subsequent jobs until the pipe is closed by the program generating the output. Note that nothing can be done by gspl-rpr to guess at what output might appear (possibly halfway through a line) if the program writing to the pipe does not flush its output buffer after each occasion that it writes something. Specify a parameter of zero to cancel any previous `-j` option, if required.

`-l`	`+local-only`		in a networked environment requests that the job be printed only on the printers local to the machine at which the job is submitted.
`-L`	`+network-wide`		cancels the request that the job or jobs be printed only on printers local to the machine at which the job is submitted.
`-m`	`+mail-message`		requests a message to be sent in the mail to the user on completion or termination of the job.
`-n`	`+delay-for`	*nn*	specifies that the job should be held on the queue for at least nn minutes from the current time before being printed. The time may alternatively be specified as *hh:mm* or as *hh:mm:ss*, specifying a delay in hours and minutes, or hours, minutes and seconds.
`-N`	`+delay-until`	*time*	is an alternative to the `-n` option to specify the earliest time at which the job is to be printed. The argument may be *hh:mm* or *hh:mm:ss* to give the time of day in 24-hour clock notation. If the time has passed, then tomorrow is assumed. Alternatively a date and a comma may be prefixed to the time in the form *mm/dd* or *dd/mm* depending upon the local convention for date format used. Thus the resulting argument might be `10/11,12:30`
`-O`	`+odd-even-flags`	*flag*	where flag is one of `O`, `E`, `A`, `B` or `-` (the letters may be upper or lower case), cause odd or even-numbered pages to be skipped. or - to reset this flag. `O` causes odd-numbered pages not to be printed. `E` causes even-numbered pages not to be printed. `A` and `B` are useful if more than one copy is to be printed. `A` causes even-numbered pages not to be printed on odd-numbered copies, and odd-numbered pages not to be printed on even-numbered copies. `B` is the other way around. If you do not understand this, all you have to do is remember that `gspl-rpr -c2 -OA ....` prints all the odd-numbered pages followed by all the even-numbered ones. For this to work properly, page delimiters must be set appropriately.
`-p`	`+priority`	*n*	specifies the priority of the job, between 1 (lowest) and 255 (highest) or some narrower range to which the user is limited. Increasing the priority of a job increases its chances of being printed earlier than it otherwise would be whilst increasing the charge applied to the user in a non-linear fashion.

`-P`	`+printer`	*name*	specifies that the job is to be sent to a printer with the name given, as opposed to printing it on the first available printer with the given form type. name may be a pattern to select any printer matching the pattern. A user may be limited to a range of printers which must be a superset of name. To "turn off" a printer name previously specified by a preceding `-Pname` option, put a single `-` sign as the printer name.
`-q`	`+retain`		requests that the job or jobs be retained on the queue with copies set to zero after printing, for explicit deletion, or automatically at the expiry of the timeout (as set by the `-tn` option).
`-Q`	`+host`	*host*	send the job or jobs to the given hostname. Note that hostname must be in `/etc/gnuspool.hostss` on the submitting machine and the submitting machine's hostname must be in `/etc/gnuspool.hostss` on the receiving machine.
`-r`	`+banner`		restores banner pages previously suppressed using `-s`. Note that some form types may be set up never to use banners regardless of this option.
`-R`	`+page-range`	*m−n*	Specifies that pages `m` through to `n` inclusive are to be printed. This does of course assume that the job has recognisable pages. If `m` or `n` are omitted, then "the beginning" or "the end" respectively is assumed, so `-R 3-5` would print pages 3 to 5, `-7` would print pages 1 to 7 inclusive, and `-R 4-` would print page 4 to the end inclusive. `-R 1-` would turn this option off by selecting 1 to the end.
`-s`	`+no-banner`		suppresses any banner page (large letter user name etc) which is printed before the job itself. Note that some form types may be set up always to print banners regardless of this option.
`-t`	`+printed-timeout`	*n*	
`-T`	`+not-printed-timeout`	*n*	specifies that if held on the queue without being printed, the job will be deleted automatically after n hours. The default value is 168 hours (1 week), and the maximum value is 32767 hours (nearly 4 years).
`-u`	`+post-user`	*name*	requests that the specified user name be substituted for the submitting user on the banner page optionally printed at the start of the job. The job still remains the responsibility of the submitting user. To "turn off" a user name specified in a previous `-u` option, put a single `-` sign as the user name. The user name must exist on the machine to which the job is queued, but this will be ignored if not.

`-V`	`+toggle-verbose`		alternately with successive uses turns on or off the verbose switch. This causes job number information to be output on standard error when job are submitted.
`+verbose`			turns on the verbose switch. There is no default letter option (however it is possible to create one by editing the message file).
`+no-verbose`			turns off the verbose switch. There is no default letter option (however it is possible to create one by editing the message file).
`-w`	`+write-message`		requests a message to be sent to every terminal at which the user is logged in, or by mail otherwise if the user is not logged in, when the job is completed or terminated.
`-x`	`+no-message`		turns off any job completion messages previously requested with `-m` or `-w`
`-z`	`+no-retain`		cancels a request that the job or jobs be retained on the queue after printing.
`-Z`	`+job-size-limit`	*limit*	limit the size of jobs to limit. limit may be a number, giving a size in bytes, or it may be suffixed with `P` to indicate a number of pages. If a job exceeds the limit it is truncated with a warning message, but a job is still created. If the limit is prefixed with an `E`, then the warning becomes an error, and no job is created. Supply an argument of a single `-` to turn off this option.
`+freeze-current`			Save all the current options in a `.gnuspool` file in the current directory. This will supply defaults for further gspl-rpr commands invoked subsequently when started from the directory. Note that no job will be expected from standard input if no files are specified after including this option.
`+freeze-home`			Save all the current options in a `.gnuspool` file in the user's home directory. This will supply defaults for further gspl-rpr commands invoked subsequently. Note that no job will be expected from standard input if no files are specified after including this option.

7.2.2.2 Environment variables used by gspl-rpr

`GSPL_RPR`	space-separated options to override defaults for gspl-rpr.
`SPRESTCONF`	location of alternative help file.

7.2.3 gspl-mpr

```
gspl-mpr [ -options ] [ file ... ]
```

gspl-mpr either queues up one or more print jobs for the **GNUspool** spooler using gspl-pr if it is running, or else places a series of files in an alternative spool directory, such that when run as shell scripts when the **GNUspool** spooler process spshed is running, the jobs are queued as originally specified.

7.2.3.1 Options for gspl-mpr

Options are as per gspl-pr. If the job can be queued immediately, this is directly invoked.

7.2.3.2 Exit codes

An exit code of zero is returned if the job or jobs are successfully queued using gspl-pr or successfully saved in the alternate directory. Any non-zero exit code from gspl-pr will be passed back to the user. Standard exit codes and error messages are displayed if the files cannot be opened or saved.

7.2.3.3 Files and directories used under /usr/local

`var/gnuspool_alt/CMDF*`	command file (shell script)
`var/gnuspool_alt/ASPF*`	data file (contents of job)

7.2.3.4 Environment variables

`GSPL_MPR`	space-separated options to override defaults for gspl-mpr.
`ALTSPD`	location of alternate spool directory, if not `/usr/local/var/gnuspool_alt`.

7.2.4 gspl-mrpr

```
gspl-mrpr [ -options ] [ file ... ]
```

gspl-mrpr queues up one or more print jobs to a remote queue running the **GNUspool** spooler using gspl-rpr if it is available, or otherwise places a series of files in an alternative spool directory for queueing when the remote machine is available.

7.2.4.1 Options for gspl-mrpr

Options are as per gspl-rpr. If the job can be queued immediately, this is directly invoked.

7.2.4.2 Exit codes

An exit code of zero is returned if the job or jobs are successfully queued using gspl-rpr or successfully saved in the alternate directory. Any non-zero exit code from gspl-rpr will be passed back to the user. Standard exit codes and error messages are displayed if the files cannot be opened or saved.

7.2.4.3 Files and directories used under /usr/local

`var/gnuspool_alt/CMDF*`	command file (shell script)

`var/gnuspool_alt/ASPF*`	data file (contents of job)

7.2.4.4 Environment variables used

GSPL_MRPR	space-separated options to override defaults for gspl-mrpr.
ALTSPD	location of alternate spool directory, if not `/usr/local/var/gnuspool_alt`.

7.2.5 gspl-lpcover

```
gspl-lpcover [ -options ] [ file ... ]
```

gspl-lpcover may be used instead of the standard lp command. It converts the options that most lp commands take to the equivalent gspl-pr commands and then invokes gspl-pr to submit the print jobs.

The gspl-pr program provides a much greater set of facilities than lp. Although it is strongly recommended to switch to the gspl-pr command to take advantage of these, they can be made available to users of our gspl-lpcover command. This is done by setting up GSPL_PR in the application or user environment.

gspl-lpcover is usually installed in place of lp in `/usr/bin`, with the original binary moved to something like `old.lp` in the same directory.

7.2.5.1 Options taken by gspl-lpcover

`-c`	This has no effect, but generates a warning message on standard error, as with **GNUspool** spool files are always copied.
`-d name`	Specify printer destination as name. The destination may also be specified in the environment variable `LPDEST`, which this option overrides. An argument of any to `-d` cancels any existing `LPDEST` environment variable. This is translated directly into a `-P` option to GSPL-PR.
`-f form`	specifies the supplied form type. This is translated directly into an `-f` option to gspl-pr.
`-H arg`	There is no direct mapping for the `-H` argument.
`-m`	Mail user. This is translated directly into an `-m` option to gspl-pr.
`-n n`	Specify the number of copies. This is translated directly into a `-c` option to gspl-pr.
`-o arg`	Only `-onobanner`, to suppress the header page, is interpreted and translated to the `-s` option to gspl-pr.
`-P arg`	This is translated into `-R` page range arguments for gspl-pr.
`-q n`	This is translated into a `-p` argument for gspl-pr scaled appropriately, taking into account the user's maximum, minimum and default priorities.
`-S`	This option is ignored.
`-s`	This turns off the `-v` argument for gspl-pr which is normally included.
`-T`	This option is ignored.
`-t <arg>`	This is translated to a `-h` (title) argument to gspl-pr.
`-w`	This is translated directly to the `-w` (write message) option to gspl-pr.

`-y`	This option is ignored.

7.2.5.2 Environment variables used

`LPDEST`	Specify destination for gspl-lpcover.

7.2.5.3 Diagnostics

Various warnings about unimplemented features are printed by gspl-lpcover.

Others may be generated by software invoked.

7.3 Managing the spooler

7.3.1 gspl-start

```
gspl-start [ jobsize [ printersize ] ]
```

gspl-start without arguments, or with one or two numeric arguments, may be used to start the **GNUspool** scheduler, spshed, together with associated programs, such as xtnetserv.

The first optional argument specifies to spshed that when allocating an initial block of shared memory (or memory-mapped file) for jobs it should allocate memory for at least the specified number of jobs.

The second optional argument performs the same function with printers.

These should not strictly be necessary, but on some platforms there are strange problems about allocating and deallocating blocks of shared memory whilst a program is running which these are intended to alleviate.

If spshed is already running, gspl-start will have no effect.

Printers with numeric names in the other form of gspl-start will not cause confusion if spshed is running.

All other forms of this command will initialise spshed, but this is **very strongly** not recommended. The scheduler should be started first, connections to other machines established, possibly using gspl-conn, and then printers should be started.

7.3.2 gspl-stop

```
gspl-stop [ -y ]
```

gspl-stop is a program which should be invoked to bring the **GNUspool** spooler system to an orderly halt prior to system shutdown. Any jobs in progress will be saved, and the current page noted for resumption on each print job currently in progress when the spooler is restarted. Confirmation is requested (reply `y` or `n`) unless the `-y` option is specified.

Only a user with the *Stop spooler* privilege may successfully invoke this command.

7.3.2.1 Options for gspl-stop

Note that there is no means of saving the `-y` option in a `.gnuspool` file as this is not considered to be sensible. If you have to do this, put the command in a shell "wrapper".

`-y` Do not ask for confirmation.

7.3.3 Gspl-conn

```
gspl-conn host
```

gspl-conn instructs the **GNUspool** scheduler to attempt to raise a connection to the given host, which should be specified in the file `/usr/local/etc/gnuspool.hosts`, and not currently active.

It returns immediately; however the connection may take some time and indeed may not succeed. If it does not succeed, then the command will not return an error message or code – look in the system error log file `spshed_reps` to see what happened.

The user must have *stop scheduler* permission to run this command.

7.3.4 Gspl-disconn

```
gspl-disconn host
```

gspl-disconn instructs the **GNUspool** scheduler to close a connection to the given host, which should be specified in the file `/usr/local/etc/gnuspool.hosts`, and currently active.

The command returns immediately; however the disconnection may take some time and indeed may not succeed.

The user must have *stop scheduler* permission to run this command.

7.3.5 gspl-suspend

```
gspl-suspend [ -y ] [ n ]
```

gspl-suspend halts the assignment of jobs to printers for a short time, which defaults to 5 minutes or as given as a time in seconds last argument to the command, or until released by gspl-release. Confirmation is requested unless the `-y` option is given.

Only a user with the *Stop spooler* privilege may successfully invoke this command.

The reason why gspl-suspend this may be useful is that in a network scenario, where a new host is brought on line, a considerable amount of network traffic is created, and the assignment of printers to jobs and the broadcasting of information between hosts can flood the network to an unacceptable degree.

7.3.5.1 OPTIONS

Note that there is no means of saving the `-y` option in a `.gnuspool` file as this is not considered to be sensible. If you have to do this, put the command in a shell "wrapper".

`-y`	Do not ask for confirmation.

7.3.6 gspl-release

```
gspl-release
```

gspl-release releases scheduling again where it has previously been halted with gspl-suspend. It has no effect if scheduling is not currently suspended. There are no options.

Only a user with the *Stop spooler* privilege may successfully invoke this command.

7.3.7 gspl-hostedit

```
gspl-hostedit [ -o file ] [ -s arg ] [ -I ] [ file ]
```

gspl-hostedit is a simple curses-based program to edit host tables for `/usr/local/etc/gnuspool.hosts`, the host table for GNUspool.

It knows about local addresses (selecting the correct IP address for a machine with different IPs on different networks), Windows clients, DHCP, trusted hosts, manual connections, probes and timeouts.

Input is taken from standard input unless a file name is given, and output is to standard output unless the `-o` option is given.

Normally this would be run as follows:

```
gspl-hostedit -I /usr/local/etc/gnuspool.hosts
```

You will usually have to stop and restart **GNUspool** after you have done this so that all parts of the system "know" about the new hosts, however this may not be necessary in all cases, you may only have to "`kill -1`" the process id of the xtnetserv process.

7.3.7.1 Options for gspl-hostedit

`-o file`	Output to the named file rather than Standard Output
`-s char`	Where char is `h` or `i`, sort display by host name or by IP address.
`-I`	Edit in place; a file argument must be given.

7.3.7.2 Screen commands

The following command keys are used from within the screen displayed by gspl-hostedit. As with other GNUspool commands, any commands which operate upon an existing item will do so with the item to which the cursor is moved.

`k` or cursor up	Move cursor up.
`J` or cursor down	Move cursor down.
`N` or next page	Scroll down a screenful.
`P` or previous page	Scroll up a screenful.

`q`	Quit and write hosts file.
`a`	Create a new hosts entry.
`c`	Edit the selected hosts entry.
`d`	Delete the selected hosts entry.
`l`	Edit the local hosts entry.
`u`	Set the default user name for DHCP clients.

7.4 Managing print jobs (command line)

7.4.1 gspl-qlist

```
gspl-qlist [ -options ] [ job number ... ]
```

gspl-qlist outputs a table on standard output showing jobs on the **GNUspool** spool queue, including those queued on other hosts on the network.

Each line of the output corresponds to a single job, and by default the output is generally similar to the default format of the jobs screen of the gspl-pq command, something like:

```
0024139     jmc report1 a4      23K 1 150
0026921     jmc prgm.c  listing 12K 1 150
red:0021941 pbt report  a4.lh   17K 2 150
```

The first field on each line is the numeric job number, preceded by a host name and a colon if on a remote machine.

Various options described below enable the user to control the output in various ways. The job data itself may also be extracted.

To limit the output to information about a given job or jobs, job numbers may be placed on the command line.

7.4.1.1 Options for gspl-qlist

The environment variable on which options are supplied is `GSPL_QLIST` and the environment variable to specify the help file is `SPRESTCONF`.

Option		Arg	Description
`-?`	`+explain`		causes a summary of the other options to be displayed without taking further action.
`-C`	`+classcode`	*nnnn*	restricts attention to jobs matching the class code given. This may be reduced further by the user's class code unless he/she has *override class* privilege.
`-D`	`+default-format`		cancel a `-F` option and restore the default format.
`-e`	`+no-page-counts`		do not try to represent page numbers and counts on the output (this only affects the default format).
`-E`	`+page-counts`		do try to represent page numbers and counts on the output (this only affects the default format).

Option		Arg	Description
`-F`	`+format`	*string*	Specify user defined output format. The string argument contains characters to be printed and symbols representing the required data fields. This option overrides and preceding `-D` option, and is itself overridden by a following `-D` option.
`-H`	`+header`		generate a header for each column of the output according to the field displayed there. The headers are taken from the message file, by default `/usr/local/share/gnuspool/help/rest.help`, which may be adjusted as required.
`-l`	`+local-only`		display only jobs local to the current host.
`-N`	`+no-header`		cancel the `-H` option.
`-n`	`+no-view-job`		cancel the `-V` option.

`-P`	`+view-pages`	*range*	with the `-V` option, specify a range of pages to output. A range may consist of single page numbers or subranges separated by commas, thus:
			`-V -P 1`
			`-V -P 1,3,4-9`
			`-V -P 2-7,12-`
			such arguments would respectively select page 1, pages 1, 3 and 4 to 9 and pages 2 to 7 and 12 to the end. Omitted numbers either side of the `-` denote "the beginning" and "the end".

`-p`	`+printed-jobs`		restricts attention to jobs which have been printed.
`-q`	`+list-only`	*name*	List only jobs for, or which could be printed on the specified subset of printers. The set of printers may contain just one name, a list of names or a list of patters for matching printer names. Shell-like pattern matching applies to the pattern. You may well need to put quotes around the pattern to avoid misinterpretation by the shell.
`-r`	`+remotes`		display jobs local to the current host and connected hosts.
`-t`	`+just-title`	*string*	restricts attention to jobs whose title matches the pattern given.
`-u`	`+just-user`	*user*	restricts attention to jobs owned by the given user or list of users given by a pattern.
`-V`	`+view-job`		cancel the job listing output and send the text of the job or jobs to standard output.
`-Y`	`+all-jobs`		displays all jobs, cancelling `-y` or `-p`.
`-y`	`+unprinted-jobs`		restricts attention to jobs which have not been printed.
`-z`	`+include-null`		when listing jobs restricted by a `-q` argument, include jobs with no destination printer.

Option		Arg	Description
`-Z`	`+no-include-null`		when listing jobs and printers restricted by a `-q` argument, do not include jobs without a destination printer.
`+freeze-current`			Save all the current options in a `.gnuspool` file in the current directory.
`+freeze-home`			Save all the current options in a `.gnuspool` file in the user's home directory.

7.4.1.2 Patterns

The `-q`, `-t` and `-u` options may all be given a name, a comma-separated list of names or shell-like wildcard patterns. The patterns all take the standard form:

`*`	matches anything
`?`	matches a single character
`[a-mp-ru]`	matches any one character in the range of characters given
`[!n-z9]`	matches any one character not in the range of characters given

Be careful to enclose these sequences in quotes in case the shell misinterprets them.

For example

```
gspl-qlist -q ljet,ptr3
gspl-qlist -q 'ptr*,lp[1-9]'
gspl-qlist -u 'j*'
gspl-qlist -t 'Acc*'
```

7.4.1.3 Format codes

The format string consists of a string containing the following character sequences, which are replaced by the corresponding job parameters. The string may contain various other printing characters or spaces as required.

Each column is padded, usually on the right, to the length of the longest entry. If a header is requested, the appropriate abbreviation is obtained from the message file and inserted.

`%%`	insert a single `%` sign
`%A`	wattn if the write attention message flag is set.
`%a`	mattn if the mail attention message flag is set.
`%C`	class code
`%c`	copies (right-justified)
`%D`	page delimiter
`%d`	delimiter number
`%F`	post-processing flags
`%f`	form type
`%G`	"halted at" page number.
`%H`	hold time

`%h`	job title
`%K`	size of job in bytes
`%L`	position reached in bytes.
`%l`	`local` if local job only
`%m`	`mail` if mail completion flag set.
`%N`	job number, prefixed by host name if remote
`%O`	odd/even/swap flags.
`%o`	originating host (might be different if via gspl-rpr)
`%P`	printer destination
`%p`	priority.
`%Q`	page reached.
`%q`	retain if retained on queue
`%R`	page range
`%S`	size in pages
`%s`	`nohdr` if header suppressed
`%T`	delete time if not printed
`%t`	delete time if printed
`%U`	user name to be posted to
`%u`	user name owned by
`%W`	submission or start time
`%w`	`write` if write completion message flag set

Note that the various strings such as `nohdr` etc are read from the message file also, so it is possible to modify them as required by the user.

The default format is

```
%N %u %h %f %Q %S %c %p %P
```

with the (default) `-E` option and

```
%N %u %h %f %L %K %c %p %P
```

with the `-e` option.

7.4.2 gspl-qdel

```
gspl-qdel [ -options ] job number ...
```

gspl-qdel deletes one or more jobs from the **GNUspool** job spool queue, which may be on the local host or other hosts on the network. Jobs are specified by the job number, which is reported by gspl-qlist in the first column of the default output, or by gspl-pr or gspl-rpr when the job is queued with the verbose option.

Remote jobs are specified by prefixing the job number by the host name and a colon, thus `host:1234`. Leading zeroes in the job number may be omitted.

gspl-qdel will not delete jobs not marked as "printed" unless the `-y` option is given.

7.4.2.1 Options

The environment variable on which options may be supplied is `GSPL_QDEL` and the environment variable to specify the help file is `SPRESTCONF`.

Option		Arg	Description
`-?`	`+explain`		causes a summary of the other options to be displayed without taking further action.
`-C`	`+classcode`	*nnnn*	may be used to set a class code to narrow or expand the range of jobs from which jobs are selected. It is probably only relevant where the user has *override class* privilege and it is required to expand the default class code to select additional jobs.
`-d`	`+delete`		cancel a previously-specified `-k` option.
`-D`	`+directory`	*name*	Save unqueued jobs to name rather than the current directory.
`-e`	`+do-not-unqueue`		cancel a previously-specified `-u` option. The jobs will just be deleted, not unqueued.
`-J`	`+job-prefix`	*name*	specify the given name as the prefix for the job file, followed by the job number, to be used by the `-u` option rather than the default of `J` (which in turn may be changed by editing the message file).
`-k`	`+do-not-delete`		when performing an unqueue as specified with the `-u` option, do not delete the job at the end.
`-n`	`+keep-unprinted`		is the default case. Jobs which have not yet been printed are not deleted.
`-S`	`+command-prefix`	*name*	specify the given name as the prefix for the command file, followed by the job number, to be used by the `-u` option rather than the default of `C` (which in turn may be changed by editing the message file).
`-u`	`+unqueue`		unqueue jobs, i.e. copy to current directory or directory selected with the `-D` option. Jobs are copied as two files, a job file containing the text of the job and a command file containing an gspl-pr command. The command file may be run as a shell script to re-queue the job with the same options.
`-y`	`+delete-unprinted`		cancels all jobs, whether printed or not.
`+freeze-current`			Save all the current options in a `.gnuspool` file in the current directory. If there are no jobs given to operate on and this is specified, then gspl-qdel will quit without error. If it is unable to save the options and jobs are specified, then gspl-qdel will continue.
`+freeze-home`			Save all the current options in a `.gnuspool` file in the user's home directory. If there are no jobs given to operate on and this is specified, then gspl-qdel will quit without error. If it is unable to save the options and jobs are specified, then gspl-qdel will continue.

7.4.2.2 NOTES

The `-C` option selects the class code for finding the jobs; if you have *override class* privilege, but your default class code does not allow the jobs "to be seen" in the output of gspl-qlist, then you may need to specify this.

Any errors are reported with appropriate messages on standard error. If any errors are detected operating on several jobs, the exit code will correspond to the last such error displayed.

7.4.3 gspl-qchange

```
gspl-qchange [ -options ] job number ...
```

gspl-qchange makes changes to one or more **GNUspool** spooled jobs, which may be on the local host or other hosts on the network. Jobs are specified by the job number, which is reported by gspl-qlist in the first column of the default output, or by gspl-pr or gspl-rpr when the job is queued with the verbose option. Remote jobs are specified by prefixing the job number by the host name and a colon, thus `host:1234`. Leading zeroes in the job number may be omitted.

Each of the jobs must be owned by the user, or the user must have permission to edit other user's jobs. The user must also have permission to edit remote jobs to operate on remote jobs (this is normally set for all users).

The changes are specified by options, which are as similar as possible to those for gspl-pr. It is regarded as a mistake not to specify at least one option.

7.4.3.1 OPTIONS

The environment variable on which options are supplied is `GSPL_QCHANGE` and the environment variable to specify the help file is `SPRESTCONF`.

Option		Arg	Description
`-?`	`+explain`		causes a summary of the other options to be displayed without taking further action.
`-a`	`+mail-attention`		requests that a message be sent in the mail to the user if a job is awaiting attention, either because it reaches the top of the queue and the required form type is currently unavailable or because the form requires alignment.
`-A`	`+write-attention`		requests that a message be sent to every terminal at which the user is logged in, or by mail otherwise if the user is not logged in, if a job is awaiting attention, either because it reaches the top of the queue and the required form type is currently unavailable or because the form requires alignment.
`-b`	`+noatt-message`		turns off any job attention messages previously requested.

Option		Arg	Description
`-c`	`+copies`	n	sets the number of copies to be printed to n. The maximum number of copies at a time may be restricted to a number for each user such as 10. See the output from gspl-user for information on the maximum number of copies. If the user has *set any priority on queue* privilege, this limit is removed.
`-C`	`+classcode`	nnnn	may be used to set a class code narrow or expand the range of jobs from which jobs are selected. It is probably only relevant where the user has *override class* privilege and it is required to expand the default class code to select additional jobs. ***N.B.*** This does not set the classcode. Use `-S` to set the classcode.
`-f`	`+formtype`	string	specifies that the supplied form type (with optional suffix introduced by "`.`" or "`-`") is used. Note that not all users may be allowed to select form types other some pattern, however only the "paper type", i.e. the form type up to the suffix is considered when this restriction is active, so the user can still change the form type suffix if required. Note that there is no equivalent of the `-i` (interpolate) option to gspl-pr.
`-F`	`+post-proc-flags`	string	changes to the specified string, the data to be passed as the value of the environment variable `SPOOLFLAGS` to any program invoked at the time the job is printed. This might be as a post-processing filter command, or alternatively some shell command invoked as (for example) a `docstart` string.
`-h`	`+header`	string	replace the title for each job with string. The title appears on the banner page, if this is printed.
`-l`	`+local-only`		in a networked environment requests that the job be printed only on the printers local to the machine at which the job is submitted.
`-L`	`+network-wide`		cancels the request that the job or jobs be printed only on printers local to the machine at which the job is submitted.
`-m`	`+mail-message`		requests a message to be sent in the mail to the user on completion or termination of the job.
`-n`	`+delay-for`	*nn*	specifies that the job should be held on the queue for at least nn minutes from the current time before being printed. The time may alternatively be specified as *hh:mm* or as *hh:mm:ss*, specifying a delay in hours and minutes, or hours, minutes and seconds. Specify zero to cancel this.

Option		Arg	Description
`-N`	`+delay-until`	*time*	is an alternative to the `-n` option to specify the earliest time at which the job is to be printed. The argument may be *hh:mm* or *hh:mm:ss* to give the time of day in 24-hour clock notation. If the time has passed, then tomorrow is assumed. Alternatively a date and a comma may be prefixed to the time in the form *mm/dd* or *dd/mm* depending upon the local convention for date format used. Thus the resulting argument might be `10/11,12:30`
`-O`	`+odd-even-flags`	*flag*	where flag is one of `O`, `E`, `A`, `B` or `-` cause odd or even-numbered pages to be skipped. `O` causes odd-numbered pages not to be printed. `E` causes even-numbered pages not to be printed. `A` and B are useful if more than one copy is to be printed. `A` causes even-numbered pages not to be printed on odd-numbered copies, and odd-numbered pages not to be printed on even-numbered copies. `B` is the other way around. If you do not understand this, all you have to do is remember that `gspl-qchange -c2 -OA ....` prints all the odd-numbered pages followed by all the even-numbered ones.
`-p`	`+priority`	*n*	specifies the priority of the job, between 1 (lowest) and 255 (highest) or some narrower range to which the user is limited, unless he or she has the any priority on queue privilege. A user must have the *change priority on queue* privilege to invoke this option.
`-P`	`+printer`	*name*	specify *name* as the destination printer for the job. *name* may be a pattern to select any printer matching the pattern, or `-` to cancel the selection and make it the first available printer. A user may be limited to a range of printers which must be a superset of name.
`-q`	`+retain`		requests that the job or jobs be retained on the queue with copies set to zero after printing, for explicit deletion, or automatically at the expiry of the timeout (as set by the `-t` option).
`-r`	`+banner`		restores banner pages previously suppressed using `-s`. Note that some form types may be set up never to use banners regardless of this option.
`-R`	`+page-range`	*m-n*	specifies that pages `m` through to `n` inclusive are to be printed. This does of course assume that the job has recognisable pages. If m or n are omitted, then "the beginning" or "the end" respectively is assumed.

Option		Arg	Description
`-s`	`+no-banner`		suppresses any banner page (large letter user name etc) which is printed before the job itself. Note that some form types may be set up always to print banners regardless of this option.
`-S`	`+set-classcode`	*nnnn*	where nnnn is some collection of the letters `A` through to `P` inclusive (upper or lower case) with `-` to indicate ranges, requests that the job be given the specified class code. The class code of a job may restrict which printers the job may be printed on, or which other users can display details of the job. In normal circumstances (except where the user has *override class* privilege as displayed by gspl-user), the specified class code is reduced to the intersection of (i.e. anded with) the specified class code and the user's own class code.
`-t`	`+printed-timeout`	*n*	specifies that if retained on the queue the job will be deleted automatically after *n* hours. The default value is 24 hours, and the maximum value is 32767 hours (nearly 4 years).
`-T`	`+not-printed-timeout`	*n*	specifies that if held on the queue without being printed, the job will be deleted automatically after *n* hours. The default value is 168 hours (1 week), and the maximum value is 32767 hours (nearly 4 years).
`-u`	`+post-user`	*name*	requests that the specified user name be substituted for the submitting user on the banner page optionally printed at the start of the job. The job still remains the responsibility of the submitting user. To "turn off" a user name specified in a previous `-u` option, put a single `-` sign as the user name.
`-w`	`+write-message`		requests a message to be sent to every terminal at which the user is logged in, or by mail otherwise if the user is not logged in, when the job is completed or terminated.
`-x`	`+no-message`		turns off any job completion messages (`-m` or `-w`) previously requested.
`-z`	`+no-retain`		cancels a request that the job or jobs be retained on the queue after printing.
`+freeze-current`			Save all the current options in a `.gnuspool` file in the current directory. If there are no jobs given to operate on and this is specified, then gspl-qchange will quit without error. If it is unable to save the options and jobs are specified, then gspl-qchange will continue.
`+freeze-home`			Save all the current options in a `.gnuspool` file in the user's home directory. If there are no jobs given to operate on and this is specified, then gspl-qchange will quit without error. If it is unable to save the options and jobs are specified, then gspl-qchange will continue.

7.4.3.2 Notes

The `-C` option selects the class code for finding the jobs; if you have override class privilege, but your default class code does not allow the jobs "to be seen" in the output of gspl-qlist, then you may need to specify this. The `-S` option sets a new classcode, which may be anded with your default class if you do not have *override class* privilege.

Any errors are reported with appropriate messages on standard error. If any errors are detected operating on several jobs, the exit code will correspond to the last such error displayed.

7.4.3.3 Examples

To change the number of copies of job `24139` to 3

```
gspl-qchange -c3 14139
```

To changed the formtype to a4.ps for 3 jobs including one on a remote machine, avon

```
gspl-qchange -f a4.ps 24139 26921 avon:21941
```

Changing all the jobs associated with printer `laser` to come out on printer `ljet` may be done with a shell script like this:

```
gspl-qlist -F "%N %P" | grep laser | while read JOB REST
do
      gspl-qchange -P ljet $JOB
done
```

Or you can do it in one operation with this "one-liner":

```
gspl-qchange -Pljet `gspl-qlist -F"%N %P"|grep laser|sed 's/\(.*\)
.*/\1/'`
```

7.4.4 gspl-lpstatc

```
gspl-lpstatc [ -options ]
cancel jobnumber [ ... ]
```

gspl-lpstatc is a program installed in place of lpstat to emulate it as far as is practicable, generating a report of outstanding print jobs from the print queue in a format similar to the original lpstat. This is installed when gspl-lpcover is installed.

cancel is a shell script which invokes gspl-qdel on its arguments in a similar fashion as cancel does with lp. This is installed when gspl-lpcover is installed.

If options are supplied to lpstat, a warning is given that these are not implemented. Cancel does not take any options.

7.5 Managing printers (command line)

7.5.1 gspl-plist

```
gspl-plist [ -options ] [ printer ... ]
```

gspl-plist generates a list of the printers, with their current status, on the **GNUspool** printer lists, including those on remote machines, on standard output. Each line of the output corresponds to a single printer, and by default the output is generally similar to the default format of the printers screen of the gspl-pq command.

Various options allow the user to control the output in various ways as described below.

The output can be restricted to one or more named printers by listing them on the command line.

The default output of gspl-plist looks like the following

```
ljet       lp    a4      idle
epson      tty15 invoice halted
blue:epson lp    invoice idle
```

Printer names for printers on remote machines are prefixed as shown by the machine name and a colon. The output is displayed in the minimum width to let each column line up as shown, with fields separated by spaces.

The output format and types of information displayed can be redefined using the `-F` option.

7.5.1.1 Options for gspl-plist

The environment variable on which options are supplied is `GSPL_PLIST` and the environment variable to specify the help file is `SPRESTCONF`.

Option		Args	Description
`-?`	`+explain`		causes a summary of the other options to be displayed without taking further action.
`-C`	`+classcode`	*classcode*	list only printers which match (have at least one bit in common with) the specified classcode. This is reduced (anded) with) to a subset of the user's default classcode unless the user has *override class* privilege).
`-D`	`+default-format`		revert to default format. This overrides any preceding `-F` option.
`-F`	`+format`	*string*	Specify user defined output format. The string argument contains characters to be printed and symbols representing the required data fields. This option overrides and preceding `-D` option, and is itself overridden by a following `-D` option.
`-H`	`+header`		generate a header for each column of the output according to the field displayed there. The headers are taken from the message file, by default `/usr/local/share/gnuspool/help/rest.help`, which may be adjusted as required.
`-l`	`+local-only`		include only printers local to the host on which the gspl-plist command is run.
`-N`	`+no-header`		cancel the `-H` option.

Option		Args	Description

`-q`	`+list-only`	*name*	list only printers whose name matches name. name may be a pattern with wild cards matched in a similar fashion to the shell, or a list of comma-separated alternatives.
			`-q 'a*'`
			displays printers whose names start with `a`
			`-q '[p-t]*,*[!h-m]'`
			displays printers whose names start with `p` to `t` or end with anything other than `h` to `m`. To cancel this argument, give a single `-` sign as name.

`-r`	`+remotes`		include printers on the local and connected hosts, cancelling `-l`
`-S`	`+sort-ptrs`		sort printers into order by host identifier and name.
`-U`	`+no-sort-ptrs`		display printers in the order encountered (default).
`+freeze-current`			Save all the current options in a `.gnuspool` file in the current directory.
`+freeze-home`			Save all the current options in a `.gnuspool` file in the user's home directory.

7.5.1.2 Format codes

The format string consists of a string containing the following character sequences, which are replaced by the following job parameters. The string may contain various other printing characters or spaces as required.

Each column is padded on the right to the length of the longest entry.

If a header is requested, the appropriate abbreviation is obtained from the message file and inserted.

`%%`	insert a single `%` sign
`%a`	display `AB!!` if printer has received an interrupt (abort print or interrupt print) message.
`%c`	display class code.
`%d`	display device name, with angle brackets for network devices.
`%e`	display comment field.
`%f`	form type
`%h`	display `heoj` if printer marked to halt at end of current job.
`%i`	process id of spd process.
`%j`	job number being printed (preceded by host name if not a local job).
`%l`	display `localonly` if printer is marked for local jobs only.
`%m`	feedback message on standard error from network filter process.
`%n`	`N/A` if printer is marked in need of running alignment routine.

`%p`	printer name.
`%s`	printer state, followed by a colon and the feedback message if applicable.
`%t`	printer state only, no feedback message.
`%u`	user name of owner of printing job.
`%y`	minimum job size if applicable.
`%z`	maximum job size if applicable.

Note that the various strings such as `localonly` etc are read from the message file also, so it is possible to modify them as required by the user.

The default format is

%p %d %f %s %j %u

7.5.2 gspl-start

```
gspl-start [ -options ] printer [ formtype ]
```

gspl-start may be used to start printers, or to halt, change the form type and to restart in one operation.

```
gspl-start lj4
```

will start printer `lj4` if it is halted. It will have no effect if it is already running. If it is printing, but it has been set to halt at the end of the current job, then this setting will be turned off, and the printer will continue at the end of the current job.

```
gspl-start lj4 a4
```

will likewise do nothing, other than cancelling a "halt at end of current job" setting, if printer `lj4` is running with the paper type `a4` (i.e. form type up to the first "`.`" or "-" is `a4`, ignoring the "suffix"). Otherwise, printer `lj4` will be halted at the end of the current job if it is running, gspl-start will wait for it to halt, change the form type to `a4` and then restart it.

Only "paper types", ignoring the suffix, are compared in deciding whether the form type is loaded.

Shell-style wild cards (which should probably be protected from incorrect expansion by shell using quotes) may be used, in conjunction with the `-f` option, to start multiple printers, thus:

```
gspl-start -f 'l*'
gspl-start -f '*[!a-m]'
gspl-start -f 'l*,*[!a-m]'
```

will respectively start all printers whose names begin with `l`, all printers whose names do not end with `a` through to `m` and both those classes of printers. Many installations include

```
gspl-start -f '*'
```

in the "startup script" to start up all local printers.

Most of the options (e.g. `-l` and `-C`) serve only to clarify which printer or printers are being referred to where there is ambiguity or the user wishes to restrict the field.

Names of printers starting with digits will not cause any confusion if spshed is running.

Printers on remote machines may be started by prefixing the machine name, thus

```
gspl-start -f host1:l*
```

Printers are not started on more than one machine at once. You should start the printers as separate commands:

```
gspl-start -f '*'
gspl-start -f 'host1:*'
gspl-start -f 'host2:*'
```

To run gspl-start, the user must have *halt and restart printers* permission.

7.5.2.1 Options

The environment variable on which options are supplied is `GSPL_START`. The environment variable to specify the help file is `SPRESTCONF`.

Option		Arg	Description
`-?`	`+explain`		causes a summary of the other options to be displayed without taking further action.
`-l`	`+device-name`	*device*	Specify by giving the device name or network address which of several similarly-named printers is intended.
`-C`	`+classcode`	*nnnn*	Specify via the class code which of several similarly-named printers is intended.
`-f`	`+force-all`		indicate that if the printer name argument selects more than one printer, then all such printers are intended.
`-n`	`+no-force`		cancels the `-f` option.
`-W`	`+wait-complete`		wait for operations to complete before exiting.
`-E`	`+no-wait`		do not wait for operations to complete before exiting.
`+freeze-current`			Save all the current options in a `.gnuspool` file in the current directory.
`+freeze-home`			Save all the current options in a `.gnuspool` file in the user's home directory.

7.5.3 gspl-phalt and gspl-pstop

```
gspl-phalt [-options] printer
gspl-pstop [-options] printer
```

gspl-phalt may be used for halting one or more printers at the end of the current job, or immediately if they are not printing. gspl-pstop may be used for halting one or more printers immediately.

```
gspl-phalt lj4
```

will halt printer `lj4` at the end of the current job if it is running and will otherwise have no effect.

```
gspl-phalt -f 'l*'
```

will likewise halt all printers whose names start with `l` etc.

Most of the options (e.g. `-l` and `-C`) serve only to clarify which printer or printers are being referred to where there is ambiguity or the user wishes to restrict the field.

Printers on remote machines may be halted by prefixing the machine name, thus

```
gspl-pstop -f host1:l*
```

To run these commands, the user must have *halt and restart printers* permission.

7.5.3.1 Options

The environment variable on which options are supplied is `GSPL_PHALT` for gspl_phalt and `GSPL_PSTOP` for gspl-pstop. The environment variable to specify the help file is `SPRESTCONF`.

Option		Arg	Description
`-?`	`+explain`		causes a summary of the other options to be displayed without taking further action.
`-l`	`+device-name`	*device*	Specify by giving the device name or network address which of several similarly-named printers is intended.
`-C`	`+classcode`	*nnnn*	Specify via the class code which of several similarly-named printers is intended.
`-f`	`+force-all`		indicate that if the printer name argument selects more than one printer, then all such printers are intended.
`-n`	`+no-force`		cancels the `-f` option.
`-W`	`+wait-complete`		wait for operations to complete before exiting.
`-E`	`+no-wait`		do not wait for operations to complete before exiting.
`+freeze-current`			Save all the current options in a `.gnuspool` file in the current directory.
`+freeze-home`			Save all the current options in a `.gnuspool` file in the user's home directory.

7.5.4 gspl-pchange

```
gspl-pchange [-options] printer [ formtype ]
```

gspl-pchange with appropriate options and a printer name (with optional form type) changes the specified parameters of a printer, which should be halted (and is left halted).

To run gspl-pchange, the user must have *add and delete printers* permission.

7.5.4.1 Options

The environment variable on which options are supplied is `GSPL_PCHANGE`. The environment variable to specify the help file is `SPRESTCONF`.

Option		Arg	Description
`-?`	`+explain`		causes a summary of the other options to be displayed without taking further action.
`-N`	`+network-device`		indicate to that the argument to -l refers to a network device.
`-L`	`+line-device`		indicate to that the argument to -l refers to a line (e.g. tty or lp-style) device.
`-s`	`+local-only`		mark printer as being local only to the host.
`-w`	`+network-wide`		mark printer as being available to other machines.
`-l`	`+device-name`	*device*	Specify by giving the device name or network address which of several similarly-named printers is intended.
`-C`	`+classcode`	*nnnn*	Specify via the (current) class code which of several similarly-named printers is intended.
`-D`	`+description`	*string*	set the description or comment field associated with the printer to string.
`-v`	`+new-device`	*device*	reset the device name as specified.
`-S`	`+set-classcode`		reset the class code as specified. Note that the `-C` option may help to select the desired printer.
`-f`	`+force-all`		indicate that if the printer name argument selects more than one printer, then all such printers are intended.
`-n`	`+no-force`		cancels the `-f` option.
`-W`	`+wait-complete`		wait for operations to complete before exiting.
`-E`	`+no-wait`		do not wait for operations to complete before exiting.
`+freeze-current`			Save all the current options in a `.gnuspool` file in the current directory.
`+freeze-home`			Save all the current options in a `.gnuspool` file in the user's home directory.

7.5.5 gspl-padd and gspl-pdel

```
gspl-padd [-options] printer formtype
gspl-pdel [-options] printer
```

gspl-padd may be used to add a printer and gspl-pdel may be used to delete a printer. Only one printer at a time may be added or deleted.

```
gspl-padd -l tty15 lj4b a4
```

adds a new printer with the specified parameters. The `-l` option is mandatory to denote the device name (or network address) as is the initial form type, in the above example `a4`.

```
gspl-padd -l 193.112.238.95 -N ptr9 a4
```

adds a network (terminal server) printer with the IP address given.

Please note that this does nothing to create an initial setup file, which should be created first.

```
gspl-pdel lj4b
```

Deletes the specified printer.

Printers on remote machines may not be added or deleted. This should be done from the relevant machine.

To run gspl-padd or gspl-pdel the user must have *add and delete printers* permission.

7.5.5.1 Options

The environment variable on which options are supplied is `GSPL_PADD` for gspl-padd and `GSPL_PDEL` for gspl-pdel. The environment variable to specify the help file is `SPRESTCONF`.

Option		Arg	Description
`-?`	`+explain`		causes a summary of the other options to be displayed without taking further action.
`-N`	`+network-device`		indicate to that the argument to `-l` refers to a network device.
`-L`	`+line-device`		indicate to that the argument to `-l` refers to a line (e.g. tty or lp-style) device.
`-s`	`+local-only`		(for gspl-padd) mark printer as being local only to the host.
`-w`	`+network-wide`		(for gspl-padd) mark printer as being available to other machines.
`-l`	`+device-name`	*`device`*	in the case of gspl-padd, defines the device name or network address of the printer. For gspl-pdel, it may be used to specify which of several similarly-named printers is intended.
`-C`	`+classcode`	*`nnnn`*	in the case of gspl-padd, it defines the class code (subject to modification by the user's class code) which will be given to the printer. For gspl-pdel it may be used to specify which of several similarly-named printers is intended.
`-D`	`+description`	*`string`*	set the description or comment field associated with the printer to *`string`*.
`-W`	`+wait-complete`		wait for operations to complete before exiting.
`-E`	`+no-wait`		do not wait for operations to complete before exiting.
`+freeze-current`			Save all the current options in a `.gnuspool` file in the current directory.
`+freeze-home`			Save all the current options in a `.gnuspool` file in the user's home directory.

7.5.6 gspl-pinter

```
gspl-pinter [ -options ] printer
```

gspl-pinter interrupts the job on the specified printer. This may be to halt printing of that job, allow a higher-priority job to print and then resume printing of the interrupted job at the interrupted page. Note that this will only work as intended if page delimiters are correctly defined.

```
gspl-pinter lj4
```

interrupts the given printer (or printers).

Printers on remote machines may be interrupted by prefixing the machine name, thus

```
gspl-pinter host1:lj4
```

To run this, the user must have *halt and restart printers* permission.

7.5.6.1 Options

The environment variable on which options are supplied is `GSPL_PINTER`. The environment variable to specify the help file is `SPRESTCONF`.

Option		Arg	Description
`-?`	`+explain`		causes a summary of the other options to be displayed without taking further action.
`-l`	`+device-name`	*device*	Specify by giving the device name or network address which of several similarly-named printers is intended.
`-C`	`+classcode`	*nnnn*	Specify via the class code which of several similarly-named printers is intended.
`-f`	`+force-all`		indicate that if the printer name argument selects more than one printer, then all such printers are intended.
`-n`	`+no-force`		cancels the `-f` option.
`-W`	`+wait-complete`		wait for operations to complete before exiting.
`-E`	`+no-wait`		do not wait for operations to complete before exiting.
`+freeze-current`			Save all the current options in a `.gnuspool` file in the current directory.
`+freeze-home`			Save all the current options in a `.gnuspool` file in the user's home directory.

7.5.7 gspl-ok and gspl-nok

```
gspl-ok [ -options ] printer
gspl-nok [ -options ] printer
```

If a printer is in "awaiting operator" state, then gspl-ok and gspl-nok may be used to communicate with it.

If it is in single sheet mode, then the printer will be continued identically whichever command is used.

If it is waiting for the approval of an alignment page, gspl-ok will grant approval and continue with printing, gspl-nok will disapprove the alignment page, causing the alignment routine to be rerun.

To run gspl-ok and gspl-nok, the user must have *select printer list* permission.

7.5.7.1 Options

The environment variable on which options are supplied is `GSPL_OK` for gspl-ok and

`GSPL_NOK_` for gspl-nok. The environment variable to specify the help file is `SPRESTCONF`.

Option		Arg	Description
`-?`	`+explain`		causes a summary of the other options to be displayed without taking further action.
`-l`	`+device-name`	*device*	Specify by giving the device name or network address which of several similarly-named printers is intended.
`-C`	`+classcode`	*nnnn*	Specify via the class code which of several similarly-named printers is intended.
`-f`	`+force-all`		indicate that if the printer name argument selects more than one printer, then all such printers are intended.
`-n`	`+no-force`		cancels the `-f` option.
`-W`	`+wait-complete`		wait for operations to complete before exiting.
`-E`	`+no-wait`		do not wait for operations to complete before exiting.
`+freeze-current`			Save all the current options in a `.gnuspool` file in the current directory.
`+freeze-home`			Save all the current options in a `.gnuspool` file in the user's home directory.

7.5.8 gspl-pstat

```
gspl-pstat [ -options ] printer state
```

gspl-pstat may be used in shell scripts to test the state of a printer.

```
gspl-pstat lj4
```

will return an exit code of 0 (TRUE to shells) if the printer is running, (i.e. printing or idle) and otherwise 1 (FALSE to shells). Some other exit code will be returned, together with a message, if there is an error, e.g. non-existent printer.

To test for specific states, use the state names as listed by gspl-plist, (in fact it takes the state names from the same place in the same file `/usr/local/share/gnuspool/help/rest.help` by default) for example.

```
if gspl-pstat lj4 error || gspl-pstat lj4 offline
then    echo trouble with lj4
fi
```

7.5.8.1 Standard state names for gspl-pstat

The following state names are used by gspl-pstat, which are usually shared with gspl-plist in the message file `/usr/local/share/gnuspool/help/rest.help` by default. As with other GNUspool programs, the message file may be edited as required:

`offline`	printer has gone off-line or timed out
`error`	an error has been raised by the printer, driver or GNUspool
`halted`	printer is shut down and printing suspended by GNUspool

`startup`	being initialised to a ready/printing state from the halted state
`idle`	ready to print next job but nothing to print
`shutdown`	in process of being taken down to the halted state
`printing`	a job is currently being printed
`a/w oper`	awaiting operator attention

7.5.8.2 Options

The environment variable on which options are supplied is `GSPL_PSTAT`. The environment variable to specify the help file is `SPRESTCONF`.

Note that only one printer name may be operated upon at any one time.

Option		Arg	Description
`-?`	`+explain`		causes a summary of the other options to be displayed without taking further action.
`-l`	`+device-name`	*device*	Specify by giving the device name or network address which of several similarly-named printers is intended.
`-C`	`+classcode`	*nnnn*	Specify via the class code which of several similarly-named printers is intended.
`+freeze-current`			Save all the current options in a `.gnuspool` file in the current directory.
`+freeze-home`			Save all the current options in a `.gnuspool` file in the user's home directory.

7.6 Interactive job and printer administration

7.6.1 Gspl-pq

```
gspl-pq [-options]
```

gspl-pq is a screen-oriented program which displays in real time the current state of the **GNUspool** spool queue and printer list, refreshing the screen automatically as the queue changes or printer state changes, and allowing the status of jobs on the queue to be altered according to each user's permissions and privileges.

Please see Chapter 7 for more details of the interactive commands. This section focuses on the command-line options which may be used to control the initial display.

7.6.1.1 Options

The environment variable on which options are supplied is `GSPL_PQ` and the environment variable to specify the help file is `SPQCONF`.

Certain commands available on-screen enable many of these options to be changed and saved in configuration files.

Option		**Arg**	**Description**
`-?`	`+explain`		causes a summary of the other options to be displayed without taking further action.

Option		Arg	Description
-A	+no-confirm-abort		do not require confirmation of any jobs being aborted or deleted.
-a	+confirm-abort		require confirmation of all jobs being aborted or deleted. If a previous -A option has been given, this will restore the default of only requiring confirmation for unprinted jobs. A further -a option in these circumstances will require confirmation in all cases.
-B	+no-help-box		put help messages in inverse video rather than in a box (default).
-b	+help-box		put help messages in a box rather than displaying inverse video.
-C	+classcode	*nnnn*	list only jobs and printers which match (have at least one bit in common with) the specified classcode.
-D	+printed-jobs		restrict the jobs display to jobs which have been printed.
-d	+dont-care		cancel -j or -p options; the cursor will be in the jobs section of the screen if there are any jobs to display and otherwise in the printers section, unless the user does not have access to the printer list.
-E	+page-counts		try to display page counts of jobs where applicable.
-e	+no-page-counts		do not try to display page counts in jobs, just the size in bytes.
-H	+keep-char-help		when displaying a help screen, interpret the next key press as a command as well as clearing the help screen.
-h	+lose-char-help		discard whatever key press is made to clear a help screen.
-I	+include-all		display all jobs regardless of printer (but still, if appropriate, restrict the display of printers).
-j	+jobs-screen		commence with cursor in jobs section of the screen.
-l	+local-only		display only jobs or printers local to the host on which gspl-pq is invoked.
-M	+no-error-box		put error messages in inverse video rather than in a box.
-m	+error-box		put error messages in a box rather than displaying inverse video.
-n	+follow-job		if a job or printer moves on the screen, try to follow it.
-P	+number-printers	*n*	set the number of lines on the printer section of the screen.
-p	+printer-screen		commence with cursor in printers section of the screen.

Option		Arg	Description

`-q`	`+list-only`	*name*	list only jobs whose destination includes, or printers whose name matches name.
			name may be a pattern with wild cards matched in a similar fashion to the shell.
			`-q 'a*'`
			displays jobs with printer names and printers whose names start with a.
			`-q '[p-t]*,*[!h-m]'`
			displays jobs and printer whose names start with `p` to `t` or end with anything other than `h` to `m`. To cancel this argument, give a single `-` sign as name.

`-R`	`+refresh-time`	*n*	set the screen refresh time. This only affects minor changes such as the current page count. Addition or deletion of jobs etc count as major changes and will cause an immediate screen refresh.
`-r`	`+remotes`		display jobs or printers on all attached hosts.
`-S`	`+sort-ptrs`		sort printers into order by host identifier and name.
`-s`	`+keep-cursor`		if a job or printer moves on the screen, try to preserve the relative position of the cursor on the screen rather than following the job or variable.
`-t`	`+just-title`	*string*	Restrict the display to jobs whose title matches the pattern specified. The pattern may be a pattern with shell-like wild cards as in the description of the `-q` parameter.
`-U`	`+no-sort-ptrs`		display printers in the order encountered (default).
`-u`	`+just-user`	*user*	Restrict the display to jobs owned by the user specified. The user name may be a pattern with shell-like wild cards as in the description of the `-q` parameter.
`-Y`	`+all-jobs`		cancel the display restrictions of `-y` or `-D`.
`-y`	`+unprinted-jobs`		restrict the jobs display to jobs which have not been printed.
`-Z`	`+no-include-null`		when listing jobs and printers restricted by a `-q` argument, do not include jobs without a destination printer.
`-z`	`+include-null`		when listing jobs and printers restricted by a `-q` argument, include jobs without a destination printer.

7.6.2 gspl-xmpq and gspl-xpq

gspl-xmpq is a fully interactive Motif alternative to the standard print queue manager gspl-pq. As with gspl-pq the format of the screen display, the help messages and the command keystrokes can be easily altered to suit your requirements.

Unlike gspl-pq there are no specific command line options to gspl-xmpq. The facility to change or specify resources settings for an X11 (and hence Motif) program on the command line can be used.

Xspq is a rewrite of gspl-xmpq using the GTK+ toolkit which unlike Motif is supported by GNU.

7.7 User administration and charging

7.7.1 gspl-ulist

```
gspl-ulist [-options]
```

gspl-ulist lists the permissions of users known to the **GNUspool** print spooler system.

The invoking user must have *edit admin file* permission to use gspl-ulist.

7.7.1.1 Options

The environment variable on which options are supplied is `GSPL_ULIST` and the environment variable to specify the help file is `SPRESTCONF`.

Option		Arg	Description
`-?`	`+explain`		causes a summary of the other options to be displayed without taking further action.
`-D`	`+default-format`		cancel the `-F` option and revert to the default format.
`-d`	`+default-line`		display an initial line giving the default options (default).
`-F`	`+format`	*string*	format the output according to the format string given.
`-H`	`+header`		generate a header for each column of the output appropriate for the format.
`-N`	`+no-header`		cancel the `-H` option.
`-n`	`+numeric-sort`		sort the list of users by the numeric user id (default).
`-S`	`+no-user-lines`		suppress the user lines. It is an error to invoke this and the `-s` option as well.
`-s`	`+no-default-line`		suppress the initial line giving the default options.
`-U`	`+user-lines`		display the user lines (default).
`-u`	`+user-sort`		sort the list of users by the user name.
`+freeze-current`			Save all the current options in a `.gnuspool` file in the current directory.
`+freeze-home`			Save all the current options in a `.gnuspool` file in the user's home directory.

7.7.1.2 Format argument

The format string consists of a string containing the following character sequences, which are replaced by various user permission parameters. The string may contain various other printing characters or spaces as required.

Each column is padded on the right to the length of the longest entry.

If a header is requested, the appropriate abbreviation is obtained from the message file and inserted.

%a	allowed form types
%b	allowed printer types
%c	allowed class code
%d	default priority
%f	default form type
%l	minimum priority
%m	maximum priority
%n	maximum copies
%o	default printer
%p	privileges
%u	user name
%v	L if the user is a valid user

The string `DEFAULT` replaces the user name in the default values line.

Note that the various strings are read from the message file, so it is possible to modify them as required by the user.

The default format is

```
%u %d %l %m %n %c %p
```

7.7.1.3 Privileges format

The following are output via the `%p` format. Note that the actual strings are read from the message file, and are the same ones as are used by gspl-uchange. `ALL` is displayed if all privileges are set.

`Adm`	edit admin file
`Stp`	stop scheduler
`Form`	select forms other than restriction pattern
`Otherp`	select printers other than restriction pattern
`Cpri`	change priority once queued
`Otherj`	edit other users' jobs.
`Prinq`	select printer list
`Hgo`	stop and start printers
`Anyp`	select any priority once queued
`Cdef`	change own default priority within range
`Addp`	add and delete printers
`Cover`	override class
`Unq`	unqueue jobs
`Votj`	view (but not change) other users' jobs
`Remj`	access remote jobs
`Remp`	access remote printers
`Accessj`	access non-displayed job attributes

`Freeze` freeze parameters from display

7.7.2 gspl-uchange

```
gspl-uchange [-options] [ users ]
```

gspl-uchange may be used to update the user permissions file giving the user profiles of various users and the operations which they may be permitted to perform within the **GNUspool** system, together with the default permissions which are assigned by default to new **GNUspool** users.

Further options allow for a "password dump" file to be maintained. This is for the benefit of NIS-type environments where reading through most of the password database can take an unacceptably long time, the user name and userid hash table is maintained in a file and updated as necessary.

The invoking user must have *edit admin file* permission.

7.7.2.1 Options

The environment variable on which options are supplied is `GSPL_UCHANGE` and the environment variable to specify the help file is `SPRESTCONF`.

Option		Arg	Description
`-?`	`+explain`		causes a summary of the other options to be displayed without taking further action.
`-A`	`+copy-defaults`		copy the default profile to all users before setting other permissions on the named users (with the `-u` option) or after setting the defaults (with the `-D` option). The privileges of the invoking user are not changed by this operation.
`-c`	`+class`	*nnnn*	set the class code of the user(s) as specified by the argument.
`-D`	`+set-defaults`		indicate that the other options are to apply to the default profile for new users.
`-d`	`+default-priority`	*num*	set the default job priority to *num*, which must be between 1 and 255.
`-F`	`+form-allowed`	*pattern*	set the permitted form types to match *pattern*.
`-f`	`+default-form`	*form*	set the default form type
`-l`	`+min-priority`	*num*	set the minimum job priority to *num*, which must be between 1 and 255.
`-m`	`+max-priority`	*num*	set the maximum job priority to *num*, which must be between 1 and 255.
`-N`	`+no-rebuild`		cancel the `-R` option.
`-n`	`+max-copies`	*num*	set the maximum number of copies to *num*, which must be between 1 and 255.
`-O`	`+ptr-allowed`	*pattern*	set the permitted printers to match *pattern*.
`-o`	`+default-ptr`	*printer*	set the default printer to *printer*.

Option		Arg	Description
-p	+privileges	*privs*	set the privileges of the user(s) as specified by the argument.
-R	+rebuild-file		rebuild the user permissions file `/usr/local/var/gnuspool/spufile23` incorporating any changes in the password list.
-s	+no-copy-defaults		cancel the effect of the `-A` option
-u	+set-users		indicate that the other options are to apply to the users specified on the rest of the command line.
-X	+dump-passwd		dump out the hash table of the password file to avoid re-reading the password file within the other programs.
-Y	+default-passwd		default handling of password hash file dump - rebuild if it is already present and `-R` specified, otherwise not.
-Z	+kill-dump-passwd		delete any existing dumped password hash file.
+freeze-current			Save all the current options in a `.gnuspool` file in the current directory.
+freeze-home			Save all the current options in a `.gnuspool` file in the user's home directory.

7.7.2.2 Users or default

In one operation gspl-uchange either adjusts the default permissions, to be applied to new users, if `-D` is specified, or specified users, if nothing or -u is specified. So first set the required defaults:

```
gspl-uchange -D -n 20 -p Form,Prinq,Hgo,Cdef -A
```

Then set named users

```
gspl-uchange -p ALL jmc root gnuspool
```

7.7.2.3 Rebuilding the user control file

After adding new users to the system, you should rebuild the user control file by running

```
gspl-uchange -R
```

On a system with a large number of users, this can take a long time, so the previous method of adding new users as they were encountered meant that various hold-ups occurred in standard utilities or the scheduler, whichever was the first to "notice" the changes, which might, in the event, be half-complete.

We suggest that this command be added to the "add new user" procedure for your installation.

7.7.2.4 Dumping the password file

When any of the **GNUspool** programs which may require to map numeric user ids to names and vice versa start, one of the first operations is to build the appropriate hash tables. This may take some time if there are a large number of user names, especially if NIS (a.k.a. yellow pages) is in use.

A short cut is to dump out the password file into a hash table file, by default `/usr/local/var/gnuspool/pwdump0`, which may be quickly read in by the relevant programs instead of rebuilding the hash table each time.

You may opt to create the dumped password file by running

```
gspl-uchange -X
```

This should only be done when all **GNUspool** programs are stopped.

Afterwards, each time the user control file is rebuilt using the `-R` option (or equivalents in other programs such as gspl-user), this file will also be rebuilt. `-X` does not have to be specified again.

If you ever decide you want to dispense with this file, run gspl-uchange with the `-Z` option.

For completeness, the `-Y` option is provided to cancel `-X` or `-Z` in case they are provided in the environment or a `.gnuspool` file, an extremely bad idea.

7.7.2.5 Privileges

The following may be specified as the argument to `-p`, as one or more (comma-separated) of the following codes, optionally preceded by a minus to turn off the corresponding privilege. These codes are the same as those displayed by gspl-ulist.

gspl-uchange disregards the case of the codes entered.

`Adm`	edit admin file
`Stp`	stop scheduler
`Form`	select forms other than restriction pattern
`Otherp`	select printers other than restriction pattern
`Cpri`	change priority once queued
`Otherj`	edit other users' jobs.
`Prinq`	select printer list
`Hgo`	stop and start printers
`Anyp`	select any priority once queued
`Cdef`	change own default priority within range
`Addp`	add and delete printers
`Cover`	override class
`Unq`	unqueue jobs
`Votj`	view (but not change) other users' jobs
`Remj`	access remote jobs
`Remp`	access remote printers
`Accessj`	access non-displayed job attributes
`Freeze`	freeze parameters from display

`ALL` may be used to denote all of the permissions. For example:

```
-p Otherj,Otherp
-p ALL,-Adm
```

Notice how `ALL` is set first and then `Adm` taken away in the second example.

A hexadecimal value is also accepted, but this is intended only for the benefit of the installation routines.

7.7.3 gspl-user

```
gspl-user  [-options ]
```

gspl-user provides 3 functions:

With no arguments it lists the permissions for the invoking user and exits.

With the `-c` option it enables the invoking user to edit his own default priority, form type and printer. The user must have *change default priority* permission to do this.

With the `-i` option it enables the invoking user to interactively review and update the list of permissions for all users. The user must have *edit admin file* permission to do this.

Please see Chapter 8 for more details of the interactive commands. This section focuses on the command-line options which may be used to control the initial display.

7.7.3.1 Options

The environment variable on which options are supplied is `GSPL_USER` and the environment variable to specify the help file is `SPUSERCONF`.

Certain commands available on-screen enable many of these options to be changed and saved in configuration files.

Option		**Description**
`-?`	`+explain`	causes a summary of the other options to be displayed without taking further action.
`-B`	`+no-help-box`	put help messages in inverse video rather than in a box.
`-b`	`+help-box`	put help messages in a box rather than displaying inverse video.
`-c`	`+form-and-prio`	interactively set default priority, form type and printer for the invoking user.
`-d`	`+display`	is the default. A list of permissions is output to the standard output.
`-H`	`+keep-char-help`	when displaying a help screen, interpret the next key press as a command as well as clearing the help screen.
`-h`	`+lose-char-help`	discard whatever key press is made to clear a help screen.
`-i`	`+administer`	view and update the list of users
`-M`	`+no-error-box`	put error messages in inverse video rather than in a box.
`-m`	`+error-box`	put error messages in a box rather than displaying inverse video.
`-n`	`+numeric-sort`	sort users by numeric userid (default).

Option		Description
`-u`	`+user-sort`	sort users by user name

7.7.4 gspl-xmuser and gspl-xuser

gspl-xmuser is a fully interactive Motif alternative to the standard user control program gspl-user, but only in the fully interactive mode, so it cannot be started by a user without *Edit admin file* privilege.

Unlike btuser there are no specific command line options to gspl-xmuser. The facility to change or specify resources settings for an X11 (and hence Motif) programs on the command line can be used.

The path name for the alternative message file is `XMSPUSERCONF`.

gspl-xuser is a rewrite of gspl-xmuser using the GTK+ toolkit, which unlike Motif is supported by GNU.

7.7.5 gspl-charge

```
gspl-charge [-options] [user] ...
```

gspl-charge displays and/or updates the charges recorded by **GNUspool** against various users specified on the command line, or all users if none are specified.

It is only available to users with the *Edit admin file* privilege.

Charges relate to jobs which have originated on the machine on which gspl-charge was invoked. The actual charging rate is determined by the machine on which the job is printed.

It is intended to deprecate charges in the next release of GNUspool.

7.7.5.1 Options

The environment variable on which options are supplied is `GSPL_CHARGE` and the environment variable to specify the help file is `SPRESTCONF`.

Option		Arg	Description
`-?`	`+explain`		causes a summary of the other options to be displayed without taking further action.
`-C`	`+consolidate`		causes a consolidation record to be written for each user mentioned in the file, which summarises the charges to date for each user. The records remain in the file, this creating a current balance of account record to save processing entries in the accounts file.
`-c`	`+add-charge`	*fee*	causes the specified fee to be added to the charge for the specified user or users, or all users if none are given. If combined with a print option (`-p` or `-P`), the print is done after the charge is applied.
`-K`	`+cancel-flags`		Cancels all previously-set options `-p`, `-P`, `-z`, `-C` and `-R`. Useful if you want to start again from some indeterminate state in a configuration file.

Option		Arg	Description
`-P`	`+print-full`		causes the full log of charges for the specified user or users, or all users if none are given, to be displayed on standard output.
`-p`	`+print-users`		(the default if no other options are specified) causes the charge summary for the specified user or users, or all users if none are given, to be displayed on standard output.
`-R`	`+reset-file`		causes all records to be deleted from the file except for consolidation records for each user with a non-zero charge. If the charge for each user is zero (as a result of applying the `-z` option perhaps), then the file will be left empty. Note that the spool scheduler spshed should not be running when this option is invoked.
`-z`	`+zero-charge`		clears to zero the charges for the specified user or users, or all users if none are given. This is done by appending an appropriate record or records to the charge file, the previous information (as displayed by `-P` will still be available. If combined with a print option (`-p` or `-P`), the print is done before the option is applied.
`+freeze-current`			Save all the current options in a `.gnuspool` file in the current directory.
`+freeze-home`			Save all the current options in a `.gnuspool` file in the user's home directory.

7.8 Web browser interface support

The following programs are parts of the web browser interfaces but are not documented here.

rspccgi	Operations on printers CGI program
rspcgi	List printers CGI program
rsqccgi	Operations on jobs CGI program
rsqcgi	List jobs CGI program
rsqcrcgi	Create jobs CGI program
rsqdcgi	Delete jobs CGI program
rsqvcgi	View jobs CGI program
spccgi	Operations on printers CGI program
spcgi	List printers CGI program
sqccgi	Operations on jobs CGI program
sqcgi	List jobs CGI program
sqcrcgi	Create jobs CGI program
sqdcgi	Delete jobs CGI program
sqvcgi	View jobs CGI program

7.9 Spool directory management

7.9.1 gspl-cjlist

```
gspl-cjlist [-D dir] [-e n  [-u] [-s] [-f] [-m] jobfile out workdir
```

gspl-cjlist converts **GNUspool** job files held in the spool directory to an executable shell script which may be used to recreate them. This may be useful for backup purposes or for one stage in upgrading from one release of **GNUspool** to another.

The jobs are copied into the backup directory `workdir`, and the generated shell script file `out` refers to files in that directory.

In addition to options, three arguments are always supplied to gspl-cjlist.

Job list file	This is the file containing the attributes of the jobs, `spshed_jfile` in the spool directory, by default `/usr/local/var/gnuspool`, or as relocated by re-specifying `SPOOLDIR`.
Output file	This file is created by gspl-cjlist to contain the executable shell script which may be used to create the jobs.
Backup directory	This directory is used to hold the job data.

Note that saved jobs make use of the `-U` option to gspl-pr to set the ownership correctly but do not attempt to use the `-E` and -o options to set external queue or originating hosts.

7.9.1.1 Options

Option	Description
`-D directory`	specifies the source directory for the jobs and job file rather than the current directory. It can be specified as `$SPOOLDIR` or `${SPOOLDIR-/usr/local/var/gnuspool}` etc and the environment and/or `/etc/gnuspool.conf` will be interrogated to interpolate the value of the environment variable given. If you use this, don't forget to put single quotes around it thus: `gspl-cjlist -D '${SPOOLDIR-/usr/local/var/gnuspool}' ...` otherwise the shell will try to interpret the `$` constructs and not gspl-cjlist.
`-e n`	Tolerate *n* errors of the kinds denoted by the other options before giving up trying to convert the file.
`-f`	Ignore format errors (invalid form types etc) in the saved job file up to the limit of errors given by the `-e` option.
`-m`	Create "missing" dummy jobs where the job file refers to jobs not in the saved directory.
`-s`	Ignore file size errors in the saved job file (up to the number of total errors given by the `-e` option).
`-u`	Do not reject the save job file if it contains invalid user names.

7.9.2 gspl-cplist

```
gspl-cplist [-D dir] [-e n] [-s] [-f] [-l] ptrfile outfile
```

gspl-cplist converts **GNUspool** printers held in the spool directory to an executable shell script which may be used to re-install them. This may be useful for backup purposes or for one stage in upgrade from one release of **GNUspool** to another.

gspl-cplist considers a **GNUspool** spool directory, by default the current directory, takes the printer file, which is usually `spshed_pfile`, and generates a shell script file *outfile*, which if executed, would recreate the **GNUspool** printers with the same options.

N.B. setup files are not copied, this should be done as a separate exercise. In many cases these can be separately copied over from one release to the next.

In addition to options, two arguments are always supplied to gspl-cplist.

Printer list file	This is the file containing the attributes of the printers, `spshed_pfile` in the spool directory, by default `/usr/local/var/gnuspool`, or as relocated by re-specifying `SPOOLDIR`.
Output file	This file is created by gspl-cplist to contain the executable shell script, containing gspl-padd commands, which may be used to reinstall the printers.

7.9.2.1 Options

Option	Description
`-D directory`	specifies the source directory for the printers and ptrfile. It can be specified as `$SPOOLDIR` or `${SPOOLDIR-/usr/local/var/gnuspool}` etc and the environment and/or `/etc/gnuspool.conf` will be interrogated to interpolate the value of the environment variable given. If you use this, don't forget to put single quotes around it thus: `gspl-cplist -D '${SPOOLDIR-/usr/local/var/gnuspool}' ...` otherwise the shell will try to interpret the `$` construct and not gspl-cplist.
`-e n`	Tolerate *n* errors of the kinds denoted by the other options before giving up trying to convert the file.
`-f`	Ignore format errors (invalid form types etc) in the saved printer file up to the limit of number of errors given by the `-e` option.
`-l`	Modify the shell script created so that printers recreated with be "local only" access rather than network wide. This may be relevant to pre-release 18 files.
`-s`	Ignore file size errors in the saved printer file (up to the number of total errors given by the `-e` option.

7.9.3 gspl-spuconv

```
gspl-spuconv [-D dir] [-e n] [-s] [-f] userfile outfile
```

gspl-spuconv considers a **GNUspool** spool directory, by default the current directory, takes the user file, which is usually `spufile` or `spufile0`, and generates a shell script file outfile, which if executed, would recreate the **GNUspool** users with the same options.

7.9.3.1 Options

Option	Description
`-D directory`	specifies the source directory for the users and user file. It can be specified as `$SPOOLDIR` or `${SPOOLDIR-/usr/local/var/gnuspool}` etc and the environment and/or `/etc/gnuspool.conf` will be interrogated to interpolate the value of the environment variable given. If you use this, don't forget to put single quotes around it thus: `gspl-spuconv -D '${SPOOLDIR-/usr/local/var/gnuspool}' ...` otherwise the shell will try to interpret the `$` construct and not gspl-spuconv.
`-e n`	Tolerate *n* errors of the kinds denoted by the other options before giving up trying to convert the file.
`-s`	Ignore file size errors in the saved user file (up to the number of total errors given by the `-e` option.
`-f`	Ignore format errors (invalid form types etc) in the saved user file.

7.9.4 gspl-netwcmd

```
gspl-netwcmd [-n] printer [form]
```

gspl-netwcmd can be used to extract the network (terminal server) printing command for a given printer and output it on standard output. This is often useful for testing terminal server commands from the command prompt in the event that error conditions arise and enquiring what a given command "would have been".

The current form type is used unless the form argument is given, but this generally should not make any difference.

A shell script is output to standard output containing the command. By default, all the `$SPOOLDEV` etc printer environment variables are expanded in the output where appropriate as the printer "would see it", however this may be suppressed by means of the `-n` option.

There is a very subtle difference if the network keyword is given with

```
network exec ....
```

rather than `network` without `exec` or via `network=`. In the first case, just the command is given, whereas in the other two cases the output is preceded with

```
#! /bin/sh
```

and a blank line.

Comments are added if no `reopen` statement appears. An error will be generated if the printer is directly connected.

7.9.5 gspl-ripc

```
gspl-ripc [-d] [-r] [-F] [-A] [-D secs] [-P psarg] [-G] [-n] [-o file]
[-S dir] [-x] [-B n] [-N char]
```

gspl-ripc traces, monitors problems with or deletes IPC facilities used by **GNUspool**. Many of the facilities are used for debugging, but it also serves as a quick method of deleting the IPC facilities, being easier to use than ipcs and ipcrm, in the event that the scheduler has crashed or been killed without deleting the IPC facilities.

To use this facility, just run gspl-ripc thus:

```
gspl-ripc -d >/dev/null
```

The diagnostic output may be useful as it reports any inconsistencies.

The monitoring option can be used to diagnose processes, possibly not **GNUspool** ones, which are interfering with **GNUspool** shared memory segments, in cases where a third-party application is suspected of damaging the shared memory.

gspl-ripc also checks for errors in memory-mapped files where the version of **GNUspool** is using those rather than shared memory.

7.9.5.1 Options

Option	Description
`-A`	Display details of jobs and printers. This often generates a lot of output and is not really necessary.
`-D secs`	Monitor which process has last attached to shared memory segments and report changes, polling every *secs* seconds.
`-d`	Delete the IPC facilities after printing out contents. This saves messing with arguments to ipcrm(1).
`-f`	Display the free chains for jobs and printers in shared memory. This generates a lot of output and isn't usually necessary.
`-n`	Suppress display from `-D` option if everything is OK.
`-o outfile`	Output to *outfile* rather than standard output. Set it to `/dev/null` if you don't want to see any output. The output is passed through fgrep(1) to find the line (if any) with the process id of the process which last attached to the shared memory.
`-G`	Used in conjunction with the `-P` option, the output from ps(1) is displayed in full, without passing it through fgrep(1).
`-P psarg`	Specify argument to ps(1) to invoke if corruption detected when monitoring with `-D` option.
`-r`	Read and display the entries on the message queue. This is normally suppressed because messages on the queue can't be "peeked at" or "unread".

Option	Description
`-S dir`	This is only relevant for versions of **GNUspool** which use memory-mapped files rather than shared memory. It specifies the location of the spool directory. If this is not specified, then the master configuration file `/etc/gnuspool.conf` is consulted to find the spool directory location, or failing that, the directory `/usr/local/var/gnuspool` is used.
`-x`	Dump the contents of shared memory or memory-mapped files in hexadecimal and ASCII characters.
`-B n`	Where n may be 1 to 8, specify the width of the hexadecimal dump output as a number of 32-bit words.
`-N <char>`	Replace the character in the ASCII part of the hexadecimal dump to represent non-ASCII characters with the specified character (the first character of the argument). The default is "`.`". To specify a space, you may need to use quotes thus: `-N ' '`

7.9.5.2 Example

To delete all IPC facilities after **GNUspool** has crashed.

```
gspl-ripc -d -o /dev/null
```

To monitor the job shared memory segment for errors, printing out the ps(1) output (where the full listing is obtained with `-ef`) search for the process id of the last process to attach to the segment. Print out the contents of the segment including in hexadecimal after corruption is detected.

```
gspl-ripc -D 30 -P -ef -o joblog -A -x
```

7.9.6 gspl-setspdir

```
gspl-setspdir [-s n] [-r] [-d] [-f form] [-y]
```

gspl-setspdir tidies up the **GNUspool** spool directory, removing old spool files, or alternatively re-queuing them with a given default form type, and eliminating jobs where the spool file is missing.

Optionally, it may be used to split the main spool directory into a series of subdirectories `000`, `001` etc, reducing the size of the main spool directory to manageable proportions and also speeding up file searches. Alternatively, the subdirectories may be linked to different file systems by the user.

For reasonably obvious reasons, gspl-setspdir should be run with the spooler shut down.

7.9.6.1 Options

`-d`	Clear (delete) any orphaned job files found in the directory or directories.
`-f form`	Requeue any orphaned job files with the form type *`form`* rather than deleting them.
`-r`	Instead of moving jobs between subdirectories, renumber the jobs instead. This should only apply when there were existing subdirectories, and the number is being increased.

`-d`	Clear (delete) any orphaned job files found in the directory or directories.
`-s n`	Split the spool directory into *n* subdirectories, or change the number of subdirectories to *n*, where *n* may be 0 (to convert back to a single spool directory), or a number up to 999. We recommend that n be a prime number, as this minimises the chance of a cycle of operations which is a factor of the number of subdirectories causing just a small number of subdirectories to be filled more or less exclusively. Confirmation will be requested if *n* is not prime. The subdirectories are given the names 000 to 999, and each job will be moved to a subdirectory given by the remainder when its job number (the number after the SP) is divided by the number of subdirectories.
`-y`	Answer "yes" to any questions which gspl-setspdir asks.

7.9.7 gspl-passwd

```
gspl-passwd [-u user] [-p password] [-f] [-d] [-F file]
```

gspl-passwd sets a password for the current user or a specified user if `-u` is given. This is separate and distinct from the user's login password. This password is used by the web interfaces, the Windows interfaces and the API.

The reason for doing this is because it is considered insecure to possibly repeatedly try login passwords from user programs.

If any users have a password set in this way, then all users will have to to have a password in the file to use any of the interfaces requiring a password.

Unlike the Unix passwd(1) routine, the old password is not prompted for and there is no confirmation.

7.9.7.1 Options

`-u user`	Set password for given user. This may only be for other than the current user if gspl-passwd is invoked by `root`.
`-p password`	Specify the password to be set other than prompting for it.
`-f`	It is normally considered an error to include a password for `root` for the same reasons that the password file is separate. However this option may be set to insist upon it.
`-d`	Delete the user's password from the file.
`-F file`	Use file to hold the password. The default if no file is given is `/usr/local/share/gspwfile` Any number of `-F` options may be given to set up several password files at once.

8 Text screen-based Programs

The following sections provide more information on the text screen-based programs gspl-pq and gspl-user, with attention to the screen commands. See the previous chapter regarding command-line options to gspl-pq and gspl-user.

8.1 gspl-pq - interactive spool queue manager

The following screens and screen-based commands apply to gspl-pq. Remember, however that all of the screen formats and the key bindings may be adjusted to suit your requirements, so the default commands listed here may have been changed on your system.

8.1.1 Position of cursor on entry

The cursor is placed in either the job section or the printer section on entry to gspl-pq.

If the user is not allowed access to the printer section, the cursor will always be placed in the job queue section on entry whatever options are specified.

Otherwise the job queue or printer list section may be selected using the `-j` or `-p` options respectively.

If neither option is specified, or `-d` is specified, the job queue will be selected if there are any jobs, otherwise the printer list will be selected.

8.1.2 Screen-based commands - cursor movement

Most of these commands apply everywhere, except when you are in the middle of updating a field such as the form type, where usually only the "help" and "screen refresh" commands apply.

We think that you may well want to assign function keys to some of these options, but you should have appropriate *terminfo* or *termcap* definitions to do this.

Command	**Action**
`j` or *down cursor* key	Move the cursor down one line, possibly scrolling down
`k` or *up cursor* key	Move the cursor up one line, possibly scrolling up
Ctrl-f	Move forward a screenful
Ctrl-b	Move backward a screenful
Ctrl-d	Move forward half a screenful
Ctrl-u	Move backward half a screenful
`B` or *home*	Go to top of screen; if already there scroll to top of list
`E`	Go to bottom of screen; if already there scroll to end of list
Ctrl-l	Redraw the screen
`?`	Display some sort of help message.

If the up and down cursor keys work on your terminal, these are equivalent to `k` and `j` and the latter should probably be removed from the valid keys for up and down movement, to minimise potentially ambiguous situations, see key definitions on how to do this.

The standard response to an invalid instruction is to flash the screen, if possible, otherwise to ring the bell. This will be accompanied by an error message, displayed, highlighted if possible. Typical invalid instructions include:

- an attempt to move off the end of the display
- an invalid control character
- an invalid operation
- a lack of privilege, such as attempting to abort someone else's job.

8.1.3 Field entry

When a command to change a field is typed the cursor is moved to the start of that field. The existing field entry is undisturbed until the first character is typed. Characters are then accepted up until a carriage return or line feed is typed.

Special control characters in this context are:

Command	**Action**
escape	Abort the edit, restore the previous value, and quit the command
kill char	Kill all characters typed so far and start again
erase char	Delete the last character typed
`?`	For numbers - display meaning and range, for strings display possibilities starting with the characters typed so far.
space or *tab*	For strings only - insert a likely possibility. Successive spaces will cycle through all possibilities. If characters have already been typed the possibilities are restricted to those starting with the characters already typed.

8.1.4 Set and save program options.

The program options with which gspl-pq was entered may be interactively displayed and modified, with immediate effect (except in 3 cases). To select this facility, type the `=` command. A screen like this should be displayed:

```
Setting program options for gspl-pq
Restrict to printer :
Restrict to user   :
Restrict to title  :
Limit jobs display: Jobs matching printer plus null
Display only     : All
Local jobs/printers only  : No
Sort printers    : Unsorted
Class code     : ABCDEFGHIJKLMNOPabcdefghijklmnop
Confirm abort/delete jobs   : Unprinted jobs
Help messages    : Inverse Video
Error messages    : Inverse Video
Clear Help message: Use next command
```

```
Page counts      : Yes
If job moves      : Follow job
Printer List size(on entry) :  3
Cursor on entry    : Don't care
Refresh Interval    : 10
```

After setting the user is given the option to save the options, which may be done by creating or editing a `.gnuspool` file in the current directory (i.e. the directory when gspl-pq was entered) or in the user's home directory.

The printer list size and sort printers options are only effective if the options are saved and gspl-pq re-entered. Likewise the "cursor on entry" setting only makes sense on entry.

To cycle through all cases, move the cursor to the relevant field and press space. For the two numeric-valued cases just type in the required number. To exit, either type `q`, or else move past the last field. You should receive the message

```
Save current parameters?
```

Type `y` to save the parameters. You should receive a further message asking whether you want to save in the current directory, and if not in the home directory.

8.1.4.1 Restricting the display by printer

It is possible to restrict the display to a set of printers, and the jobs designated for those printers. This includes jobs which, by having no printer name included, may come out on one of the set. The set of printers may contain just one name, a list of names or a list of patterns for matching printer names.

The set of printers is specified on the line

```
Restrict display to printer :
```

They may be given as a comma-separated list of alternatives, including the use of shell-style wild-cards. This is the same as with the -q option on the command line, except that there are no quotes round the string. For example:

```
Restrict display to printer : ljet
Restrict display to printer : lp1,ptr3,laser7
Restrict display to printer : ptr*,lp[1-9]
Restrict display to printer : [a-c]?9
```

The wild-card options are:

`*`	Matches anything
`?`	Matches one character
`[a-m]`	Matches one character in list or range
`[!n-z]`	Matches one character not in list or range

8.1.4.2 Restricting the display by user

The display may be restricted by username, in a similar fashion to restricting by printer name. The set of users may be a single name, a pattern, a comma separated list of names and patterns. The same shell style wild card options are available for pattern matching as are used for printers.

The set of users is specified on the line

```
Restrict display to user:
```

8.1.5 Job queue section.

There may be blank space at the top of the screen as jobs are deleted, but the cursor can not be moved into it. The size field is displayed in pages preceded by an `=` sign if the job appears to be on separate identifiable pages, otherwise by a `/` sign, in which case it is in bytes. The `-e` option to gspl-pq suppresses the page indication.

The following commands are available when the cursor is in the job list, in addition to the cursor movement and scrolling commands given above.

Commands generally apply to the job selected by having the cursor alongside it.

Key	Action	Key	Action
`A`	Delete the job aborting if being printed.	`C`	Copy job without deleting
`c`	Change the number of copies	`F`	Change post-processing flags, if displayed
`f`	Change the form type for the job	`H`	Change the hold time, if displayed
`h`	Change the job title	`I`	Display the job on the screen
`K`	Change class code, if displayed	`L`	Change "local only", if displayed
`M`	Change "mail attention", if displayed	`m`	Change "mail result", if displayed
`O`	Change other options for the job	`o`	Switch to the printer list
`p`	Change the priority of the job	`P`	Change the printer destination
`q`	Quit out of gspl-pq	`R`	Edit page range, if displayed
`r`	Set/unset "retain after printing", if displayed	`s`	Set/unset "no header", if displayed
`T`	Change the "delete if unprinted" time, if displayed	`t`	Change the "delete if printed" time, if displayed
`U`	Unqueue the job	`u`	Change "post to" user, if displayed
`W`	Change "write attention", if displayed	`w`	Change "write result", if displayed
`=`	Update/save program options	`:`	Change "halted at" page, if displayed
`#`	Change "odd/even pages", if displayed	`,`	Reassign job list display format
`X`	Inspect system error log file		
`^`	search forward for any string parameter	`\`	search backward for any string parameter

Some commands, such as `H`, are only operative if the job display format has been changed to display the relevant job attribute on the main screen. These attributes are usually only edited from the supplementary job attribute screen, displayed using the `O` command. However attributes considered important, such as form type or printer may be edited with a suitable prompt even if they have been excluded from the main job list.

Most commands are simply ignored if the job is being printed. The main exceptions are abort/delete, and change copies.

To edit attributes of jobs, it will generally have to be your job, unless you have the privilege to edit other users' jobs. There is a separate permission to allow you to view the contents of other users' jobs without being able to edit them.

Other commands require you to have specific permissions as well as or in addition to those; you will need permission to select the printer list, to change the priorities of jobs which have been queued, or to "unqueue" a job.

8.1.5.1 Abort job

The `A` command deletes the selected job, aborting printing if it is in progress.

Confirmation is usually requested if the job has not been completely printed at least once, otherwise the job is deleted silently. You can choose between three levels of confirmation:

1. You can turn off all confirmation requests by specifying `-A`.
2. You can accept the default behaviour (possibly restored after a `-A` option with a single `-a` option), which requires confirmation for unprinted jobs only.
3. You can insist that confirmation always be requested with `-a` or multiple `-a` options.

If the job is being printed, then the appropriate action for abort handling for the printer is taken; see Chapter 5; by default the message `***ABORTED***` will be appended to the output, unless the output is being passed through a filter, in which case the filter process will be sent a `SIGTERM` signal.

Remember that output may not stop immediately if the printers have their own internal buffers to be emptied.

If a job has been retained on the queue via the `-q` option to gspl-pr, or because this is standard for the paper type, this command may be used to remove it, or it may be deleted automatically when the job has reached the "delete if printed" timeout.

8.1.5.2 Change the number of copies

The `c` command moves the cursor to the copies field of the job. You may change the number of copies from 0 to the maximum which you are permitted. If printing is in progress, the number of copies to be printed includes the current one.

You may reprint a job retained on the queue by resetting the copies field.

8.1.5.3 Change the form type

The `f` command changes the form type. The cursor moves to the job's form type field.

If the form type you want is loaded on one of the printers you may press *space* until it appears in this field, and you can always edit the result by changing a suffix, for example. Alternatively you may type the initial few letters of the form type to restrict the choice to form types starting with those letters.

Typing a `?` will display a list of form types loaded on the printers, or you may type this after typing one or more characters to list just the form types whose names start with those characters.

If you may only use your default form type, then the allowable forms here may be restricted.

8.1.5.4 Priorities

There are two privileges which apply here; you may or may not be allowed to change the priority of a job on the queue, and if you are you may or may not be limited to your standard upper and lower priorities. If you cannot change the priority, you will get an error message.

When in this field, you can type a `?` to see the range of priorities you may set.

8.1.5.5 Change the printer name

The `P` command changes the printer name. You may not be permitted to do this is you are limited to use of your default printer type.

As described in the `-P` option to the gspl-pr command, this field may be a pattern giving a range of printers which may print the job.

You may cycle through printer names using spaces in the same way as you may cycle through the form types.

8.1.5.6 Unqueue job

The `U` command enables a job to be taken off the queue, perhaps for editing before being returned. To protect secure environments (e.g., where accounts software log files are printed) the user must be granted the unqueue privilege.

When a job is unqueued, two files are created. The first file is a "shell script" or "command" file containing an gspl-pr command which if run unedited, would resubmit the job with the same options as it had when taken off the queue. The second file is a copy of the job itself, which is then deleted.

When the user invokes this command, gspl-pq prompts in turn for the directory to write in, the command file, and the job file. The director to write in defaults to the current directory when gspl-pq was started.

Environment variables prefixed with `$` are expanded when the directory name is read. Also, certain directories may be specified by using the `~` character, thus:

Sequence	**Meaning**
`~`	Invoking user's home (`$HOME`) directory
`~` *`user`*	Home directory of specified *user*
`~+`	Current directory when gspl-pq started
`~-`	Value of environment variable `OLDPWD` if it exists on entry to gspl-pq

The directory being written to must be writeable by the user invoking gspl-pq and must have search (execute) permission for user `gnuspool`.

The `C` command is similar to the `U` command except that the job is not deleted.

8.1.5.7 Other options.

Many of the other options for a job may be viewed and changed using the `O` option. The other option screen looks like:

```
Job   number 6417 submitted by jmc on 13:10:35 Mon 10 Jan 1999

Title       : passwd
Suppress header   : No
Form type     : a4
Post to user    : jmc
Mail message    : No
Write message   : No
Mail attention message  : No
Write attention message : No
Class code        :  ABCDEFGHIJKLMNOPabcdefghijklmnop
P/proc flags       :
Retain after printing  : No
Start   page: 1     Last print halted at      :   5
End page         :  <end>
Skip odd pages      : No
Skip even pages     : No
Reverse skip       : No
Printed         : No
Delete time if not printed (hours) :   168
Delete time if printed (hours)   :    24
Hold job until           :  <N/A>
Local job only   : No
```

The `Last print halted at:` field is only displayed if the job had started printing but not run to completion.

Note that the order of entries on this screen, and also the starting one, may be separately modified by adjusting the message file.

These options may be changed in just the same way as the options in the main screen are changed. For "flag" type options you may type `Y` or `T`, `N` or `F` and `~` to set, unset or "toggle" the flag. Note that the "this job has been printed" flag may be set or unset, however it will be set every time the job is printed, and if the "retain after printing" flag is not set and the printer setup file does not specify retain the job will be deleted then.

You may exit a field leaving it unchanged by using the up or down arrow keys, or *Enter* provided that nothing has been typed. You may quit back to the main screen by typing `q` provided that a string is not expected; otherwise move to another field and type `q`. Alternatively map this function onto some function key.

(To get the full benefit of using this and other screens in a smooth fashion, you should make sure that your terminal and GNUspool are set up to use cursor and function keys, particularly for the "help" and "quit" functions. We'd like to distribute versions of the key definitions which include function keys, but terminals and *terminfo* descriptions seem to differ so widely in this area, often by not defining them at all).

You can set a "hold until" time by typing `S` on the field, or delete it by typing `U`. When setting the time you can move across the parts of the time and date using the left and right arrow keys, `h` or `l` respectively. To change the time, press `+` to "increment" the component which the cursor is on or `-` to "decrement" it. The digits of the time and the day of the month may be explicitly overtyped.

A separate help message may be displayed by typing `?` except where this may be legitimately part of a string. You can map the help function onto a function key (see chapter 11 on Configurability for more details) if you want to be able to display help here. The help message includes possible user names when editing the "Post to User" field. If one or more characters have already been typed, only user names starting with that string appear.

In the same way, a space typed in the "Post to User" field will suggest a possibility from the list of users, or from that subset starting with the character or characters already typed.

If the job is currently being printed, GNUspool will display its progress as it is being printed. Should the job disappear whilst you are in this screen, you will receive a message. Just type *Enter* to continue.

If the job was halted whilst being printed, the page at which it was halted will be displayed after the startpage. You may reset this page number also if it is displayed. When the job is printed again, the first copy will commence at the page it was halted at, if applicable, and otherwise at the start page. Second and subsequent copies will commence at the start page. In all cases the last page to be printed will be the end page.

8.1.5.8 View job file

The `I` command displays the job on the screen. The job title and form type are displayed at the top of the screen, together with a "ruler" for measurement of character positions.

Since the whole of the printed page is unlikely to fit on the screen, the file is viewed in "screenfulls", and the end of printed pages is indicated on screen by highlighted markers, like `End of page 1`. The page counts are based either on the specified delimiters given to gspl-pr, or if the job has been printed, and the page delimiters reassigned, then these new assignments. If no delimiters are given, then formfeed is taken as the delimiter.

The simplest commands here are *space* or *carriage return* to advance through the file, and `q` to quit back to the main screen. When you are looking at the start, end or halt page, an appropriate message will be displayed at the top right-hand corner of the screen. If there is text which has been truncated on the right to fit onto the screen, then such text is marked by a "`>`" in the rightmost column.

Facilities are available to:

- Set the start, end and halt pages for the job
- Scroll right and left to view the whole of the printed page on wide stationery

- Go to the beginning or end of the file
- Search forward or backward for a specified string.

The complete set of commands available is:

Command	Action
space	next page
ENTER	next page
`k` or *up cursor*	previous page
`j` or *down cursor*	next page
`^`	search forward for string
`\`	search backward for string
`l`	scroll left 30%
`r`	scroll right 30%
`<`	scroll to left margin
`>`	scroll to rightmost column in file containing text
`s`	set start page of job to current page
`e`	set end page of job to current page
`H`	set "halted at" page to current page
`q`	quit back to job list

The only commands requiring further explanation are the searching commands. These provide a limited ability to search forward and backward through the file for a specified string; they prompt for a string to be searched for; case is insensitive, and the only "wildcard" allowed is "`.`", which means "any character".

8.1.5.9 Inspect system error log file

The `X` command inspects the contents of the system error log file, `spshed_reps`, in the spool directory, by default `/usr/local/var/gnuspool`. It may be useful to examine this in case something crashes, especially if a printer halts with error status.

The display starts on the final page of the log, to show the most recent messages. It is similar to the view job display and uses the same commands to move round the file.

8.1.6 Reassign format of job list display

The initial layout of the job list display is read from the message file. The format of the job list display of the main screen may be adjusted and the result saved in a new message file.

To select the display format, type a "`,`" character. The display will change to something like:

```
          3    n  Number in queue
" "
<         6    N  Job number
" "
```

```
           6    u  User submitting job
```

The numbers specify field widths, the code letters give the desired field and the text gives an explanation. Material in quotation marks designate separator characters. Lines prefixed by a `<` character denote fields which will be left-shifted to overtype the previous field if a specific instance is too large to fit (e.g. job number prefixed by a machine name; this is inserted sacrificing the number in the queue). There is a similar `>` character which denotes "skip any fields to the right if too large", but this is mainly useful for printer "feedback" messages.

To adjust the formats, move to the relevant row and type:

Command	Meaning
`q`	Quit and prompt to save
`i`	Insert new field before current
`a`	Insert new field after current
`'`	Insert new separator before current
`"`	Insert new separator after current
`D`	Delete field or separator
`w`	Reset width
`c`	Reset format code
`<`	Toggle left overwrite field
`>`	Toggle right overwrite field
`S`	Reset separator text

When a separator is entered, there is a "minimum" of a single space which will be assumed if a null string is given.

The following format codes may be given:

Code	Meaning	Code	Meaning
`A`	Signify "write attention"	`a`	Signify "mail attention"
`C`	Class code	`c`	Number of copies
`F`	Post-processing flags	`f`	Form type
`H`	Hold time	`h`	Job title
`K`	Size in KB	`l`	Signify "local only"
`L`	Position reached in KB	`m`	Signify "mail result"
`N`	Job number	`n`	Number in queue
`O`	Signify odd/even pages	`o`	Display originating host
`P`	Printer	`p`	Priority
`Q`	Page reached	`q`	Signify "retain on queue"
`R`	Page range	`s`	Signify "suppress header"
`S`	Size in pages	`t`	Delete time if printed
`T`	Delete time if not printed	`u`	User submitting job
`U`	Post user	`w`	Signify "write completion"
`W`	Submission time		

Notes:

1. Job number includes the machine prefix if the job is on a remote host.
2. Display originating host includes Windows PCs and jobs submitted from remote hosts via the gspl-rpr command.

When you have finished, type `q` to return. You will be prompted as to whether you want to save the formats. If you do, you can opt to save in your home directory or the current directory. You would choose the former if you want to select this every time you use gspl-pq and the latter if you wanted it only from the given directory.

A "header" line to the job list section consisting of only a single letter `j` will be replaced by an appropriate header consisting of headings appropriate to the format selected.

8.1.7 Printer list section of gspl-pq

This section of the screen reflects the active printers. To access it the user must have the appropriate privilege. A further privilege enables access to the halt and restart commands `H`, `G`, ! and `h`, and yet a further privilege to `N` and `D`.

Each printer known to the spooler may be in one of 8 states:

Startup
: The printer has been started with `G`, or with gspl-start, and is being initialised with the `setup` string.

Idle
: The printer is ready to print jobs with the given form type.

Printing
: The printer is printing a job, whose number and user are shown.

Shutdown
: The printer is in the process of shutting down, i.e. it was idle or printing, but has been halted, and is now shutting down the printer. This state is transient, probably not being seen on lightly loaded machines and printers with simple drivers / setup files.

Halted
: No jobs are going to the printer. This is the initial state. You will need to change to this state to change the form type.

a/w oper
: In this state the printer is awaiting a confirmation from the operator before continuing, either because it is in single job mode, or because an alignment page has just been printed. The spooler requires approval before continuing. The marker N/A (not aligned) appears on the right if it requires alignment.

Offline
: The printer has been detected as being off-line. This state is actually a variation on the Halted state. A message may be displayed to the right of this column giving further details.

Error
: An unrecoverable error has been detected on the printer. A message may be displayed to the right of this column giving further details. Some causes of errors are:

  - Syntax errors in setup files
  - Inaccessible device special files

- I/O errors other than those caused by off-line detection.

For the `Error` state, a message is left in the file `spshed_reps` in the spool directory, by default `/usr/local/var/gnuspool`, describing the problem. You can inspect this with the `X` command.

The extra commands available in the printer section of the screen are, in addition to the cursor movement and scrolling commands described earlier:

Key	**Action**	**Key**	**Action**
`!`	Interrupt current job	`A`	Abort current job
`c`	Edit description field for printer	`C`	Reset the class code
`D`	Delete printer from list	`f`	Change form type
`G`	Start printer	`h`	Halt printer at end of current job
`L`	Impose lower limit on job size	`U`	Impose upper limit on job size
`V`	Edit device name for printer	`H`	Halt printer at once
`N`	Add a new printer	`n`	Reject alignment page
`o`	Switch to job queue	`P`	Change printer name
`q`	Quit out of gspl-pq	`X`	View system error log file
`r`	Restart current job	`y`	approve alignment page
`=`	Save program options	`,`	Reset printer list display format
`^`	Search forward by contents of a string field	`\`	Search backward by contents of a string field
`?`	display help message		

The ctrl-f, `j` etc commands also operate here. Often the user will set the size of this section of the screen to display all the printers by entering gspl-pq with the `-P` option.

8.1.7.1 Stop/start commands

Obviously the `G` command will only have any useful effect on a halted printer, and the `h` and `H` commands will only affect a running printer.

If an alignment file is specified for the given form type (see the Printer Definitions chapter 5 for more details) then the marker `N/A` will appear on the right. If there are one or more jobs to be selected, then the printer will output the contents of the alignment file and switch to `a/w oper` mode to enable the alignment to be approved (by typing `y`) or retried (by typing `n`). If the job has the mail or write attention flags set, then the owner will be sent an appropriate message. If there are no jobs to be selected, the printer will move to `idle` mode (unless there is an error). You can type a `y` if you are confident that the paper is correctly aligned; however you will be asked for confirmation.

An idle printer can have the requirement for alignment reinstated, if applicable, by typing `n`. The `N/A` marker will reappear, and the next job will invoke the `a/w oper` procedure.

The difference between `h` and `H`, is that `h` waits until the end of the currently printing job if any.

Note that the `H` and `h` commands do not always cause an immediate state change;

they initiate a state change which may take some time to complete. So, in order to avoid confusion, the `H` command will mark the printer's line with "`AB!!`" (for abort) so you know it really is going to stop.

Similarly, the `h` command will mark the printer's line "`(heoj)`" (for "halt at end of job") while it is waiting for the job to finish.

If there is a large `docend`, `sufend` and/or `halt` string specified in the setup file, there may be some wait before the printer status changes from `shutdown` to `halted`. However you may force the issue by pressing `H` again, if you wish. The filter process group will be killed with a `SIGTERM` signal.

Beware of the fact that printers may send X-OFF characters when put off-line, and the reason why the `docend` or other string is being delayed may be because the port is waiting for an X-ON. Whilst `H` should nevertheless clear it, it is much better to only take it off-line after it has halted.

8.1.7.2 Printer size limits

You may select lower and upper limits for sizes of jobs to be printed on a given printer by typing `L` and `U` respectively. The sizes are in bytes, and a value of zero indicates no limit in either case.

If a current lower limit of job size applies on a printer, then a `<` will appear after the state in the display. If an upper limit applies, then a `>` will appear.

Jobs will only be selected for printing if the size in bytes falls between the limits set. The size is that for the whole job, disregarding any page ranges or odd/even page selection.

8.1.7.3 Restarting halted jobs

It is possible to "interrupt" a job while it is printing, then re-start it later from the last page printed.

This capability is especially useful when a small "rush job" arrives while a big job is printing. It is possible to interrupt the big job, print the rush job, then re-start the big job.

The `H` command may halt a printer in mid-job. If this happens, GNUspool will note the page at which printing stopped. To find out at where it stopped, either use the `O` (other options) command in the job list, or look at the file with `I`. If the job was halted, the `O` screen will display a field "Last print halted at" followed by the page number at which it was halted. When viewing the job with I, the "halted at" page is noted by a highlighted message at the top of the screen.

If you re-start the printer, the job may continue from the "halted at" page. If, however, a higher priority job has appeared on the queue, this job will instead be printed, and the interrupted job will be printed when it again appears at the head of the queue.

To make interrupting jobs easier, the ! command exists; this is equivalent to H followed by `G` on the same printer. This enables you to "rush through" a job interrupting a long job. Be careful; GNUspool is not omniscient about interrupting escape sequences and graphics.

If you wish, you may update the "halted at" page while the job is halted, to re-start it at an earlier or later page.

If the job was assigned to a specific printer by the user, it will stay assigned to that printer. If, on the other hand, GNUspool had assigned it to a printer automatically, it will be un-assigned and will start to print on the next available printer, with the correct form type loaded, when it gets to the head of the queue.

Again, beware of large printer buffers; these can limit the usability of this otherwise very useful facility.

8.1.7.4 Abort/restart current job

The abort command `A` is similar to that in the job queue, but no confirmation is asked for unprinted jobs. Note, the next job will be selected - it is usually better to halt the printer.

The restart command `r` causes the current print job to be restarted from the beginning. The action may be controlled for each printer by setting the `restart` string or command for the printer, see Chapter 5.

Both of these commands are limited in their effectiveness if the printer has a large buffer.

8.1.7.5 Change printer name

The command `P` moves the cursor to the printer name field of the selected printer, which must be halted.

The printer name may then be changed. As with other string-valued options, you may use spaces to put up various possibilities, and a `?` to list them. These are all derived from directory names in the directory `/usr/local/share/gnuspool/ptrs`, or from a locally supplied help file.

8.1.7.6 Change form type

The command `f` moves the cursor to the form type field of the selected printer, which must be halted.

The form type may then be entered. Again, as with other string-valued options, you may use spaces to put up various possibilities, and a `?` to list them. The possibilities are derived from form types in the job queue starting with the first job, followed by the names of setup files in the directory for the current printer. Alternatively a local help file may be displayed.

If you want to avoid erroneously displaying or offering other files, such as record or alignment files at this point, these files must have names commencing with a "`.`" or a "`-`".

8.1.7.7 Delete printer

This command removes a printer from the list. It may only be executed on a halted printer.

8.1.7.8 Add a printer

The printer name, port, form type and description are accepted in turn.

Note that in this case the *escape* key aborts the whole operation, not just the field being edited as with other string-valued commands.

When putting the printer name in, as with the `P` command, spaces cycle through the names of directories in `/usr/local/share/gnuspool/ptrs`.

When entering the line or device name, space cycles through currently unused (by GNUspool) entries in directory `/dev` which start with `tty` or `lp` (but excluding `/dev/tty`), which have read-write access to user `gnuspool`.

When putting the form type in, spaces cycle through setup file names in the directory for the printer type just entered.

The class code assigned to the printer will be that obtained from the user's default class code and the `-C` option, if specified.

8.1.7.8.1 FIFOs as printers

When adding a printer, a "line" or device name starting with a `/` is not interpreted relative to the `/dev` directory. A FIFO or named pipe may be given instead of a character special device.

If the environment variable `FIFODIR` is defined when the scheduler spshed is started, then the contents will be prepended to line file names starting with `/`.

8.1.7.8.2 Terminal servers as printers

The line or device name is not used by GNUspool, but can be used in the printer setup files. Typically a port number or unique identifier for the printer would be used. This port number is enclosed in angle brackets `<>`. The trailing `>` may be omitted when entering this but is inserted when displayed.

Note that the "guess" command doesn't work here.

Note that if you don't put the `<>`s in when you should, or do when you shouldn't, the error will be detected when the printer is started and the state error will appear. However an appropriate message will be put in the system error message file `/usr/local/var/gnuspool/spshed_reps`, which you can display with the `X` command.

8.1.7.9 Reset class code

The `C` command causes a line to be displayed with the current class code. The user can then reset the code to a new value. However if the user does not have the override class privilege, this will be bitwise *and*ed with their default class. A class code of zero (possibly as a result of this *and* operation) is an error and will not be allowed.

8.1.7.10 Operator attention commands

The commands `y` and `n` may be entered when the printer is in `a/w oper` status, as a result of printing an alignment page, or at the start of a single sheet. You can distinguish the two cases by the fact that `N/A` appears at the right of the line to indicate that the paper is not aligned.

If printing an alignment page, `y` accepts the page, and continues with the first job, whilst `n` causes another alignment page to be printed. The `a/w oper` state will not be re-entered unless the printer is halted and restarted, which should be done if the paper is wrecked.

In single job mode, both `y` and `n` have the same effect.

These commands may also be used when the printer is idle to bypass the requirement for alignment after the printer has been started, or to reinstate it. In either case confirmation is requested, type `Y` or `N` (for yes or no this time) as appropriate.

8.1.8 Reassign format of printer list display

The format of the printer list on the main screen may be adjusted and the result saved to suit the user's taste.

The initial layout of the printer list display is read from the message file, and this option will, if desired, save the desired format in a new message file.

To select the display format, type a "`,`" character. The display will change to something like:

```
" "
          14   p  Printer name
" "
          10   d  Device (or network address)
" "
          16   f  Form type
```

The numbers refer to field widths, the code letters give the desired field and the following text gives an explanation. Material in quotation marks designate separator characters. Fields prefixed by a `<` character denote fields which will be left-shifted to overtype the previous field if a specific instance is too large to fit . There is a similar `>` character which denotes "skip any fields to the right if too large"; this is mainly useful for printer "feedback" messages.

To adjust the formats, move to the relevant row and type:

Command	Meaning
`q`	Quit and prompt to save
`I`	Insert new field before current
`a`	Insert new field after current
`'`	Insert new separator before current
`"`	Insert new separator after current
`D`	Delete field or separator
`w`	Reset width
`c`	Reset format code
`<`	Toggle left overwrite field
`>`	Toggle right overwrite field

Command	Meaning
`S`	Reset separator text

When a separator is entered, there is a "minimum" of a single space which will be assumed if a null string is given.

The following format codes may be given:

Code	Meaning
`a`	Signify "abort pending"
`c`	Class code
`d`	Device or network address
`e`	Description
`f`	Form type
`h`	Signify "halting at end of job"
`i`	Process id (only on current machine)
`j`	Job number being printed
`l`	Signify "local only"
`m`	"Feedback" message if off-line/error
`n`	Signify "non-aligned"
`p`	Printer name
`s`	State with *`:message`*
`t`	State without message
`u`	User owning job being printed
`w`	Combined abort / non-aligned message
`x`	Limit(s) set, display as `<` and/or `>`
`y`	Minimum job size
`z`	Maximum job size

When you have finished, type `q` to return. You will be prompted as to whether you want to save the formats. If you do, you can opt to save in your home directory or the current directory. You would choose the former if you want to select this every time you use gspl-pq and the latter if you wanted it only from the given directory.

A "header" line to the printer list section consisting of only a single letter `p` will be replaced by an appropriate header consisting of headings appropriate to the format selected.

8.2 gspl-user - user administration tool

The gspl-user command has 3 functions:

1. To enable any user to inquire about their permissions and charges.
2. To let a suitably privileged user change their own default form type or default priority within given upper and lower bounds.
3. To allow a fully privileged user to change system defaults and user details.

This program adjusts the user control file `spufile23` in the spool directory, by default `/usr/local/var/gnuspool`.

8.2.1 gspl-user enquiry

The output of this is on the standard output, and is based on the user id of the requesting user. For a non-privileged user, this might take the form:

```
Default form A4 form types a4
Maximum Copies 10
Priority: min 100 max 200 default 150 Current Charge: 220
Default printer `' printer types `'
Class code A-Pa-p
You may Use other printers
You may Access queue options
You may Save new default options
```

8.2.2 gspl-user -c

If permitted this facility enables the user to change their default priority, default printer type or default form type. The current values are displayed on the screen in the following format:

```
Parameters for user: jmc
Default Priority: 150
Default form type: standard
Default printer type:
Priority: min 100 max 200 sys default 150
```

The user may type:

Command	**Action**
ctrl-l	Redraw the screen
`?`	Display help
`q`	Quit
`=`	Save program options
`d`	Change the default priority
`f`	Change the default form type.
`p`	Change the default printer type

The default priority may only be changed within the limits of the minimum and maximum priorities.

8.2.3 gspl-user -i.

This option causes a full-screen display of all users and their privileges and limits to be shown. The display is in the order of user id (the numeric code used internally by Unix), unless the `-u` option is additionally given, whereupon the user names will be displayed in alphabetic order.

As with gspl-pq the display of help and error messages may be adjusted with the options `-b`, `-B`, `-m` and `-M`. Likewise the options may be saved in the current or home directories using the = command.

This mode of operation may only be entered by a user with the privilege to *Edit Admin. File*. This privilege may only be set by someone else with this privilege; when GNUspool is initially installed this privilege is granted to the users `root` and `gnuspool`. Beware of the fact that this can be used to deprive any or all users of the privilege, preventing re-entry to this mode of operation. However you will be able to carry on provided that you do not leave the program.

If you accidentally deprive all users of the "Edit Admin File" privilege, the only way to repair the situation is to stop the spooler (if no one can do that using gspl-stop, then by brute force), delete the file `spufile23`, in the spool directory, by default `/usr/local/var/gnuspool` and restart. However all user and default details will be reinitialised to the installation defaults.

The format of the screen displayed is as follows:

```
  User    def min max cps form            ptr                     S/Priv

  DEFAULT 150 100 200 50  a4

  root    150 100 200 50  a4                                      >DP
  daemon  150 100 200 50  a4
  bin     150 100 200 50  a4
  sys     150 100 200 50  a4
  adm     150 100 200 50  a4
  uucp    150 100 200 50  a4
  muucp   150 100 200 50  a4
  nuucp   150 100 200 50  a4
  news    150 100 200 50  a4
  gnuspool 150 100 200 50  a4                                      >DP
  talk    150 100 200 50  a4
  batch   150 100 200 50  a4
  sync    150 100 200 50  a4
  lp      150 100 200 50  a4
  listen  150 100 200 50  a4
  jmc     150 100 200 50  a4                                      >DP
  dave    150 100 200 50  a4                                      >DP
  susan   150 100 200 50  a4                                      N/S P
  guest   150 100 200 50  a4              matrix                  <DP
  accts   150 100 200 50  invoice
                             -- 3 more below --
      GNUspool gspl-user Copyright Xi Software Ltd 2000 (? for help)
```

After the user name is given the default, minimum and maximum priorities, the maximum number of copies, the current form type, printer type, the class code and an indication of the privileges. This is either blank to denote that the user has the default privileges, or has the marks "<DP", ">DP" or "!=DP" to indicate that the privileges are respectively less extensive, more extensive, or different (less extensive in some respects and more in others) than the default.

The first row displayed gives the system-wide defaults and subsequent rows are in order of increasing numeric user ID or user name depending on whether or not the `-u` option is in effect. You can page through the user list using the cursor movement and scroll commands as for gspl-pq and set new values using:

Command	Action
`?`	display help
`c`	display current charge for user
`d`	change default priority for user
`D`	change system default priority
`l`	change minimum priority for user
`L`	change default minimum priority
`u`	change maximum priority for user
`U`	change default maximum priority
`f`	change default form type for user
`F`	change system default form type
`o`	change default printer type for user
`O`	change system default printer type
`r`	change allowed form type (pattern) for user
`R`	change system default allowed form type (pattern) for user
`g`	change allowed printer type (pattern) for user
`G`	change system default allowed printer type (pattern)
`n`	change maximum number of copies
`N`	change default maximum copies
`A`	copy system default priorities and form type to every user
`a`	copy system default priorities and form type to this user
`p`	change privileges for user using a further screen
`P`	change default privileges using a further screen
`q`	quit out of gspl-user
`=`	save program options
`s`	change class code for user
`S`	change default class code

The **`DEFAULT`** row will always stay in place as you scroll through the various users.

Type `q` to exit from the program.

8.2.3.1 Resetting system-wide defaults

When setting up the system for the first time, you should set the default privileges (upper case commands) and then set the privileges of users who are to differ in any way from the default. Note the commands to copy what you set to all other existing users.

The defaults are important not only because it is easier to set them up initially, but so that any new users are automatically added with the correct default privileges. Such new users are noticed and the details silently added to the user control file as they are encountered.

8.2.3.2 Setting default privileges

The privileges are not copied using the `A` command, but are set up separately. Note that the S/Priv column marks users whose privileges are different in some way from the default set, possibly by being less privileged.

First type the `P` command to enter the default privileges screen, which will look like this:

```
Default privileges
Default is to allow users to Edit admin file: No
Default is to allow users to Override class: No
Default is to allow users to Stop spooler: No
Default is to allow users to Use other forms: No
Default is to allow users to User other printers: No
Default is to allow users to Change priority on Q: No
Default is to allow users to Edit other users' jobs: No
Default is to allow users to Select printer list: No
Default is to allow users to Halt/restart printers: No
Default is to allow users to Add/delete printers: No
Default is to allow users to Set any priority on Q: No
Default is to allow users to Change own default prio/form: No
Default is to allow users to Unqueue jobs: No
Default is to allow users to View other users' jobs: No
Default is to allow users to Edit remote jobs: No
Default is to allow users to Edit remote printers: No
Default is to allow users to Access queue options: Yes
Default is to allow users to Save new default options: Yes
```

You can type

Command	Action
`y`	to allow a privilege
`n`	to disallow a privilege
Enter	to leave one as it is
`k` or *up cursor*	to move up
`j` or *down cursor*	to move down
`B`	to move to the beginning of the list
`E`	to move to the end of the list
`q`	to quit back to the main screen
`?`	to get a help screen

Note that changing some privileges always changes other ones, for example there is no point in allowing a user to halt printers but not to select the printer list, so a Yes will pop up in the latter row if you allow the former.

Likewise if you can edit the admin. file (what you are doing now), you can do anything anyway, so a Yes will pop up everywhere if you set this.

When you quit or finish the last privilege, you will be prompted as to whether you want to copy the default privileges to all users other than yourself. Use this facility carefully. It is possible to remove essential permissions from everybody including write administration file privilege. If you do this, you will have to stop the scheduler, possibly by brute force, and then delete the user control file `/usr/local/var/gnuspool/spufile0`.

8.2.3.3 Setting default class

To set the default class code, type `S`. The cursor moves to the class code field in the DEFAULT row, at the position corresponding to the letter `A`. The corresponding letter appears for any class code bit that is set, otherwise a "`.`" is displayed.

The commands to set bits in the class code are as follows:

Command	**Function**
`+`	Turn on corresponding bit
`-`	Turn off corresponding bit
`!` or `~`	Toggle corresponding bit
cursor left	Move left
cursor right	Move right
Enter	Quit back to main screen
`A` to `P` and `a` to `p`	Move to bit and set it.
`&`	Clear all bits (but you must set some on before leaving)
`*`	Set all bits

When you have finished setting the default class, you will be prompted as to whether you want to copy to all users, however this time you will be included in the copy, as it is not as catastrophic as depriving yourself of the edit admin file privilege would be!

8.2.3.4 Setting individual user privileges

To set individual privileges, move the cursor to the relevant user, and use the p command. The screen shown is almost identical to that for the default privileges thus:

```
Privileges for user jmc

User jmc is allowed to Edit admin file: No
User jmc is allowed to Override class: No
User jmc is allowed to Stop spooler: No
User jmc is allowed to Use other forms: No
User jmc is allowed to Use other printers: Yes
User jmc is allowed to Change priority on Q: No
User jmc is allowed to Edit other users' jobs: No
User jmc is allowed to Select printer list: No
User jmc is allowed to Halt/restart printers: No
User jmc is allowed to Add/delete printers: No
User jmc is allowed to Set any priority on Q: No
User jmc is allowed to Change own default prio/form: No
User jmc is allowed to Unqueue jobs: No
User jmc is allowed to View other users' jobs: No
User jmc is allowed to Edit remote jobs: No
User jmc is allowed to Edit remote printers: No
User jmc is allowed to Access queue options: Yes
```

```
  User jmc is allowed to Save new default options: Yes
```

The commands to set or unset privileges are the same as for the default privileges, with the addition of the command:

`D` - set to default privileges

However on exit, there is no prompt to copy before returning to the main screen.

9 X Programs

Motif is not supported by GNU and these tools have been rewritten as gspl-xpq and gspl-xuser using the GTK+ toolkit.

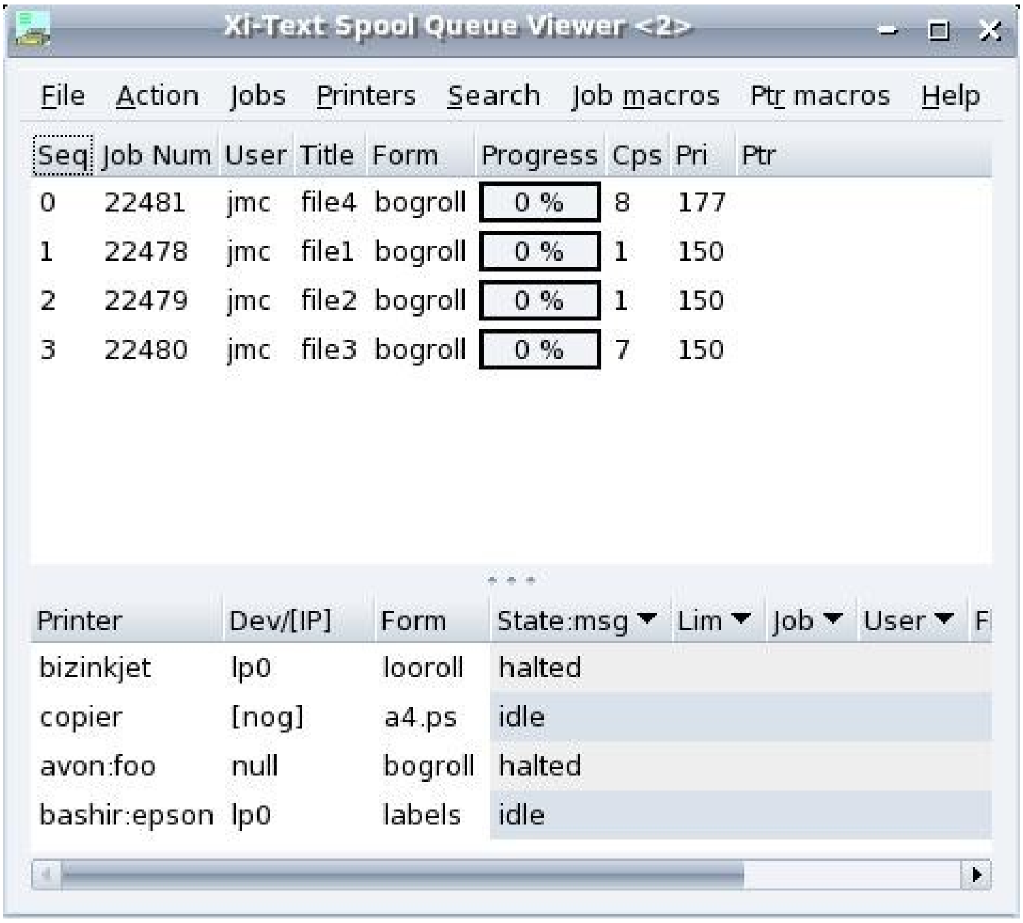
Much is broadly similar except macros are much easier to define and are held on a per-user basis.

9.1 Gspl-xpq – GTK+ Spool Queue Tool

Xspq is a fully interactive GTK+ alternative to the standard print queue manager, gspl-pq. As with gspl-pq the format of the screen display, the help messages etc can easily be altered to suit your requirements.

9.1.1 Gspl-xpq main screen

The main screen of gspl-xpq is as follows:



Seq	Job Num	User	Title	Form	Progress	Cps	Pri	Ptr
0	22481	jmc	file4	bogroll	0 %	8	177	
1	22478	jmc	file1	bogroll	0 %	1	150	
2	22479	jmc	file2	bogroll	0 %	1	150	
3	22480	jmc	file3	bogroll	0 %	7	150	

Printer	Dev/[IP]	Form	State:msg	Lim	Job	User	F
bizinkjet	lp0	looroll	halted				
copier	[nog]	a4.ps	idle				
avon:foo	null	bogroll	halted				
bashir:epson	lp0	labels	idle				

In the top part the jobs are displayed – if the job is actually being printed the printer name on which it is being printed is inserted, and the "progress" bar will slide over to 100%. The copies field will count down as well as each copy is completed. The progress bar shows that of the current copy.

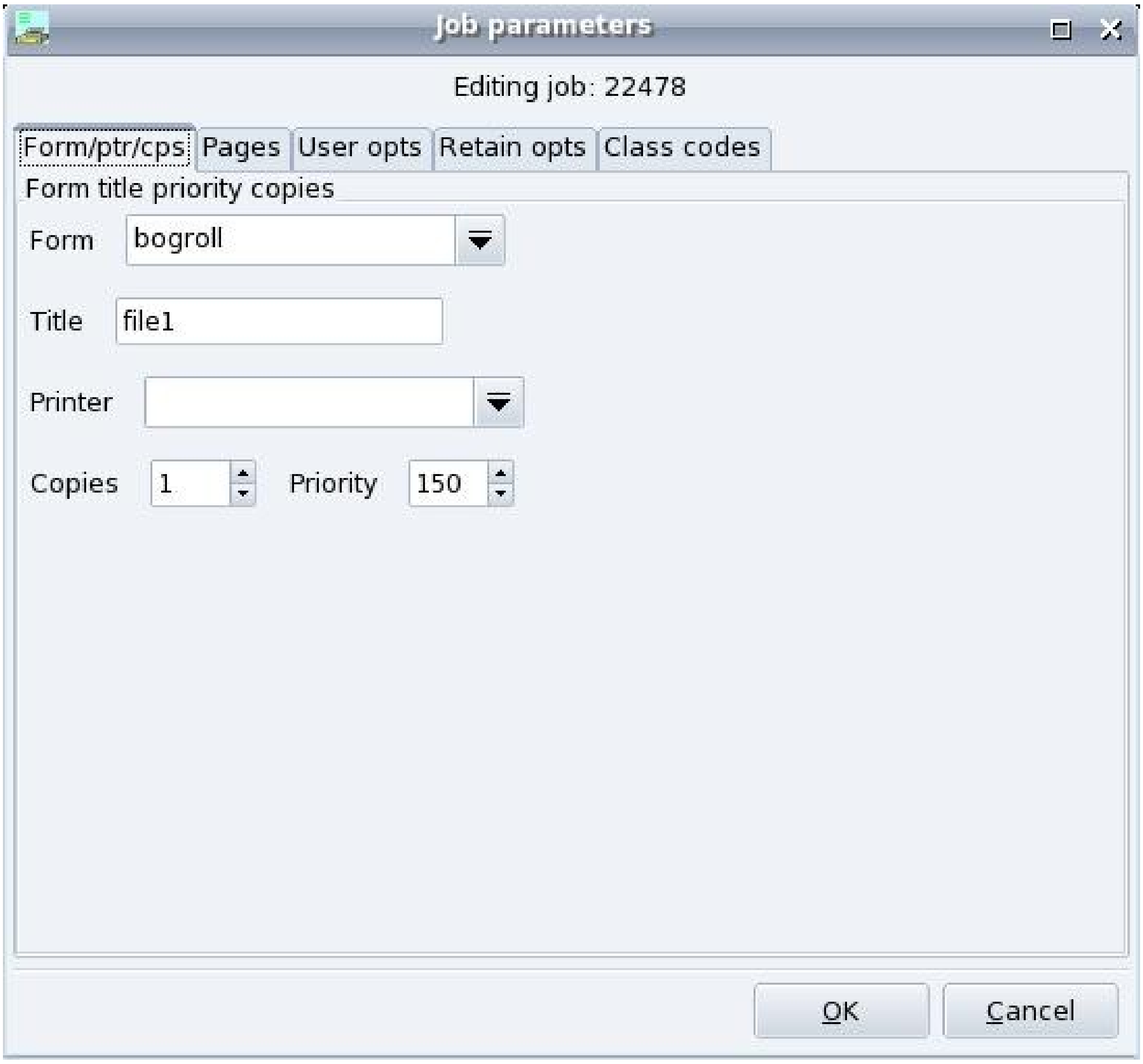
In most cases operations on jobs and printers are performed by clicking on the required job or printer and selecting the menu entry or keyboard accelerator.

Right-click popup menus are provided for common operations on jobs or printers.

9.1.2 Job operations

Most of the job menu selections bring up a “notebook” initialised to the selected page.

9.1.2.1 Form type dialog



Editing job: 22478

Form: bogroll

Title: file1

Printer:

Copies: 1

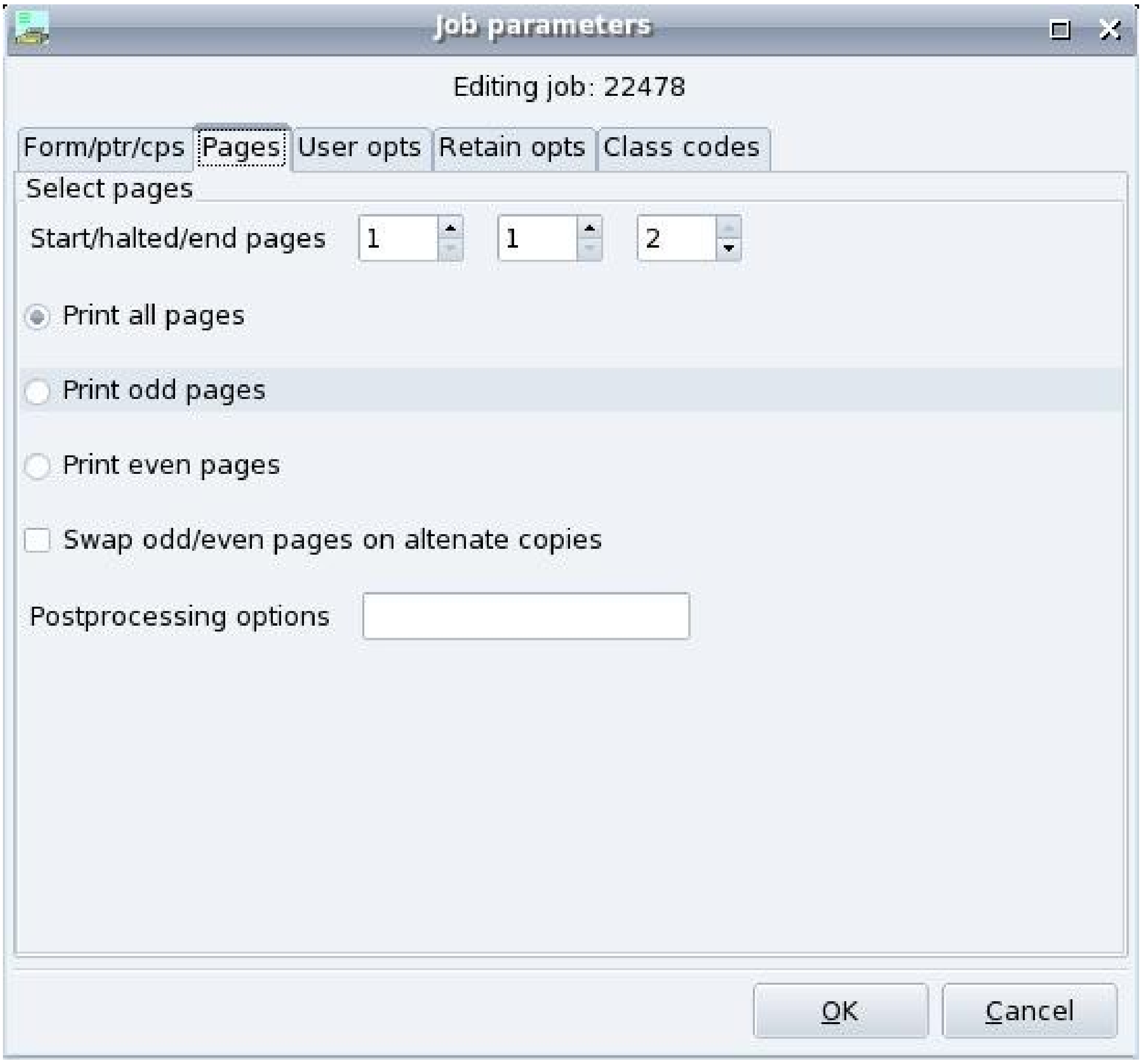
Priority: 150

The form type for the job can be reset using the “combo box” which includes form types on available printers, or the form type can be manually entered.

The printer likewise uses a “combo box” to select from available printers but don't forget that the printer name can use “globbing” such as `laser*` to indicate that any matching printer is acceptable.

The copies and priority fields are set using “spin boxes” within the range of acceptable values given in the user permissions.

9.1.2.2 Page dialog



Editing job: 22478

Start/halted/end pages 1 1 2

● Print all pages

○ Print odd pages

○ Print even pages

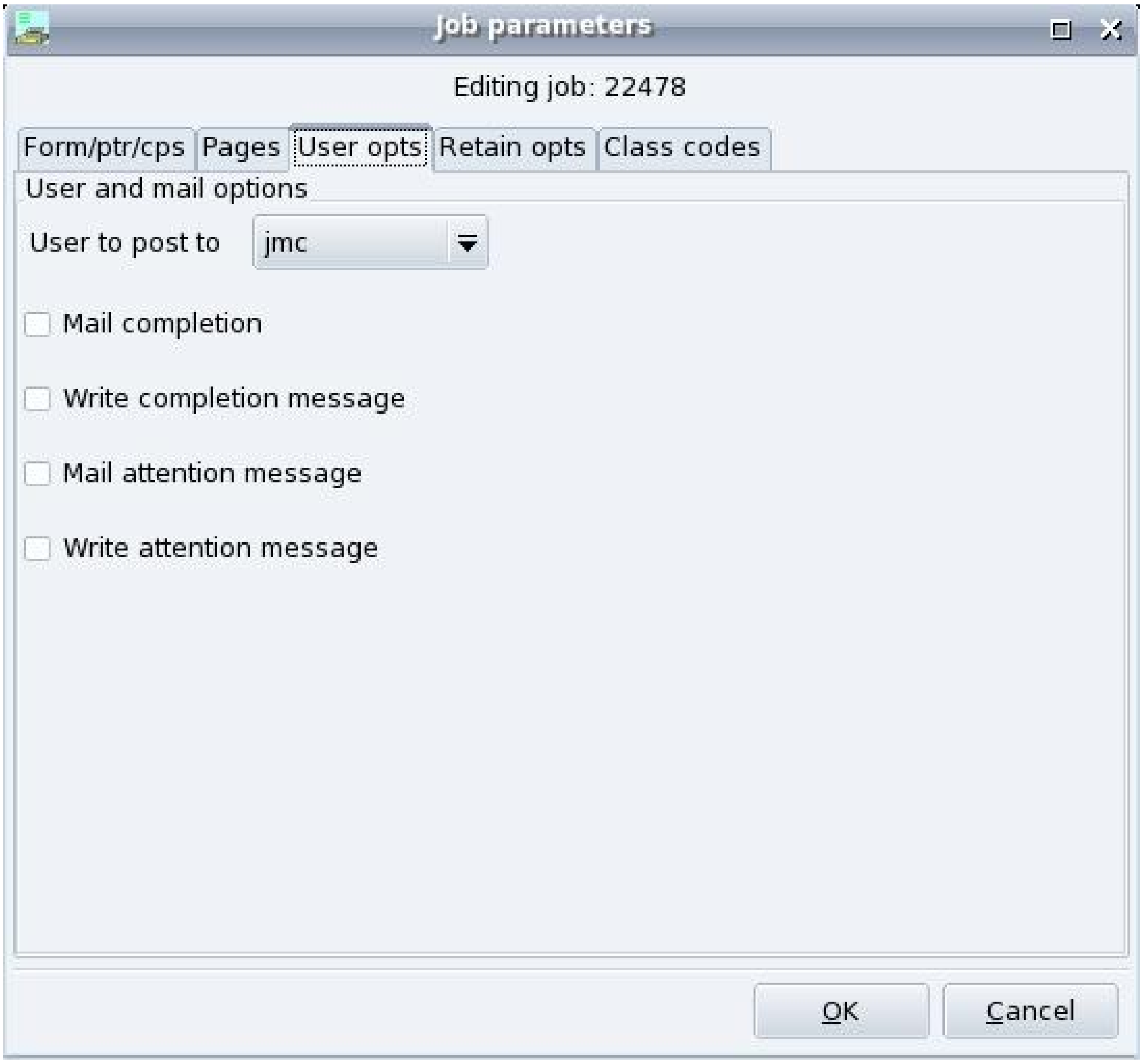
☐ Swap odd/even pages on altenate copies

Postprocessing options

The 3 spin boxes show the page range, including any possible “last print halted at” page if printing was incomplete on the last print made.

The final text box gives any option text to be passed to a postprocessing filter as `SPOOLFLAGS`.

9.1.2.3 User options



Editing job: 22478

User and mail options

User to post to: jmc

☐ Mail completion

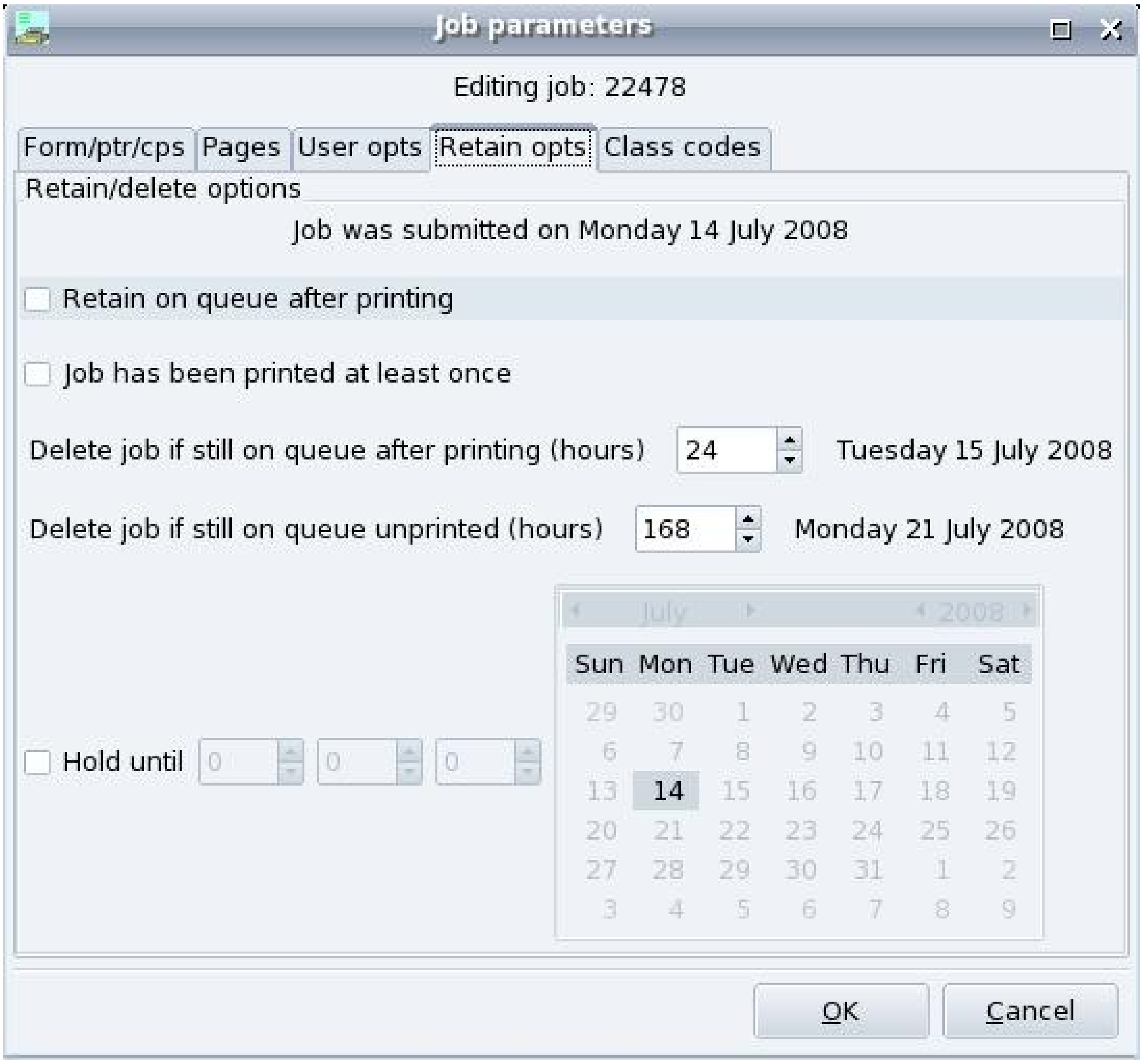
☐ Write completion message

☐ Mail attention message

☐ Write attention message

The combo box with user sets the “post to” user which appears on banner pages as opposed to the job owner.

9.1.2.4 Retain options



Editing job: 22478

Retain/delete options

Job was submitted on Monday 14 July 2008

☐ Retain on queue after printing

☐ Job has been printed at least once

Delete job if still on queue after printing (hours) 24 Tuesday 15 July 2008

Delete job if still on queue unprinted (hours) 168 Monday 21 July 2008

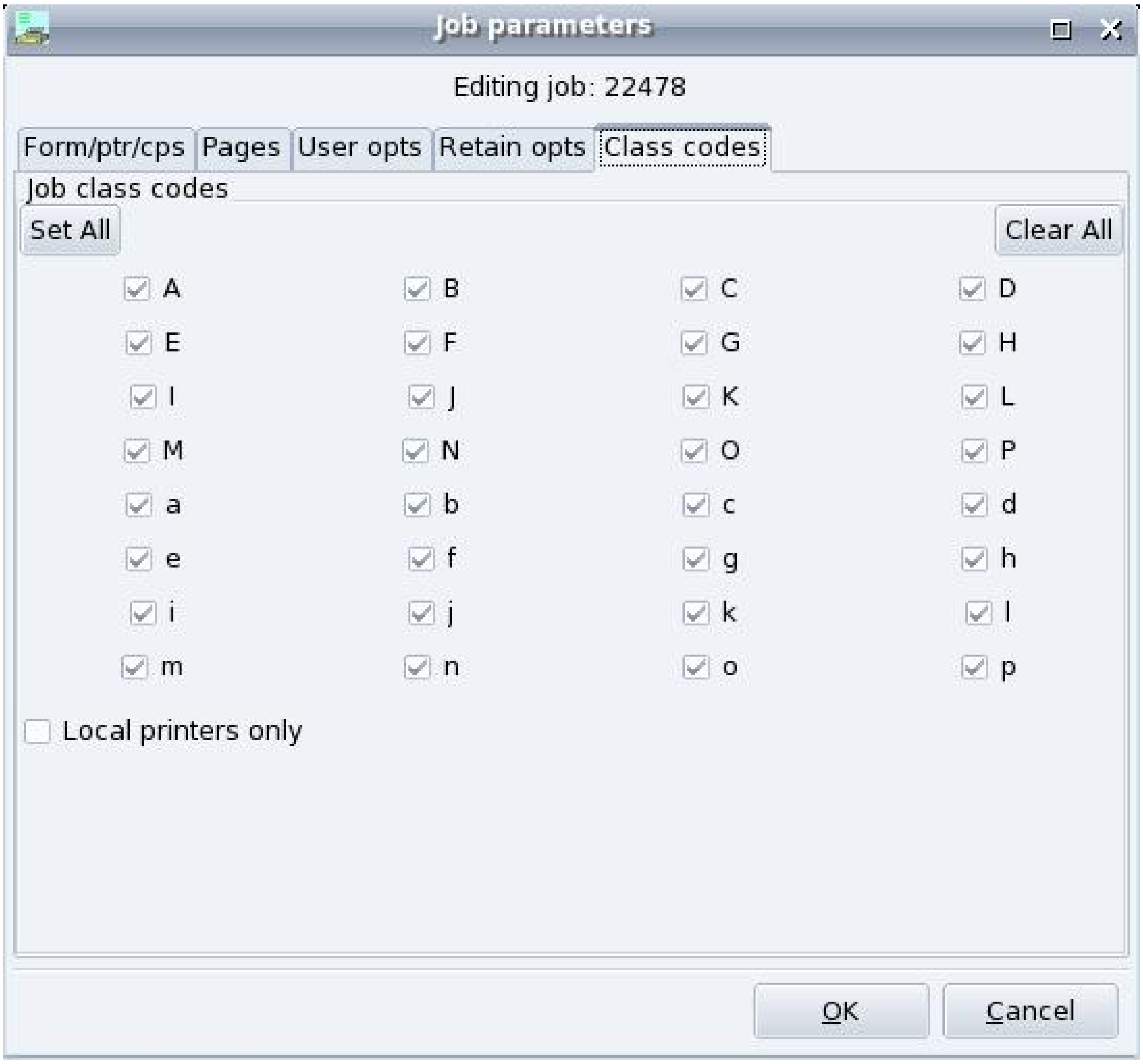
☐ Hold until 0 0 0

July 2008

Sun	Mon	Tue	Wed	Thu	Fri	Sat
29	30	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2
3	4	5	6	7	8	9

This dialog sets various operations for retaining the job on the queue and setting an earliest print job time and date.

9.1.2.5 Class codes



Editing job: 22478

Job class codes

☑ A ☑ B ☑ C ☑ D
☑ E ☑ F ☑ G ☑ H
☑ I ☑ J ☑ K ☑ L
☑ M ☑ N ☑ O ☑ P
☑ a ☑ b ☑ c ☑ d
☑ e ☑ f ☑ g ☑ h
☑ i ☑ j ☑ k ☑ l
☑ m ☑ n ☑ o ☑ p

☐ Local printers only

This screen enables the class codes bit to be set.

Note that the “set all” and “clear all” buttons have different effects if the user's default class code is not all ones and the user does or does not have “override class” privilege.

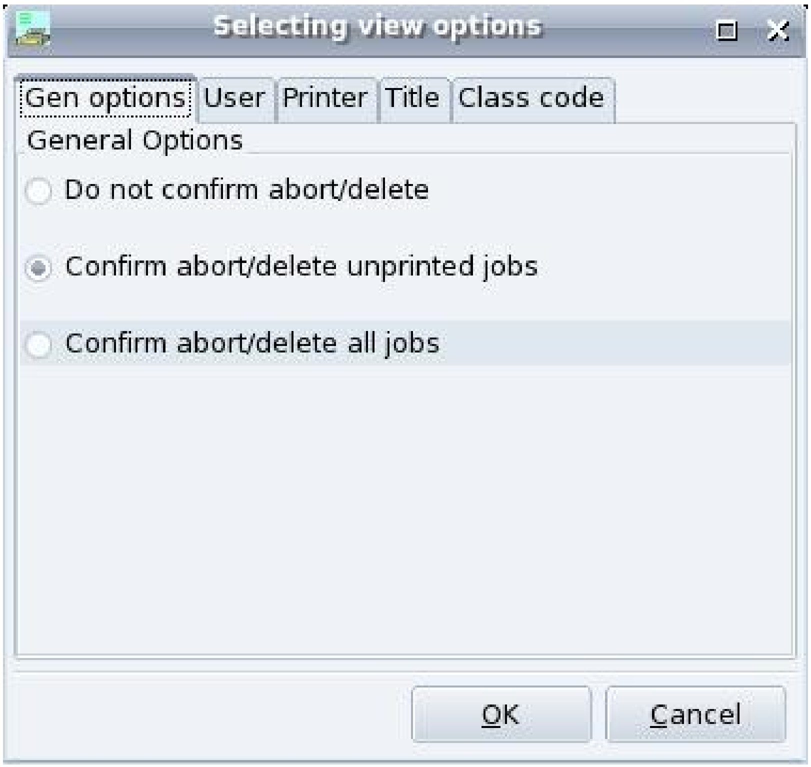
1. If the user does not have “override class” permission then the “set all” button will set to the default class and “clear all” will turn all bits off (you won't be able to leave it with all bits set off).
2. If the user has “override class” permission then “set all” will set to the default class if some of the default class bits are not set and otherwise to all ones. “Clear all” will set to the default class if bits not in the default class are set and otherwise to all zeroes.

“Local printers only” selects that the job will only be printed locally.

9.1.3 View options

Various options are available to limit the display to jobs and printers which the user is interested in, together with various other options.

These are available from the “File – View Options” menu entry. The options are likewise organised in “Notebook” form. The first page is:



General Options

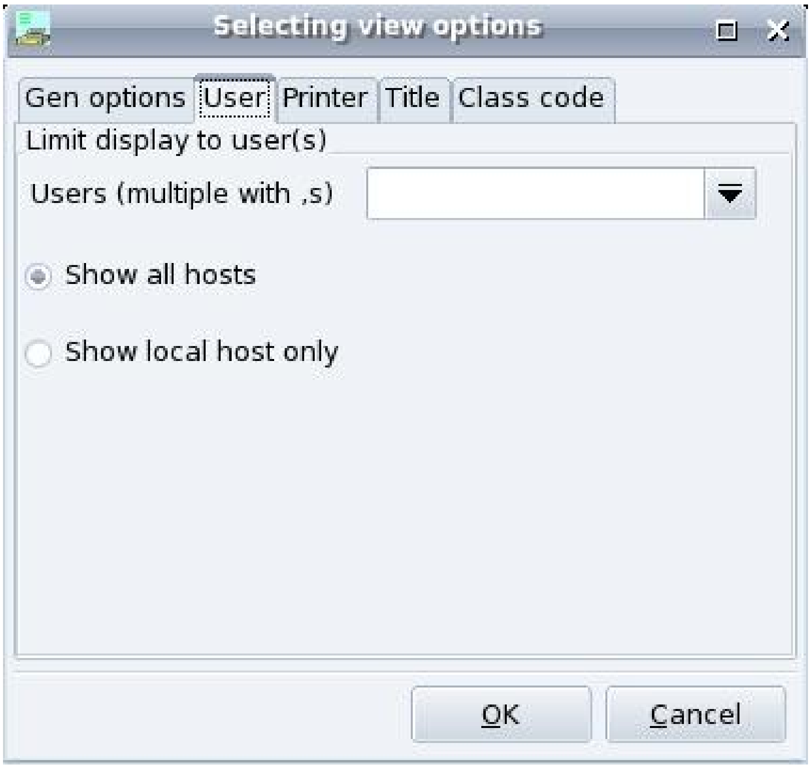
○ Do not confirm abort/delete

● Confirm abort/delete unprinted jobs

○ Confirm abort/delete all jobs

This determines whether confirmation is required for deletion of jobs, depending possibly on whether they have been printed.

The next page is:



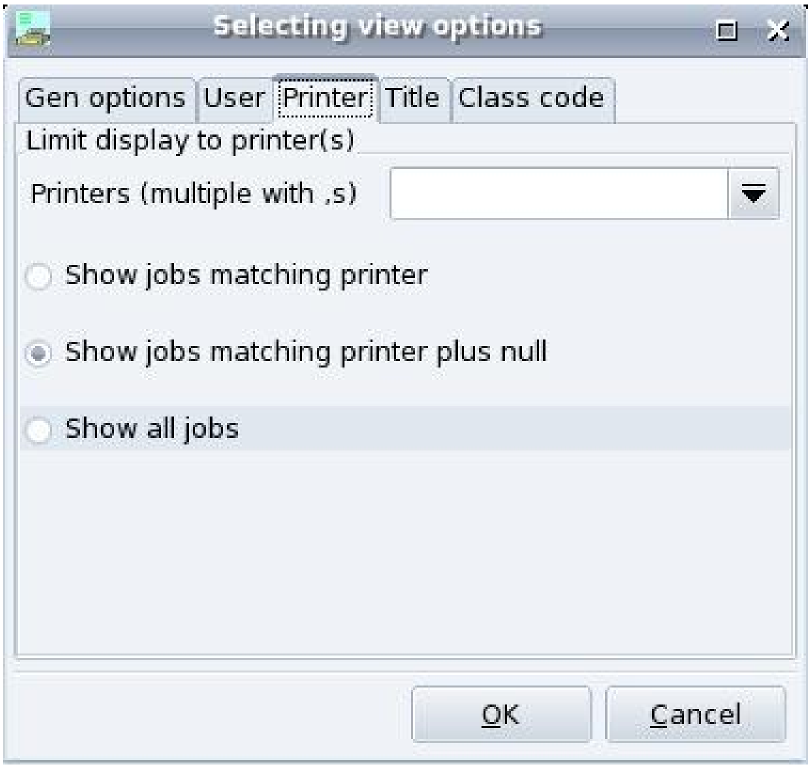
Limit display to user(s)

Users (multiple with ,s)

● Show all hosts

○ Show local host only

This enables the display to be limited to jobs for the given user or users. A further option selects whether just jobs and printers for the local host only are displayed.



Limit display to printer(s)

Printers (multiple with ,s)

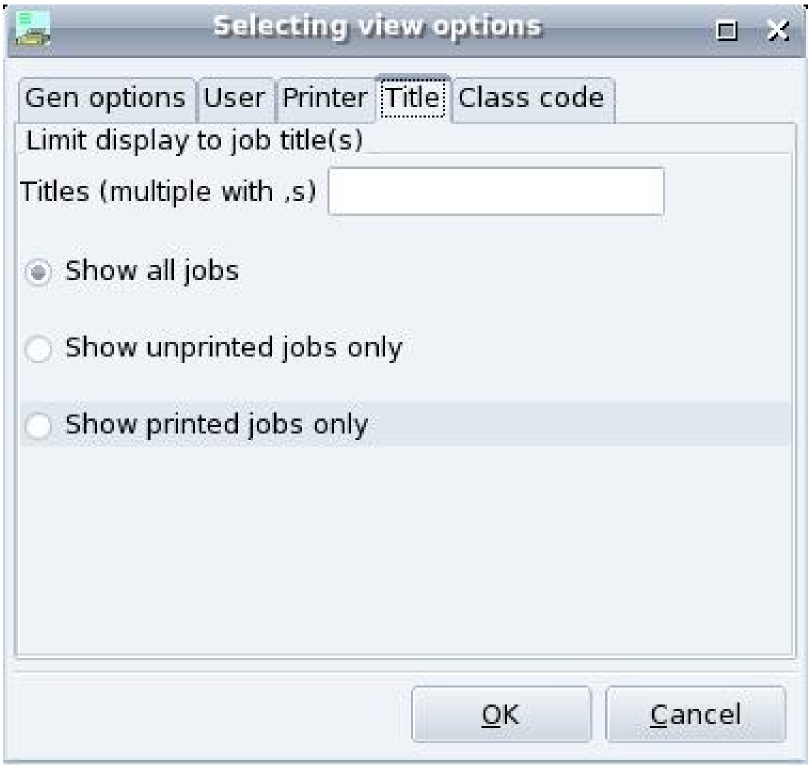
○ Show jobs matching printer

● Show jobs matching printer plus null

○ Show all jobs

This controls the printer display and whether jobs for a specified printer or printers only are displayed.

The radio buttons can be used to turn on or off the printer selection, or to allow the display of null printer field jobs, which in principle could be printed on any printer, to be displayed.



Limit display to job title(s)

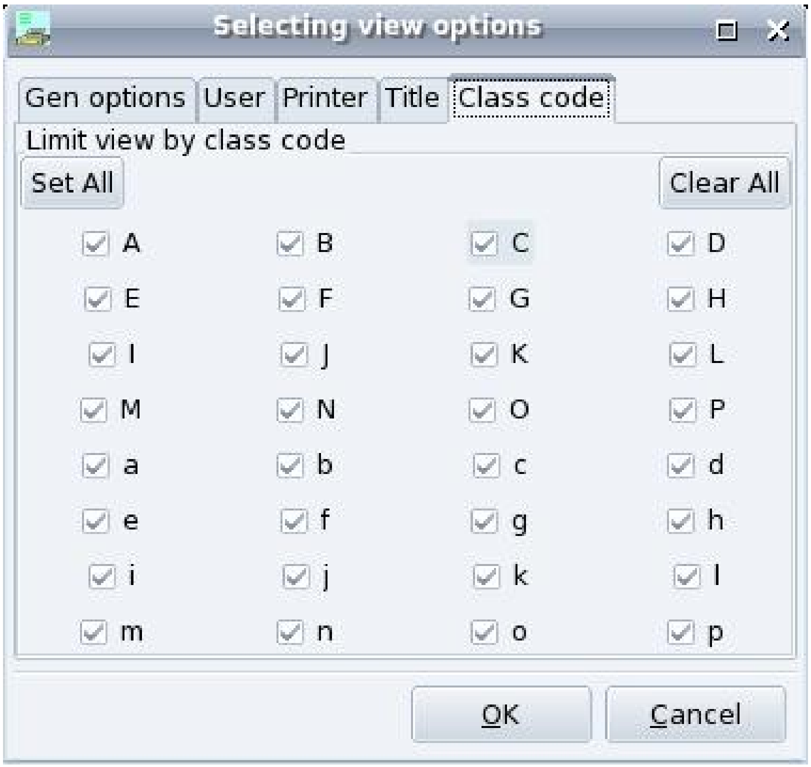
Titles (multiple with ,s)

● Show all jobs

○ Show unprinted jobs only

○ Show printed jobs only

This allows jobs to be selected by title, and also jobs which have or have not been printed to be selectively displayed.



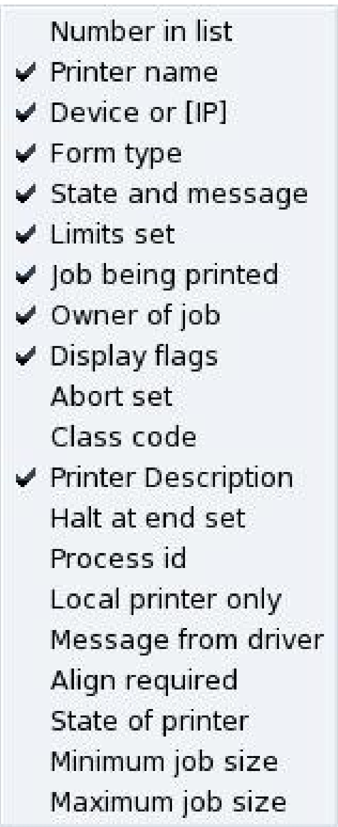
Limit view by class code

☑ A ☑ B ☑ C ☑ D
☑ E ☑ F ☑ G ☑ H
☑ I ☑ J ☑ K ☑ L
☑ M ☑ N ☑ O ☑ P
☑ a ☑ b ☑ c ☑ d
☑ e ☑ f ☑ g ☑ h
☑ i ☑ j ☑ k ☑ l
☑ m ☑ n ☑ o ☑ p

This allows job and printer view to be selected by class code.

9.1.4 Setting job and printer display attributes

To select different job and printer attributes to display, right click the appropriate header bar, to obtain a popup menu such as:



- Number in list
- ✓ Printer name
- ✓ Device or [IP]
- ✓ Form type
- ✓ State and message
- ✓ Limits set
- ✓ Job being printed
- ✓ Owner of job
- ✓ Display flags
- Abort set
- Class code
- ✓ Printer Description
- Halt at end set
- Process id
- Local printer only
- Message from driver
- Align required
- State of printer
- Minimum job size
- Maximum job size

Select or deselect the required fields to add or remove columns.

There is no option as such to rearrange fields, however newly added fields are added to the right, so to move fields just deselect them and reselect them in the required order.

The field selections are saved when the view options are saved.

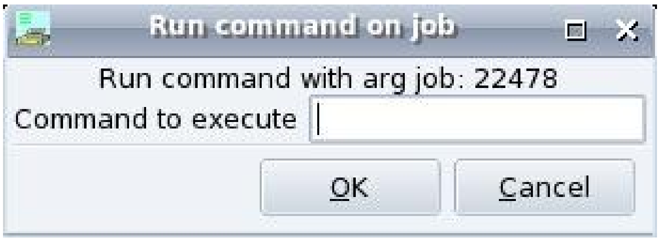
9.1.5 Macros

Gspl-xpq provides for “remembering” up to 10 commands each on jobs and printers where the selected job or printer is passed as a parameter to an arbitrary command.

You can run a command at any time by selecting “Job Macro – Run macro command” or similar for printer.

For example, suppose you wanted to have a “set copies to 8” macro.

First select this to get the display:



Run command on job

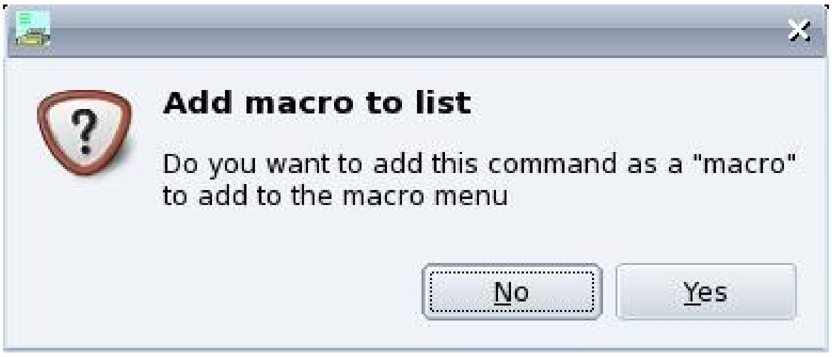
Run command with arg job: 22478

Command to execute:

Enter

```
gspl-qchange -c8
```

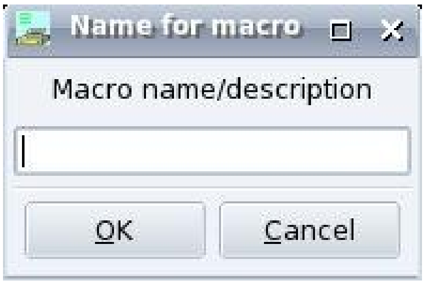
in the box and press OK. If it runs OK you will get



Add macro to list

Do you want to add this command as a "macro" to add to the macro menu

If you say yes you will get



Enter a suitable description, e.g. “set copies to 8” and this will be added to the macro menu, which “File – Save options” will remember for you.

Do not try to do anything too sophisticated especially things which require lots of quotes – if in doubt write a shell script and have the macro execute that.

Other items on the macro menu let you edit and delete existing macros.

9.2 gspl-xmpq - Motif GUI Spool Queue Tool

gspl-xmpq is a fully interactive Motif alternative to the standard print queue manager, gspl-pq. As with gspl-pq the format of the screen display, the help messages and even the command keystrokes can be easily altered to suit your requirements.

Unlike gspl-pq, there are no specific command line options to gspl-xmpq. The facility to change or specify resource settings for an X11 (and hence Motif) program on the command line can be used.

Remember that GNU does not support Motif – if possible use the GTK+ interface which is better.

9.2.1 Options

The option `-xrm "str"` may be used to specify a resource `str` to override the resource given in the resource file.

Do not worry about the terminology it is sufficient to recognise what these resources look like. For example the resource which specified the title for the main window of gspl-xmpq is:

```
gspl-xmpq.title: gspl-xmpq - GNUspool Spool Queue Viewer
```

The text before the colon "`:`" is the resource name (i.e. `gspl-xmpq.title`) and the text after is the data, in this case a string. The data can be boolean, integer or a string.

9.2.2 Useful Resources

All of the resources for program gspl-xmpq start with "`gspl-xmpq`" To keep the initial column width to manageable proportions just the text from the "`.`" or "`*`" is listed.

`.Title`	Specifies the title as it will appear on the main window of gspl-xmpq.
`.IconName`	Sets the text for the icon when the copy of gspl-xmpq is iconised.
`.localOnly:`	This is a boolean with the values True or False. When True the display will be restricted to show only local printers and jobs. When False printers and jobs on all co-operating GNUspool hosts will be shown.
`.unPrintedOnly:`	Selects whether to show all print jobs, just those which have been printed or those awaiting printing. The selection is indicated by: 0 All jobs 1 Unprinted jobs 2 Printed Jobs
`.inclJobs:`	Selects which print jobs to show when the view is being restricted by printer: The options are: 0 Show only jobs which match the printer restriction, excluding those which do not have a specified printer. 1 Show only jobs which match the printer restriction, including those which do not have a specified printer. 2 Show all jobs disregarding any printer restriction

`.onlyPrinter:`	Limit display to the specified set of printers, and the print jobs associated with them. This includes jobs which, by having no printer name included, may come out on one of the set.
	The set of printers may be one name, a list of names or a list of patterns for matching printer names. They may be given as a comma-separated list of alternatives, including the use of shell-style wild-cards. For example
	`Ljet`
	`Lp1,ptr3,laser7`
	`Ptr*,lp[1-9]`
	The wild-card options are:
	`*` Matches anything
	`?` Matches one character
	`[a-m]` Matches one character in list or range
	`[!n-z]` Matches one char. not in list or range

`*jlist.visibleItemCount:`	An integer specifying the number of lines in the job list pane of the Main Window.
`*plist.visibleItemCount:`	An integer specifying number of lines in the printer list pane of the Main Window.

9.2.3 Examples

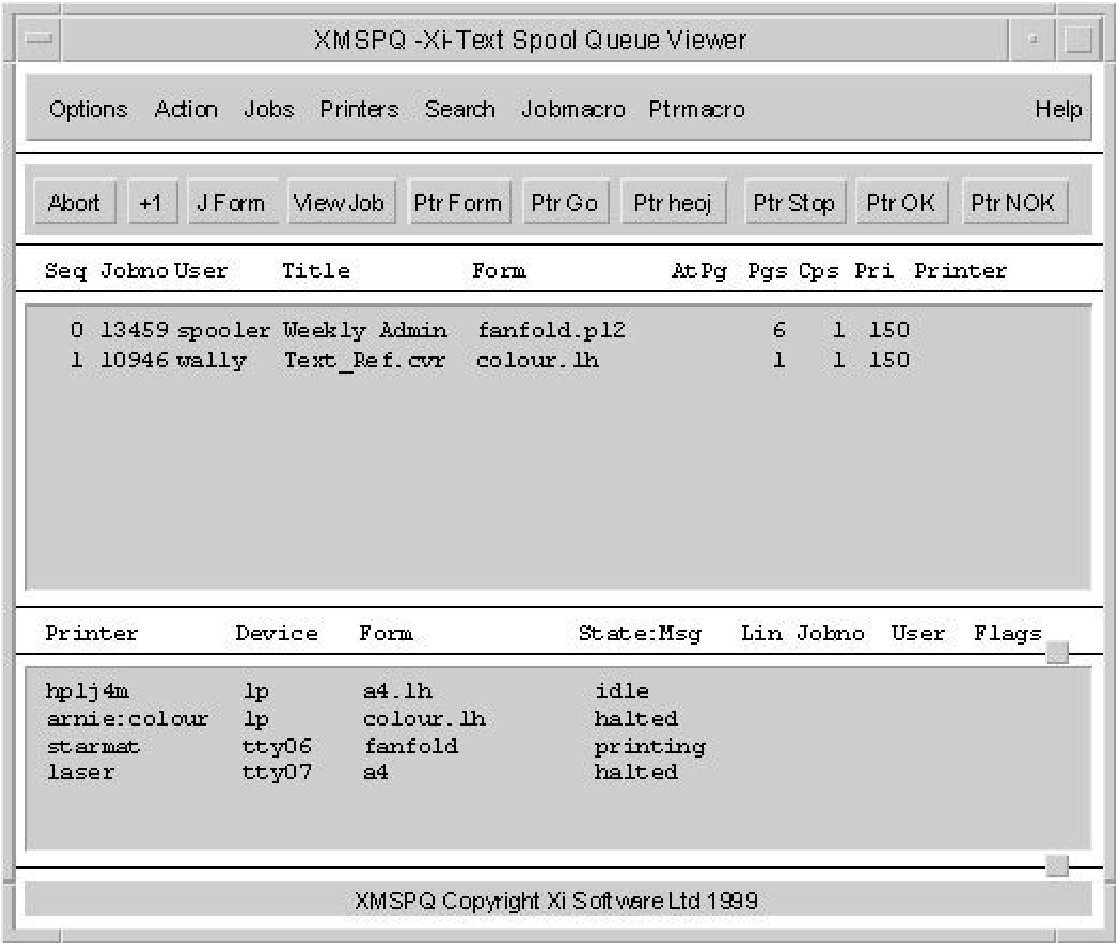
The display can be restricted to a particular user or set of users and the Window title changed to act as a reminder. One `-xrm` option is used to set the title and another is used to restrict the view by user:

```
gspl-xmpq -xrm "gspl-xmpq.title: Tony\'s Print Jobs" \
-xrm "gspl-xmpq.onlyUser: tony" &
```

In this example the user is `tony` and the Window title is "`Tony's Print Jobs`". The command has been broken over two lines by preceding the newline character with a back slash. It is not uncommon to use several lines if specifying `-xrm` options.

9.2.4 The Main Window

When gspl-xmpq is invoked the main window will be displayed. By default it will look something like this:



```
Seq Jobno User    Title          Form             AtPg Pgs Cps Pri  Printer
  0 13459 spooler Weekly Admin   fanfold.pl2            6    1  150
  1 10946 wally   Text_Ref.cvr   colour.lh              1    1  150
```

```
Printer         Device   Form           State:Msg    Lin Jobno   User   Flags
hplj4m          lp       a4.lh          idle
arnie:colour    lp       colour.lh      halted
starmat         tty06    fanfold        printing
laser           tty07    a4             halted
```

XMSPQ Copyright Xi Software Ltd 1999

The main screen is divided into two key functional areas. The top area contains menus and short cut buttons for issuing commands. The bottom area displays the print jobs and printers, which may be selected to have commands performed upon them.

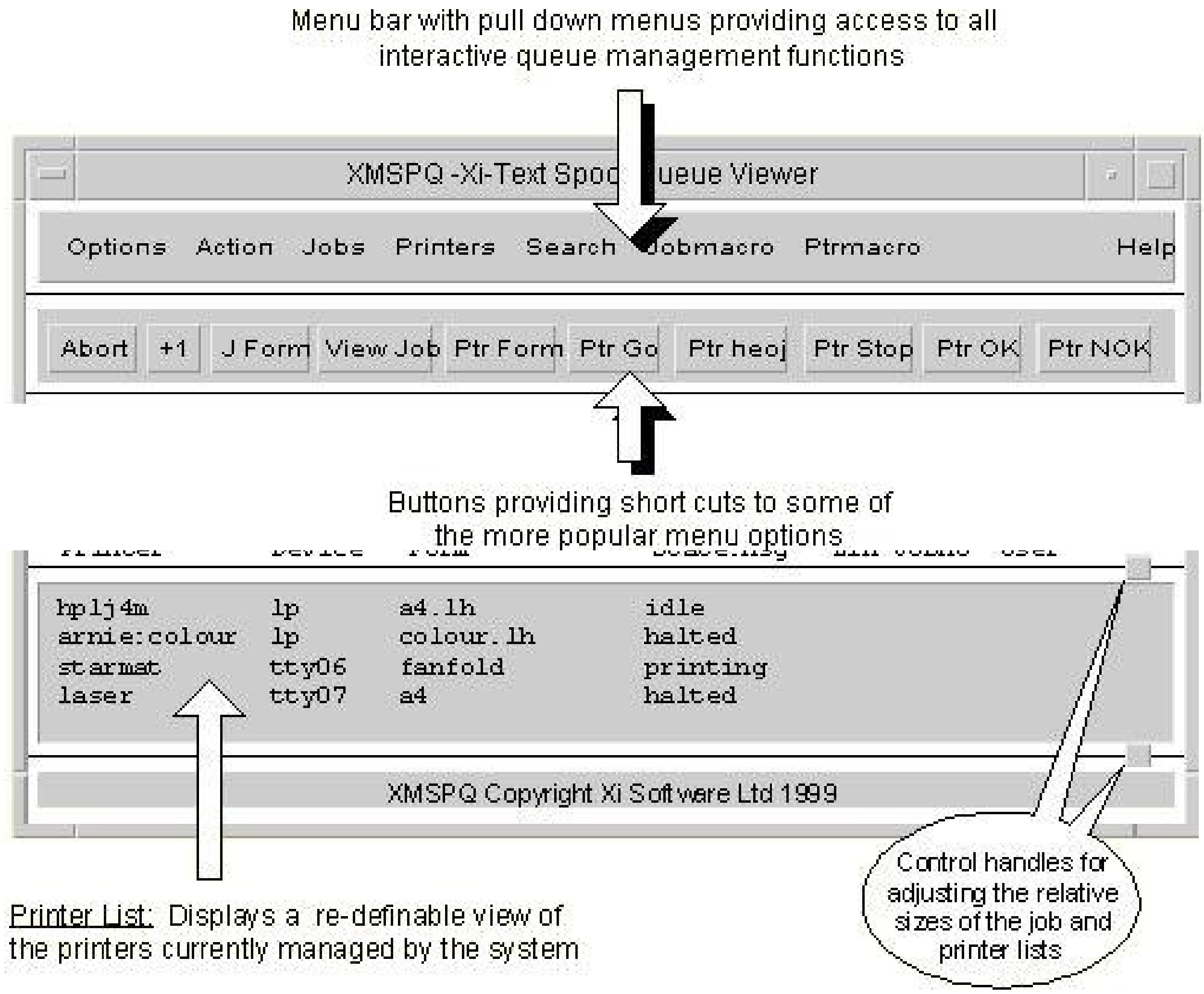
The key features of the print jobs and printers area are:

Printers and jobs can be operated upon by the appropriate menu options. The printer or job must first be selected. This is done by clicking on the line identifying it using the mouse. Once selected the line will be highlighted.

If you cannot see the printer or job that you want then you may:

- use the scroll bar or search menu options to find it
- change the view to add the item or remove unwanted items from the display.

The key features of the top area are:


Menu bar with pull down menus providing access to all interactive queue management functions

Buttons providing short cuts to some of the more popular menu options

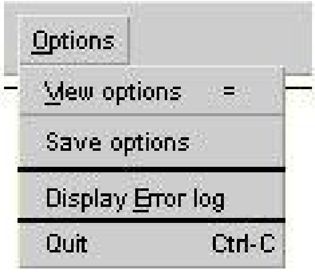
```
hplj4m        lp      a4.lh       idle
arnie:colour  lp      colour.lh   halted
starmat       tty06   fanfold     printing
laser         tty07   a4          halted
```

XMSPQ Copyright Xi Software Ltd 1999

Printer List: Displays a re-definable view of the printers currently managed by the system

Control handles for adjusting the relative sizes of the job and printer lists

9.2.4.1 The Options Menu



For tailoring the look and feel of gspl-xmpq, saving the tailored settings, viewing the error log and quitting.

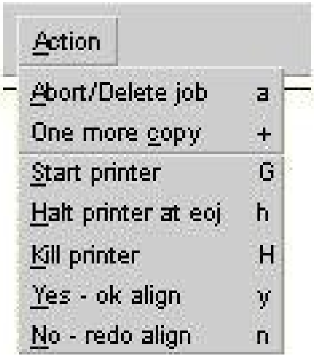
View options brings up the Display options dialogue, to tailor the look and feel. Pressing the = key also invokes this option.

Save options creates a local copy of the View options.

Display Error log brings up a Viewer showing any messages held in the GNUspool system log file, `/usr/local/var/gnuspool/spshed_reps`.

9.2.4.2 The Action Menu & Buttons

For high level actions: starting, stopping and aligning paper in printers; killing or printing another copy of a print job.



Abort/Delete job will stop the selected job if printing. Deletes it from the queue. This option is also available using the 'Abort' shortcut button.

One more copy increments the number of copies parameter for the print job. This option is also available using the '+1' button.

Start printer takes a printer out of the halted state, initialises it and enables printing. Any jobs pending for that printer start coming off as soon as the initialisation is complete.

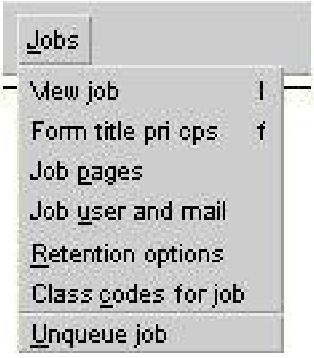
Halt printer at eoj waits for the current print job (if any) to finish. It then sends any shutdown commands required and puts the printer into the halted state.

Kill printer abandons the current print job and puts the printer into the halted state.

Yes - ok align confirms that alignment is satisfactory after some test output has been produced.

No - redo align rejects the alignment. This allows adjustment of the output media and performs another alignment test.

9.2.4.3 The Jobs Menu & Buttons



Provides options for inspecting and managing print jobs which are currently visible in the job list.

View job opens a text browser showing the data to be output. No translation of the data is performed.

Form title pri cps brings up a dialogue of the jobs formtype, title, priority and number of copies.

Job pages displays the dialogue for selecting which pages are to be output.

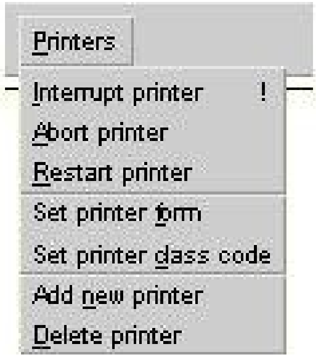
Job user and mail opens the dialogue for setting user notification options.

Retention options brings up the dialogue to select how long a job may stay on the queue and if it is to be retained after printing.

Class codes for job shows the dialogue for setting class codes.

Unqueue job allows a print job to be taken off the queue and saved in an easy to re-submit form.

9.2.4.4 The Printers Menu & Buttons



Provides options for managing printers, including adding and removing printers from GNUspool control.

Interrupt printer

Abort printer

Restart printer

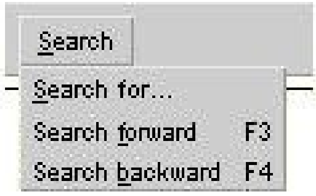
Set printer form brings up a selection browser to change the form type of the selected printer.

Set printer class code opens a dialogue for making changes to the class codes of the selected printer.

Add new printer sets up GNUspool to use another printer.

Delete printer removes a printer from GNUspool control.

9.2.4.5 The Search Menu



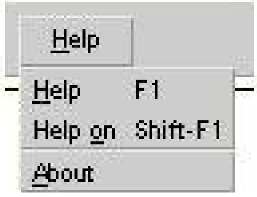
Both the printer and job lists may be navigated by using search options to find an item of interest.

Search for selected item or pattern.

Search forward from the current position

Search backward from the current position

9.2.4.6 Help



Context sensitive help for using gspl-xmpq.

Help displays help for the current window.

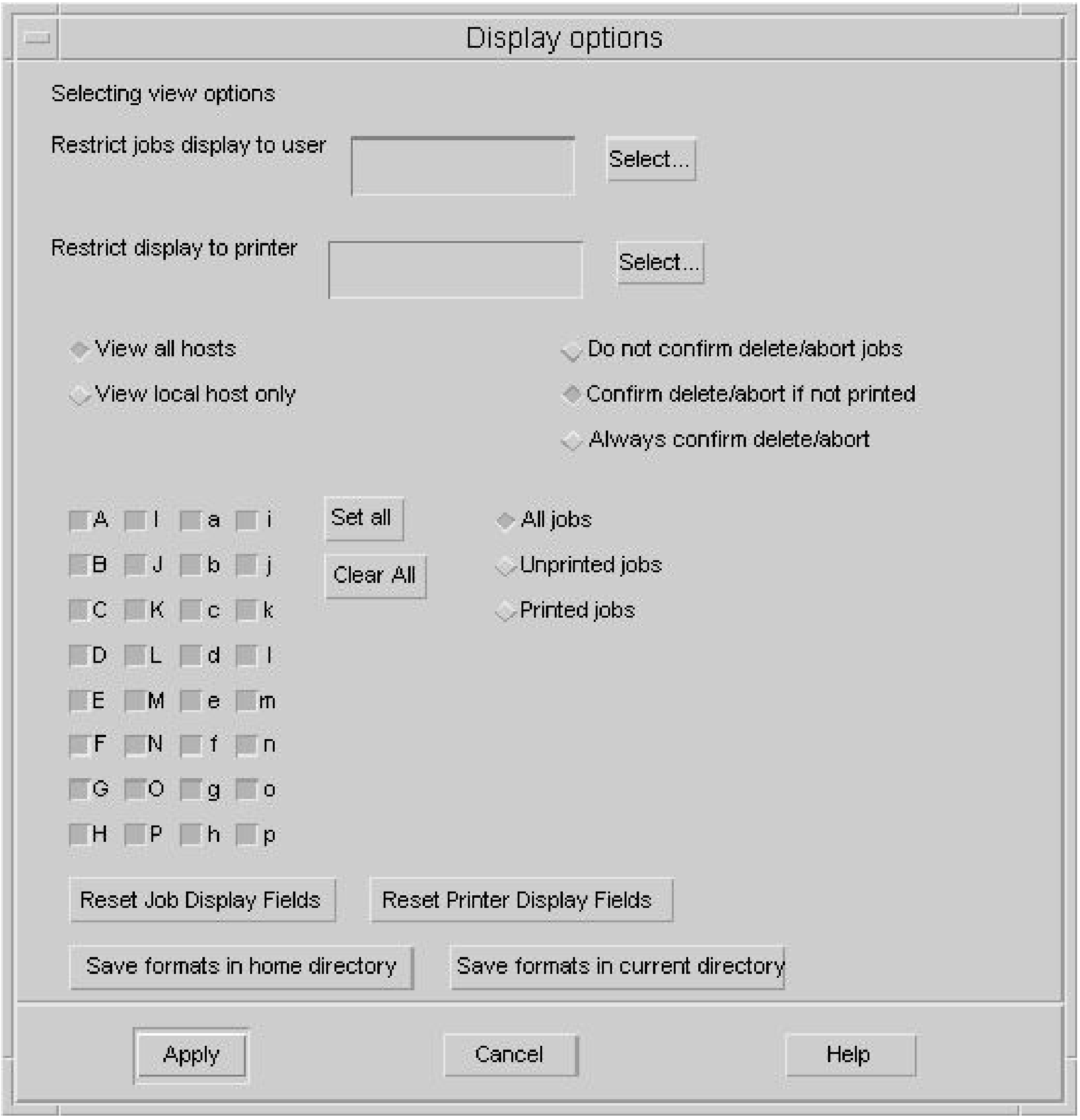
Help on changes the operating mode from taking commands to displaying help on any object (menu, button etc.) that is selected.

About displays information, such as release number, about the version of gspl-xmpq that is running.

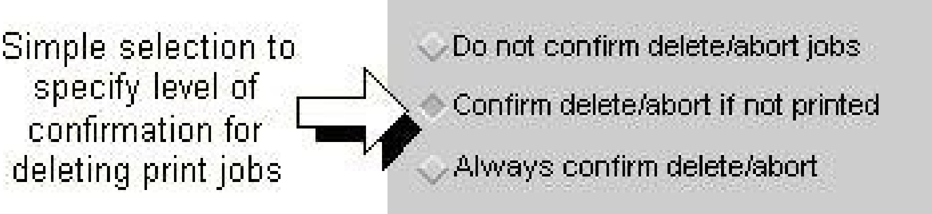
9.2.5 Setting the View Options

The content and format of information displayed by gspl-xmpq can be customised via the View options menu option under the Options menu. The level of confirmation for the more drastic commands may also be set under this option.

Selecting this option opens the following dialogue window.



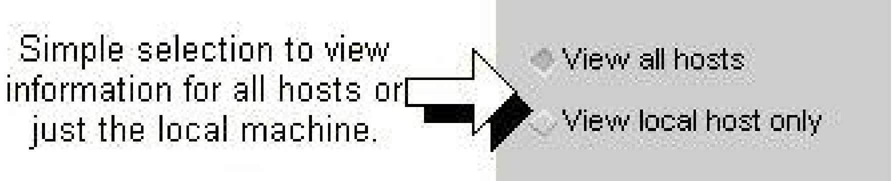
9.2.5.1 Setting the Confirmation level


Simple selection to specify level of confirmation for deleting print jobs

○ Do not confirm delete/abort jobs
● Confirm delete/abort if not printed
○ Always confirm delete/abort

By default gspl-xmpq asks for confirmation before deleting an unprinted job from the queue. This may be relaxed to allow all jobs to be deleted without confirmation. Alternatively the rule may be tightened to ask for confirmation of all jobs before deletion even if they have been printed.

9.2.5.2 Restricting the display

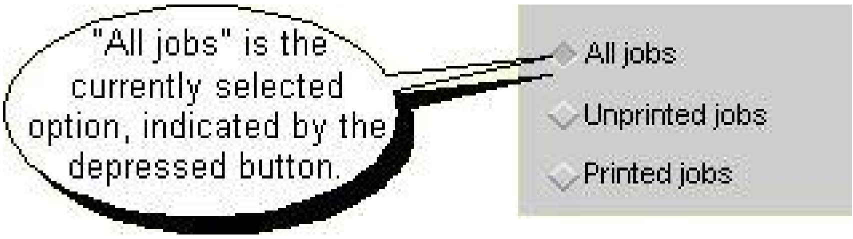

Simple selection to view information for all hosts or just the local machine.

The display may be restricted by effectively filtering to only show information for selected users, printers, class codes, job status and local or all GNUspool hosts.

9.2.5.2.1 Restricting the display to the local host

All hosts running GNUspool in the networked mode can be treated as a single system. By default gspl-xmpq will show all of the externally visible jobs and printers. The view can be restricted to show just the local job queue and printers.

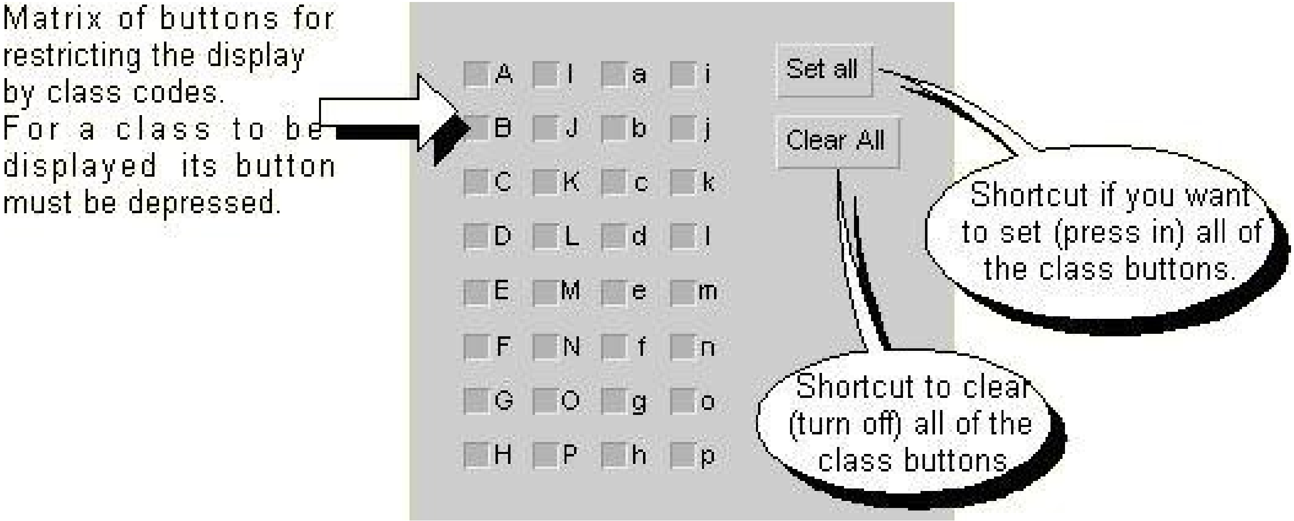
9.2.5.2.2 Restricting the display by job status


"All jobs" is the currently selected option, indicated by the depressed button.

By default gspl-xmpq displays all print jobs. This can be restricted to show just the printed or unprinted jobs.

9.2.5.2.3 Restricting the display by class code

The display can be restricted by selecting a subset of class codes. Selecting just classes A, B and C will only show those printers and jobs which have at least one of those classes set.


Matrix of buttons for restricting the display by class codes. For a class to be displayed its button must be depressed.

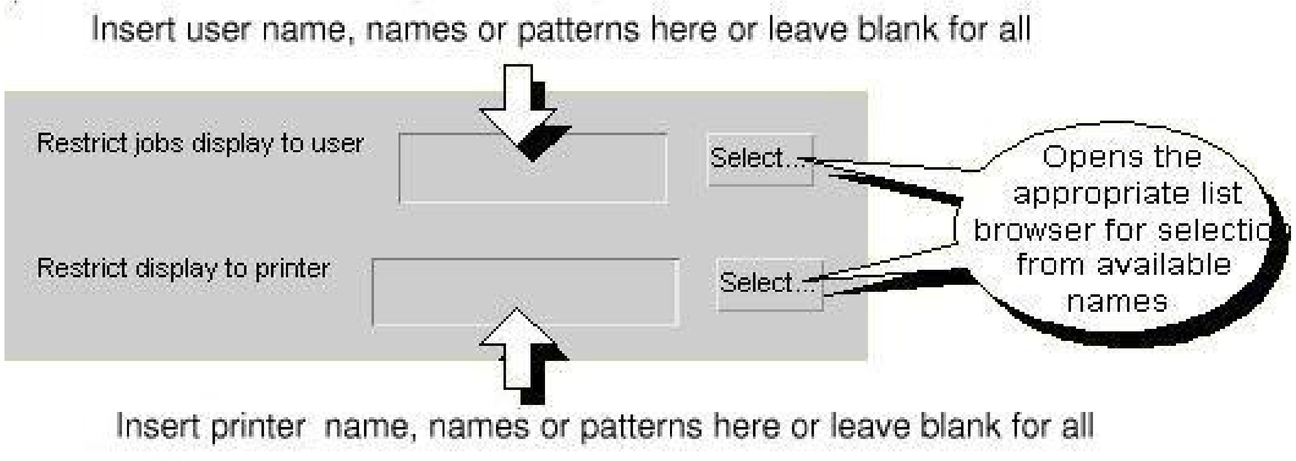
Shortcut if you want to set (press in) all of the class buttons.

Shortcut to clear (turn off) all of the class buttons

For users with "Override Class" permission the Set all and Clear all buttons provide more functionality. Pressing the **Set all** button once adds all classes in the user's set, then adds the rest if pressed a second time. Pressing **Clear All** once removes all classes not in the users set then clears everything on the second press.

9.2.5.2.4 Restricting the display by user & printer

The display may be restricted to a user or set of users and their associated jobs. Similarly it may be restricted to one or more printers, and the jobs for those printers. When used together they restrict the display to the intersection of the sets of selected jobs and printers.


Insert user name, names or patterns here or leave blank for all
Restrict jobs display to user
Select...
Restrict display to printer
Select...
Opens the appropriate list browser for selectio from available names
Insert printer name, names or patterns here or leave blank for all

Sets of users or printers may contain just one name, a list of names or a list of patterns for matching names. The list of jobs displayed when restricting by printer name includes jobs which, by having no printer name included, may come out on one of the selected printers. There is no corollary for user names, since all jobs are tied to the user who owns them.

The printer and user names may be given as a comma-separated list of alternatives, including the use of shell-style wild-cards. For example

Users	**Printers**
`Fred`	`ljet`
`jmc,tony,ukops_jmc,ukops_wal`	`lp1,ptr3,laser7`
`ukops*,ukadmin[1-5]`	`ptr*,lp[1-9]`
`[m-z]*`	`[a-c]?9`

The wild-card options are:

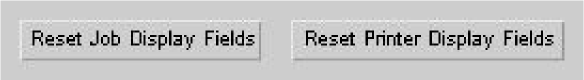
`*`	Matches anything
`?`	Matches one character
`[a-m]`	Matches one character in list or range
`[!n-z]`	Matches one char. not in list or range

9.2.5.3 Changing the fields displayed and their format

There is far more information available for both printers and jobs than could be displayed in the main window of gspl-xmpq. Different columns of information may be displayed as required. The field widths and handling of field overflow may also be adjusted.

For example: If your print jobs often have a hold time and long titles, and you do not need the printer column in the job display, then you could do the following:

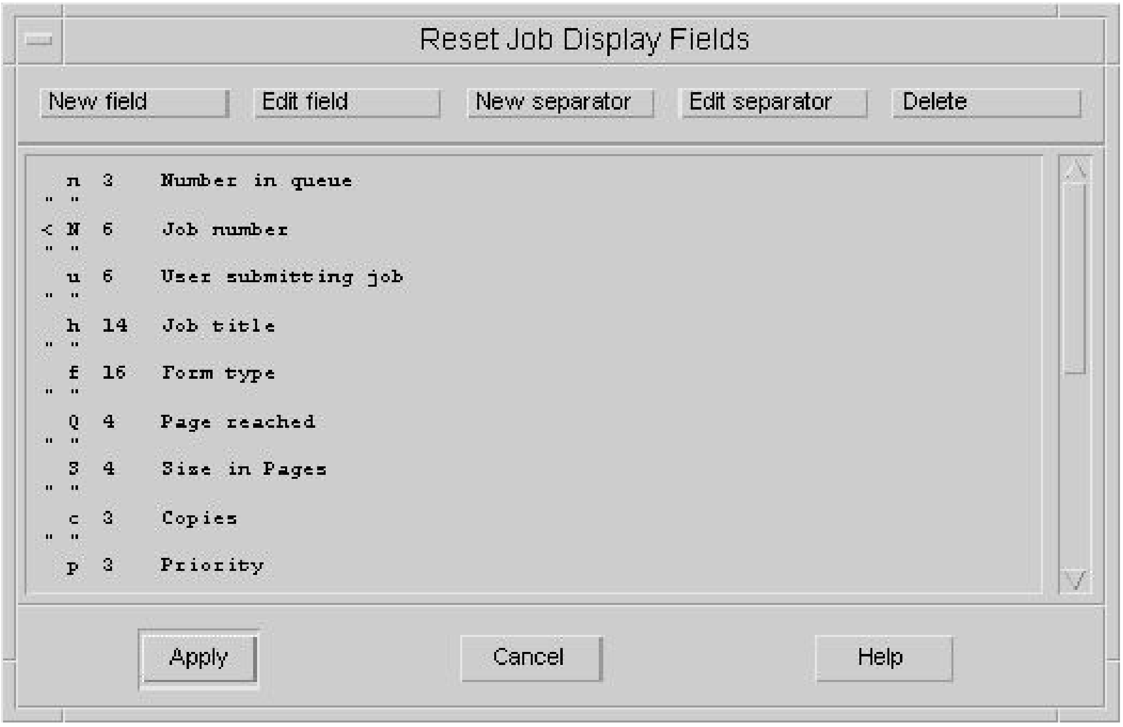
- Delete the printer column
- Make your gspl-xmpq main window wider by dragging it with the mouse.
- Add a job hold time column
- Increase the Title width from 14 to say 25 characters.



To edit the job or printer displays click on the appropriate button:

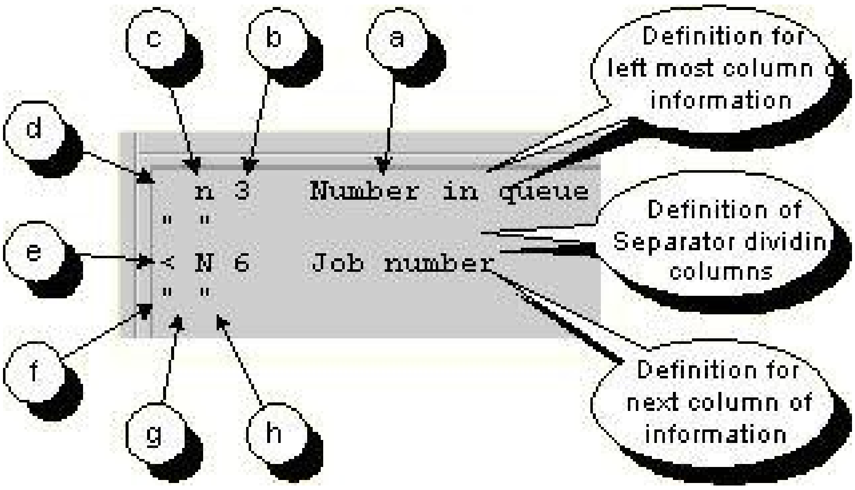
9.2.5.3.1 Changing the Job Display

Clicking on the **Reset Job Display Fields** brings up the following window. The row of buttons at the top are for adding, changing and deleting fields or separators. The fields are the columns of information and the separators are the column dividers.



```
  n  3   Number in queue
" "
< N  6   Job number
" "
  u  6   User submitting job
" "
  h  14  Job title
" "
  f  16  Form type
" "
  Q  4   Page reached
" "
  S  4   Size in Pages
" "
  c  3   Copies
" "
  p  3   Priority
```

Underneath the row of buttons is a scrollable text window showing the display format.



```
  n 3   Number in queue
" "
< N 6   Job number
" "
```

c
b
a
Definition for left most column
information
d
Definition of Separator dividin
columns
e
Definition for next column of information
f
g
h

Each line holds the specification for one column or column divider, as follows:

(a)Field description / title

(b)Width in characters

(c)Field Identifier

(d)No action on field overflow

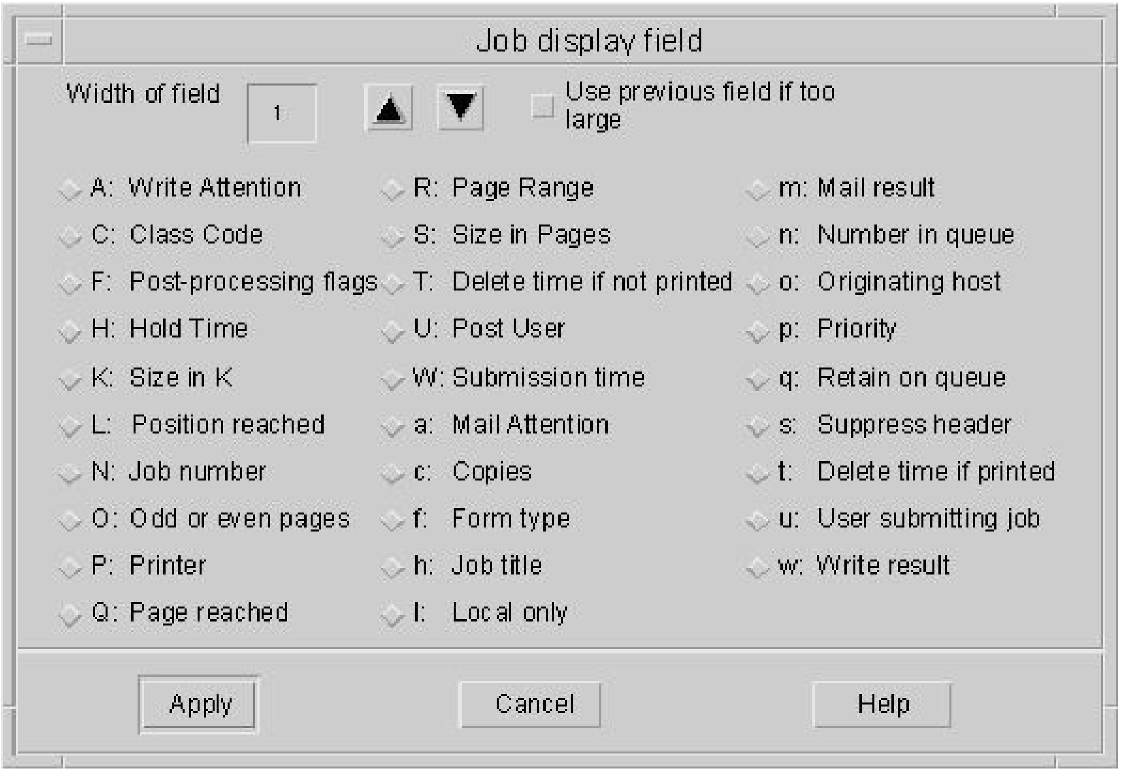
(e)Overflow onto left hand field is permitted

(f) Open quote before separator

(g)Separator character(s)

(h)Close quote after separator

To edit an existing field select the line showing the specification for that field and click on the Edit field button. To insert a new field select the line underneath the point at which you want to insert it and click on the New field button. Either of these actions will bring up a "Job display field" window looking something like this:



The width of field can be adjusted by typing in a new value or using the arrow buttons. The "Use previous field..." button is to allow a field to overflow into the field on its left. Pressing the button changes it from the deselected to selected state and vice versa.

The other buttons allow selection of what information will be displayed in the column. Only one of these can be in the selected state. When a button is selected the field width is set to a suitable value, which may then be adjusted as required.

9.2.5.3.2 Changing the Printer Display

The principles for the printer display are the same as those for the job display, with one addition. If there is too much data in a filed it may overflow into the previous field (as in the job display) or into the next field. The same windows are used, only the display information inside is different.

9.2.5.4 Saving the Format Changes

These are saved in local copies of the file gspl-xmpq helpfile which is given a user specified name at the time of saving. If a local help file has been set up, gspl-xmpq will be told to use it by the `XMSPQCONF` parameter in the local `.gnuspool` file.

9.2.5.5 Saving the View Options (excluding Format Changes)

All of the view options, excluding job and printer display formats, are saved in a users local copy of the resource file. This file is called `XI` and is kept in the user's home directory. The customised options are added to the end of the file.

The view options may be overridden by entering the Motif parameters on the command line using the -xrm option for the window manager. For example to restrict the job display to just those belonging to users kate and tom give the option:

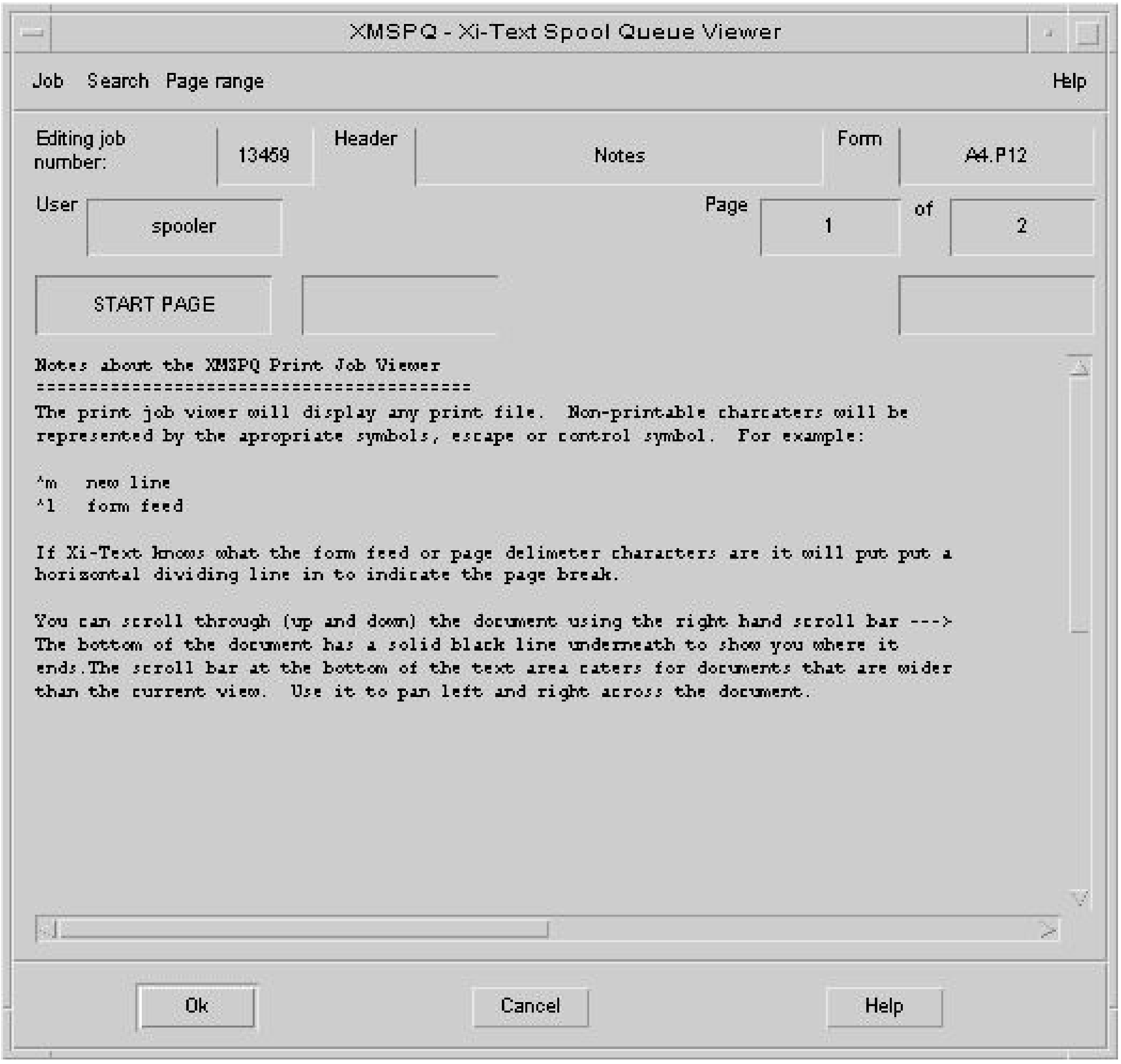
```
gspl-xmpq -xrm "xmbtq.onlyUser: kate,tom"
```

The complete set of view options, held at the end of a user's `XI` file, look like this:

```
!! gspl-xmpq User-defined options 15:02:29 98/07/10

gspl-xmpq.keepJobScroll: False
gspl-xmpq.localOnly: False
gspl-xmpq.unPrintedOnly: 0
gspl-xmpq.inclJobs: 1
gspl-xmpq.onlyClasscode: -1
gspl-xmpq.confirmAbort: Unprinted
gspl-xmpq.onlyUser:
gspl-xmpq.onlyPrinter:
gspl-xmpq.onlyTitle:
gspl-xmpq.sortPtrs: True
gspl-xmpq*jlist.width: 594
gspl-xmpq*jlist.visibleItemCount: 10
gspl-xmpq*plist.visibleItemCount: 6
```

9.2.6 Viewing a Print Job

Selecting a print job from the display then selecting the **View job** option from the **Jobs** menu opens this window.

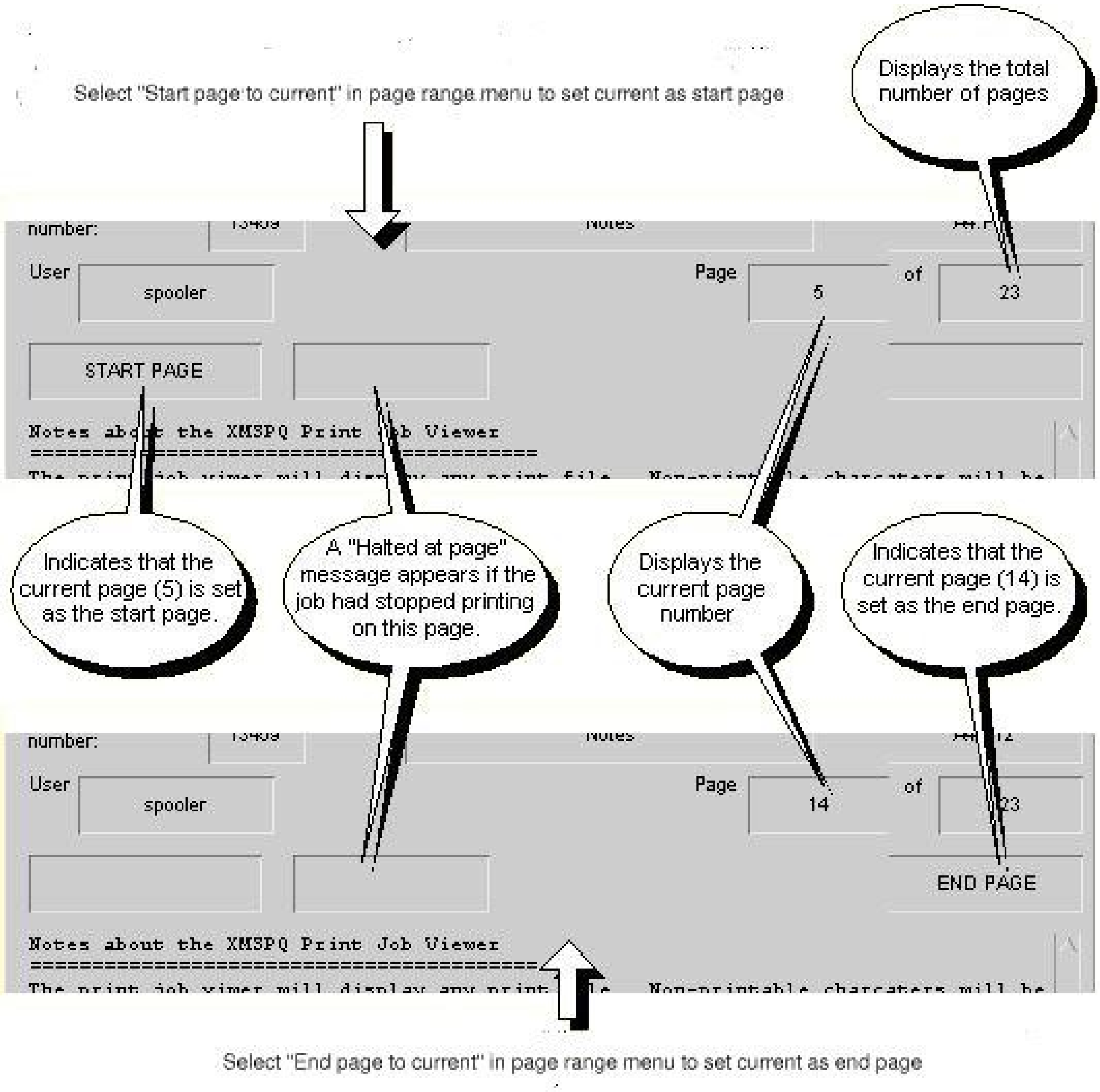


Editing job number: 13459
Header Notes
Form A4.P12
User spooler
Page 1 of 2

```
Notes about the XMSPQ Print Job Viewer
=========================================
The print job viwer will display any print file.  Non-printable charcaters will be
represented by the apropriate symbols, escape or control symbol.  For example:

^m   new line
^l   form feed

If Xi-Text knows what the form feed or page delimeter characters are it will put put a
horizontal dividing line in to indicate the page break.

You can scroll through (up and down) the document using the right hand scroll bar --->
The bottom of the document has a solid black line underneath to show you where it
ends.The scroll bar at the bottom of the text area caters for documents that are wider
than the current view.  Use it to pan left and right across the document.
```

The display may be scrolled through the document and panned across the text using the vertical and horizontal scroll bars. The Search Menu provides options for specifying and finding text strings within the print job.

A selected range of pages from a document may be printed. The start and end pages may be viewed and set from within this window. If a print job has stopped during printing the page on which it halted is also shown.



Select "Start page to current" in page range menu to set current as start page
Displays the total number of pages
number:
Notes
User
Page
of
START PAGE
Indicates that the current page (5) is set as the start page.
A "Halted at page" message appears if the job had stopped printing on this page.
Displays the current page number
Indicates that the current page (14) is set as the end page.
END PAGE
Select "End page to current" in page range menu to set current as end page

User spooler Page 5 of 23

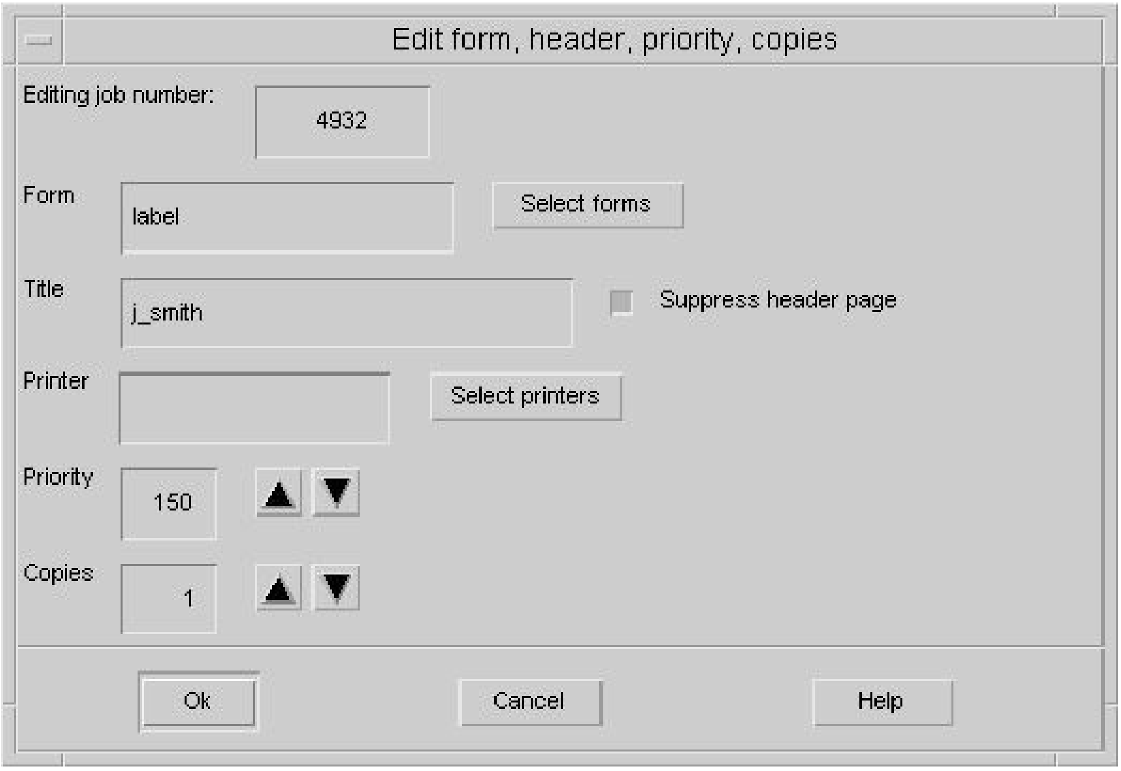
Notes about the XMSPQ Print Job Viewer

User spooler Page 14 of 23

Notes about the XMSPQ Print Job Viewer

9.2.7 Changing Job and Printer parameters

A job may be deleted, changed, reviewed by clicking on the line representing it in the job list and then selecting the required menu option or short cut button. Printers may operated upon in exactly the same way. Some menu options and short cut buttons will have an immediate effect. Others will open a dialogue window for additional information, such as this one for changing the form type, header, priority and number of copies for a print job.



Editing job number: 4932

Form: label

Title: j_smith

☐ Suppress header page

Printer:

Priority: 150

Copies: 1

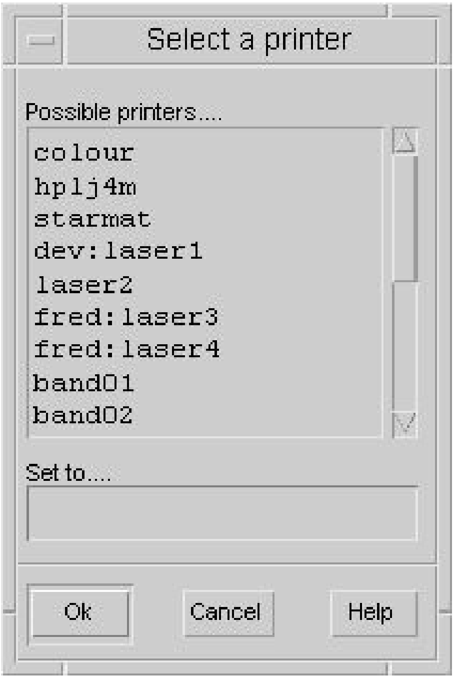
Fields like Priority and Copies take a numeric value which may be typed in or adjusted using the increment and decrement buttons.

Plain text fields like Title simply allow any free text to be entered or modified.

Square buttons represent options which can be enabled or disabled (true or false) like the Suppress header page. The Diamond shape buttons (not shown in this example) are used for selecting one option out of a set of 2 or more possible options.

Some fields like the Printer and Formtype in this example have a Select button next to them. These support two methods of parameter entry, straight text entry as in the Title field or by selecting one option from a list using a selection dialogue. (With some versions of the Motif library, this may be a "Combo Box").

Clicking on the Select printers button, for example, opens this selection dialogue.



The information displayed is context and configuration sensitive, showing only the permitted and/or appropriate information.

In this example gspl-xmpq is looking at all printers and jobs on a set of networked GNUspool machines. `laser1` is on host `dev`, `laser3` and `laser4` are on host `fred`.

Printers which the user is not allowed to access due to class restrictions will not appear in their view of this list.

9.3 gspl-xmuser - Motif GUI User Administration Tool

gspl-xmuser is a fully interactive Motif alternative to the standard user permission manager (invoked using the command `gspl-user -i`). It is provided to maintain the list of user privileges, charges and default modes for jobs and printers.

Unlike gspl-user there are no command line options to gspl-xmuser, it is always in the interactive mode (similar to gspl-user -i). The facility to change or specify resource settings for an X11 (and hence Motif) program on the command line can be used.

Motif is not supported by GNU and this has been rewritten as gspl-xuser using GTK+. The interface is broadly similar and this document will be updated to reflect it.

9.3.1 Options

gspl-xuser (the GTK+ version) does not use a resource file.

The option `-xrm "str"` may be used to specify a resource *`str`* to override the resource given in the resource file.

Do not worry about the terminology it is sufficient to recognise what these resources look like. For example the resource which specified the title for the main window of gspl-xmuser is:

```
gspl-xmuser.title: gspl-xmuser - Edit user options
```

The text before the colon "`:`" is the resource name (i.e. `gspl-xmuser.title`) and the text after is the data, in this case a string. The data can be boolean, integer or a string.

9.3.2 Useful Resources

All of the resources for program gspl-xmuser start with "`gspl-xmuser`". To keep the initial column width to manageable proportions just the text from the "`.`" or "`*`" is listed.

`.Title:`	Specifies the Title as it will appear on the main window of gspl-xmuser.
`.IconName:`	Sets the text for the icon when gspl-xmuser is iconised.
`.sortAlpha:`	This is a boolean with the values True or False. When True the list of users will be sorted into alphabetical order by username.

9.3.3 Examples

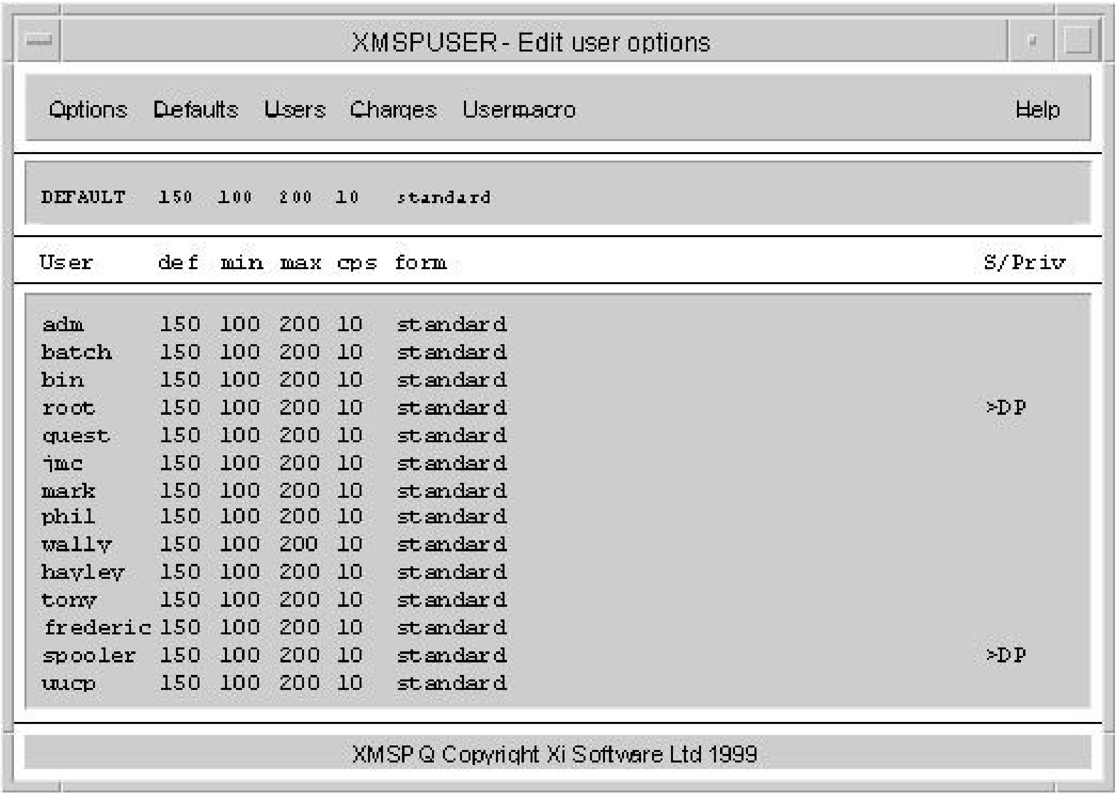
When using an X-Terminal connected to several GNUspool machines it can be very useful to include the hostname in the Window and Icon titles. The display can be restricted to a particular user or set of users and the Window title changed to act as a reminder. One `-xrm` option is used to set the title and another is used to restrict the view by user:

```
gspl-xmuser -xrm "gspl-xmuser.title: gspl-xmuser - $HOSTNAME" \
-xrm "gspl-xmuser.IconNAME: $HOSTNAME" &
```

In this example the machine's name will be picked up from the environment variable `HOSTNAME`. It is not uncommon to use several lines if specifying `-xrm` options. Of course this can all be put in a a shell "wrapper" to save typing each time.

9.3.4 The Main Window

When gspl-xmuser is invoked the main window will be displayed. By default it will look something like this:

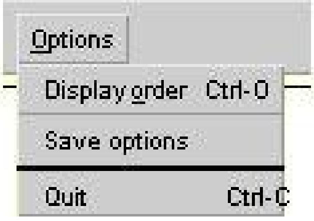

XMSPUSER - Edit user options
Options
Defaults
Users
Charges
Usermacro
Help
XMSP Q Copyright Xi Software Ltd 1999

```
DEFAULT   150  100  200  10   standard

User      def  min  max  cps  form                                    S/Priv

adm       150  100  200  10   standard
batch     150  100  200  10   standard
bin       150  100  200  10   standard
root      150  100  200  10   standard                                >DP
quest     150  100  200  10   standard
jmc       150  100  200  10   standard
mark      150  100  200  10   standard
phil      150  100  200  10   standard
wally     150  100  200  10   standard
hayley    150  100  200  10   standard
tony      150  100  200  10   standard
frederic  150  100  200  10   standard
spooler   150  100  200  10   standard                                >DP
uucp      150  100  200  10   standard
```

The bottom area contains a list of all users in the password file with their GNUspool permissions and settings. It will have a scroll bar if there are more users than can fit on the screen. Above this is a pane containing the default settings.

At the top of the screen is the menu bar supporting all of the gspl-xmuser commands. Each menu option opens a dialogue or operates on the specified data immediately. With a few exceptions these are straight forward and easy to understand.

9.3.4.1 The Options Menu



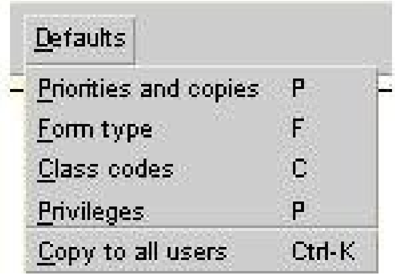
This is for changing the ordering of users in the display, saving the settings and quitting.

Display order brings up the Display options dialogue, to tailor the look and feel. Pressing the = key also invokes this option.

Save options saves a local copy of the resources file with the user's choice of Display order.

Quit saves any changes to the default and user accounts. N.B. This is the only way of saving these changes. Quitting by any other method will discard the changes.

9.3.4.2 The Defaults Menu



Administers the default account parameters.

Priorities and copies opens the dialogue for setting up maximum number of copies, priority range and defaults.

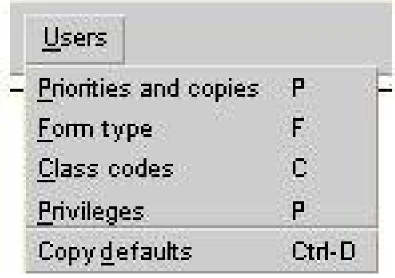
Form type opens the form type selection dialogue.

Class codes opens the class code selection dialogue.

Privileges opens a dialogue showing and allowing changes to the privileges (e.g. starting and stopping printers).

Copy to all users Copies the default settings to all users - use with care.

9.3.4.3 The Users Menu



Administers the accounts of individual users. **Priorities and copies** opens the dialogue for setting up maximum number of copies, priority range and defaults.

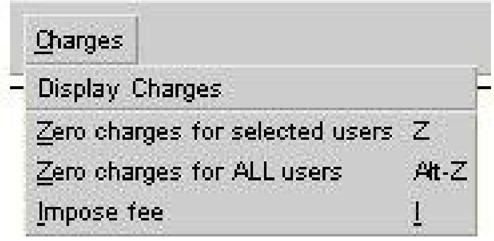
Form type opens the form type selection dialogue.

Class codes opens the class code selection dialogue.

Privileges opens a dialogue showing and allowing changes to the privileges (e.g. starting and stopping printers).

Copy defaults resets the users account to the default settings - use with care.

9.3.4.4 The Charges Menu


Charges
Display Charges
Zero charges for selected users Z
Zero charges for ALL users Alt-Z
Impose fee I

Displays and manages the printing charges.

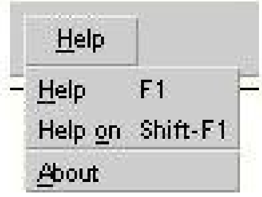
Display Charges operates for all selected users.

Zero charges for selected users

Zero charges for ALL users

Impose fee on selected users.

9.3.4.5 The Help Menu



Context sensitive help for using gspl-xmuser.

Help displays help for the current window.

Help on changes the operating mode from taking commands to displaying help on any object (menu, button etc.) that is selected.

About displays information, such as release number, about the version of gspl-xmuser that is running.

9.3.5 Selecting multiple users for menu options

Many of the menu options can be carried out on more than one user at a time. First select all of the relevant users from the list. Then select the required menu option for the operation you wish to perform. The set of users remains selected after you have completed the operation in case there are other options that you want to use on them.

You can scroll up and down a long list of users without losing those you have selected so far. The following mechanisms allow you to select a group of users:

- Hold the selection mouse button down and drag it across a contiguous set of users.
- Using the mouse click on the first user in the required set. Move to the last user in the set, hold the shift key down on the keyboard and click on this user.
- Individual users may be added or removed from the set by clicking on them whilst holding the Control key down on the keyboard.

9.3.6 Copying defaults to all users

Available via the **Copy to all users** option under the **Defaults** menu. It copies the default settings to all users except the one you are logged in as. This command should be used with care. It is possible to remove essential permissions from everybody including write administration file privilege.

9.3.7 Resetting a user to the default

This is a very useful mechanism for setting permissions for a group of users. Set the default to the required value then apply it to the required users. If you do not want new users to inherit this default setting remember to return it to the original state.

Be careful not to remove write administration file privilege from all users.

10 Terminal Server support commands

The following support commands for terminal servers are provided as standard with **GNUspool**, installed in the internal programs directory, by default `/usr/local/libexec`.

10.1 Xtlpc

```
/usr/local/libexec/xtlpc -H hostname
                          -f ctrlfile
                         [-F fifo]
                         [-S host]
                         [-P ptrname]
                         [-N] [file]
```

xtlpc is a terminal server program for use by **GNUspool** where the destination is an LPD protocol server. Input is usually taken from standard input, or from the supplied file name.

For information on the format of the control file for xtlpc, please see chapter 13.7.

10.1.1 Options for xtlpc

Option	Description
`-H hostname`	This is the host name or IP address to be used to access the printer. In **GNUspool** setup files, it may be appropriate to substitute `$SPOOLDEV` or `$SPOOLPTR` here.
`-f ctrlfile`	This specifies the location of the control file, typically `xtlpc-ctrl` in the internal date directory, usually `/usr/local/share`.
`-S hostname`	This is the host name or IP address of the sending host, in the cases where gethostname(3) does not yield an appropriate answer.
`-P ptrname`	The LPD protocol supplies a name to the printer. By default, this is taken from SPOOLPTR, i.e. the name given as the name of the printer under **GNUspool**. Some terminal servers require a specific name to be given to the printer, perhaps to select which printer, or printer option of several is to be specified. This option overrides this printer name in the protocol. (N.B. This option was introduced with version 23.10 of xtlpc in May 2002).
`-F fifo`	Take input from the named fifo rather than standard input or the supplied file. This is useful in situations where the LPD protocol has to be run from a command line, i.e. a console session, and files sent to it by having **GNUspool** write to the fifo.
`-N`	Avoid sending zero length files, which confuse some servers.
`-U`	Do not insist on a reserved port locally

10.1.2 Exit codes from xtlpc

Normal termination is denoted by an exit code of zero.

1. Usage error, invalid option etc.

2. No control file given.
3. No destination host given.
4. Cannot open input file.
5. Unknown host name.
6. Cannot open control file.
7. Syntax errors in control file.
8. No spool directory defined.
9. Cannot select spool directory.
10. Control card format not given in control file
11. Invalid format control card in control file
12. Cannot write to spool directory
13. Cannot create spool file
14. Cannot allocate file name.
15. Cannot open FIFO.
16. Name given is not a FIFO.
17. Cannot specify input and FIFO.
18. Cannot create sequence file
19. No port name in control file
20. TCP protocol missing
21. Cannot find named port in service file
22. Cannot open socket.
23. Cannot bind port
24. Cannot allocate TCP device
25. Cannot make connection
26. Cannot allocate TCP space or change stream
27. Write failure writing to socket
28. Lost spool file
29. Read failure ack packet.
30. Error return from remote.
31. Cannot determine own host name
32. Invalid host name

33. Exit on signal.

10.2 gspl-lpq

```
gspl-lpq [ -l ] [ -s ] printer
```

gspl-lpq obtains and outputs a list of pending jobs on an external print server supporting the LPD protocol.

10.2.1 Options

This is a user-level program that interfaces with the LPD interface software and the option-handling mechanism is supported.

The environment variable on which options are supplied is `SPLPQ` and the environment variable to specify the help file is `LPDINTCONF`.

Option		Description
`-?`	`+explain`	Causes a summary of the other options to be displayed without taking further action.
`-l`	`+long-list`	Produce a long listing.
`-s`	`+short-list`	Produce a short listing (default).
`+freeze-current`		Save all the current options in a `.gnuspool` file in the current directory.
`+freeze-home`		Save all the current options in a .gnuspool file in the user's home directory.

10.2.2 gspl-lprm

```
gspl-lprm printer job ...
```

gspl-lprm deletes pending jobs on an external print server supporting the LPD protocol. The specified printer is that known to GNUspool and the details are obtained from the printer setup file.

10.3 Xilp

```
/usr/local/libexec/xilp -H hostname
                        [ -p port ]
                        [ -h ]
                        [ -r ]
                        [ -# n ]
                        [ -U user ]
                        [ -P ptr ]
                        [ -J name ]
                        [ -F name ] [ file ... ]
```

xilp is a terminal server program for use by **GNUspool** where the destination is an LPD server.

It has been largely superseded by xtlpc.

10.3.1 Options

Option	Description
`-H hostname`	This is the host name or IP address to be used to access the printer. In **GNUspool** setup files, it may be appropriate to substitute `$SPOOLDEV` or `$SPOOLPTR` here.
`-p port number`	This is the port number or service port name to be used.
`-h`	Enable banner page
`-r`	Remove file on completion
`-# n`	Set number of copies to n.
`-U user`	Set user name to user.
`-P ptr`	Set printer name to ptr
`-C class`	Set printer name to class
`-J file`	Synonym for `-F`.
`-F file`	Set file name to file.

10.4 Xtelnet

```
/usr/local/libexec/xtelnet -h hostname
                             -p port
                           [ -d ]
                           [ -l loops ]
                           [ -L loopwait ]
                           [ -t sleep ]
                           [ -s lingertime ]
```

xtelnet is a terminal server program for use by **GNUspool** where the destination is a server which accepts "reverse telnet" protocol. Data is accepted on standard input.

10.4.1 Options

Option	Description
`-h hostname`	This is the host name or IP address to be used to access the printer. In **GNUspool** setup files, it may be appropriate to substitute `$SPOOLDEV` or `$SPOOLPTR` here.
`-p port number`	This is the port number (or name in `/etc/services`) to be used. In **GNUspool** setup files, it may sometimes be appropriate to substitute `$SPOOLDEV` here, but this is unlikely.
`-l n`	Make n connection attempts before giving up trying to connect to the server.
`-L n`	In conjunction with `-l` option, sleep n seconds between attempts to connect to the server.
`-t n`	Sleep n seconds before exiting, after closing the connection to the server. This is sometimes necessary with some servers, as **GNUspool** operates too fast for them. However you may prefer to use the `closewait` option in the setup file.
`-s n`	Specify "linger time" in seconds (which may be fractional) for the connection. This is likewise necessary with some servers.

Option	Description
`-d`	Debug. Displays activities on standard error.

10.4.2 Exit codes

Normal termination is denoted by an exit code of zero.

1. Device appears to be offline.
2. Some kind of device error, failure to connect etc.
3. Incorrect usage, unknown option etc.
4. System error, cannot find host name etc.

10.5 Xtlhp

```
/usr/local/libexec/xtlhp -h hostname
                          -p port
                        [ -d n ]
                        [ -f file ]
                        [ -c file ]
                        [ -l log ]
                        [ -b blksize ]
                        [ -c community ]
                        [ -T timeout ]
                        [ -S snmp ]
                        [ -D n ]
```

xtlhp is a terminal server program for use by **GNUspool** where the destination is an HPNPF server. Alternatively other printer devices which accept input using "reverse telnet" and allow status to be interrogated via SNMP (or other means) may be driven.

After each block is transmitted, the status of the printer is interrogated using a supplied control script.

10.5.1 Options

Option	Description
`-h hostname`	This is the host name or IP address to be used to access the printer. In **GNUspool** setup files, it may be appropriate to substitute `$SPOOLDEV` or `$SPOOLPTR` here.
`-p port`	This is the port number or service port name to be used.
`-f configname`	This is the location of the definitions file, if not `xtsnmpdef` in the **GNUspool** internal programs directory, usually `/usr/local/libexec`.
`-c file`	Specifies a control file for obtaining the printer status. See the Control File section below.
`-l logfile`	Specify error message destination if not standard error.
`-b n`	Specify output block size. n may be a number, or followed by **b**, **k**, **m** or **g** to denote blocks, kilobytes, megabytes or gigabytes. Otherwise a default block size of 10K is assumed.

Option	Description
`-d n`	Debug. Displays activities on standard error, if n is greater than zero. Higher values of n will give more output, including tracing of SNMP operations.
`-D n`	As for **-d**, except that no output is attempted, the control script is executed once only and the appropriate exit code returned. Use an argument of zero to just invoke the script and exit.
`-c name`	Specify the community in SNMP operations. The default is `public`.
`-T timeout`	Specify the timeout to wait for SNMP replies. The default is 1 second, but the value given may be fractional.
`-S snmp`	Specify the SNMP port name to use, if not `snmp`.

10.5.2 Configuration file

The configuration file `xtsmnpdef` provides a series of mnemonics for the SNMP definition of the form:

```
system        1.3.6.1.4.1.11.2
gdStatusBytes system.3.9.1.1.1
```

The function of these definitions is to "macrogenerate" constructs which appear in the control file, for example (with appropriate definitions)

```
gdStatusPaperOut.0
```

may be used as an abbreviation for

```
1.3.6.1.4.1.11.2.3.9.1.2.8.0
```

Previous versions of xtlhp expected quotes around the columns of the file, but this is no longer expected and they will be discarded, provided that they are properly matched. The file may define names in terms of previously-defined names and may redefine names. If a name is redefined, names already defined using the previous definition are unaffected.

10.5.3 Control file format

The control file consists of a series of command structures involving options to obtain SNMP values, run commands, string and numeric comparisons and assignments.

Constructs are based upon the shell, but control names are in upper case.

10.5.3.1 Expressions

`'string'`	A single-quoted string is taken literally, as per the shell.
`"string"`	A string enclosed in double-quote characters is subject to name expansions of macros defined in the configuration file and variables assigned in the script.
`SNMPVAR(objid)`	The specified *objid*, which may be specified using macros, is returned, treated as a numeric value.
`SNMPSTR(objid)`	The specified *objid*, which may be specified using macros, is returned, treated as a string value.

`` `string` ``	A back-quoted string is a shell command to execute and the output taken as a string. Variable names are expanded within the string before execution.
`[string]`	A string enclosed in square brackets is a shell command to execute, with the output taken to be a decimal numeric value, the whole returning a numeric value. Variable names are expanded within the string before execution.
`{string}`	A string enclosed in braces is a shell command to execute, with the output assumed to be a hexadecimal value. (This is required for snmpinfo(8) on AIX systems, which returns hexadecimal values). Variable names are expanded within the string before execution.
`number`	a numeric value.
`name`	The value currently assigned to name, if any, is the value of the expression.
`LASTVAL`	The last SNMP value obtained (possibly using `DEFINED` or `UNDEFINED`) as a numeric or string value.

10.5.3.2 Constructs

The following constructs or "statements" are used in the control file.

`=`	Assignment, as in `X = "ABC"` The value on the right is assigned to the name on the left.

`:=`	One time assignment. As per assignment, but the operation is turned into a no-op after the first execution.
	For example
	`` Host = `hostname` ``
	would be evaluated each time the script is run (after every block), but
	`` Host := `hostname` ``
	would only be evaluated the first time.

comparison	Comparison of strings is performed with alphabetic operators `LT`, `LE`, `EQ`, `NE`, `GE` and `GT`, and numerics with symbolic operators `<` etc.
`DEFINED(objid)`	Return a true value if the given object id (which may be specified using macros) is defined, otherwise false. If the value is defined, `LASTVAL` is set to whatever value that was to save further fetches.
`UNDEFINED(objid)`	As for `DEFINED` but the other truth value is returned. `LASTVAL` is still set if the value is defined.
`ISNUM value`	Return a true value if the value is a number.
`ISSTRING value`	Return a true value if the value is a string.

AND OR && ||	Combine boolean operations in tests. AND is more binding than OR. "Short circuit" evaluation is performed, so beware of relying on LASTVAL
IF .. THEN .. [ELIF .. THEN] [ELSE ..] FI	Conditional construct with optional ELIF and ELSE parts.
MSG STRING	Display message on standard error.
EXIT number	Exit xtlhp with specified exit code.
FLUSH	Send flush command (ESC-E) to socket.

10.5.3.3 Example

This is the control file supplied by default, with definitions for gdStatusLineState etc being provided in the definitions file supplied.

Note that if no EXIT is reached, then script file has "succeeded" and output continues.

```
   # Test the status to see if there are problems

  IF SNMPVAR(gdStatusLineState.0) != 0
  THEN
          # Look at paper out indication and exit
          # message

          IF SNMPVAR(gdStatusPaperOut.0) != 0
          THEN
                  MSG 'Out of paper'
                  EXIT 100

          # Likewise paper jam

          ELIF SNMPVAR(gdStatusPaperJam.0) != 0
          THEN
                  MSG 'Out of paper'
                  EXIT 101

          # Likewise toner low

          ELIF SNMPVAR(gdStatusTonerLow.0) != 0
          THEN
                  MSG 'Out of toner'
                  EXIT 102

          #  Give up just give general message

          ELSE
                  MSG SNMPSTR(npSysStatusMessage.0)
                  EXIT 103
          FI
   FI
```

10.6 Xtftp

```
/usr/local/libexec/xtftp -h hostname
                     [ -p port ]
                     [ -d ]
                     [ -u username ]
                     [ -w password ]
                     [ -o outfile ]
                     [ -i infile ]
                     [ -D directory ]
                     [ -T timeout ]
                     [ -A host ]
                     [ -T ]
                     [ -B ]
```

xtftp is a terminal server program for use by **GNUspool** where the destination is a server which accepts FTP protocol. Data is taken from standard input, unless overridden by the `-i` option.

10.6.1 Options

Option	Description
`-h hostname`	This is the host name or IP address to be used to access the printer. In **GNUspool** setup files, it may be appropriate to substitute `$SPOOLDEV` or `$SPOOLPTR` here.
`-A hostname`	Specifies the sending hostname or IP address to be as given. This may be required when sending from hosts with multiple IP addresses.
`-p port number`	This is the port number to be used, if not `ftp`.
`-u user`	This is the user name to be quoted to the FTP server.
`-w word`	This is the password to be quoted to the FTP server.
`-o outfile`	This is the output file to be stored at the server end. In some cases this may designate the actual printer, and it may be useful to use `$SPOOLDEV` as the argument to be expanded in **GNUspool** setup files.
`-D directory`	This is the output file to be stored in at the server end, if not the "home directory". In some cases this may designate the actual printer, and it may be useful to use `$SPOOLDEV` as the argument to be expanded in **GNUspool** setup files.
`-T timeout`	This gives the timeout in milliseconds to await for responses from the server.
`-i input`	Specify input file if not standard input.
`-t`	Specify text mode.
`-b`	Specify binary mode.
`-d`	Debug. Displays activities on standard error. Increasing information is given if more than one `-d` option is specified.

10.6.2 Exit Codes

Normal termination is denoted by an exit code of zero.

1. Usage error, invalid option etc.
2. No host name or ip address given.
3. Cannot open input file.
4. Expecting password as well as user name.
5. Incorrect password.
6. Transmit failure - file rejected.
7. Data accept failed.
8. Cannot create/bind control socket.
9. Cannot connect control socket.
10. Cannot create data socket.
11. Cannot bind data socket.
12. Cannot listen data socket.
13. Cannot get socket name data socket.
14. Unknown service name
15. Invalid IP address
16. Invalid host name
17. Socket write error.
18. Socket read error.
19. No response code from server.
20. Port or store command failed.
21. Generated command too long (excessive length file name etc).
22. Aborted with a signal.

11 Configurability

This section looks at the various options available to user programs such as gspl-pr and gspl-pq. In some cases, particularly with gspl-pr and gspl-qchange, there is a perhaps bewildering array of options. It is not intended that users should have to remember them all as they can be specified by default and overridden as required.

This is done by using *configuration files* and environment variables.

11.1 Configuration files

A configuration file is a file called `.gnuspool`, which may appear in the current directory, or in the user's home directory. This file is a text file, containing environment variable type assignments. It may also be used, as described below, to specify alternative help message files.

Options to the user programs enable these files to be generated and edited automatically.

Hopefully you will be familiar with the `-f` *`form`* and the `-c` *`n`* options to gspl-pr to set form type and copies. Suppose you want to print a number of documents with form type `a4.lh`, and with 3 copies. Rather than specifying each time

```
gspl-pr -f a4.lh -c3 ......
```

it would be easier to put these options into the environment by means of:

```
$ GSPL_PR='-f a4.lh -c3'
$ export GSPL_PR
```

and then they would be automatically specified each time, until you set the environment variable to something else. If you wanted only one copy for a particular job, or a different form type, then you could re-specify the option on the command line, this taking precedence. Note the general rule that for every option which "does" something, there is a corresponding option to "undo" it, to provide for this case.

This may be inconvenient in some cases. If you move to a different directory to do alternative work, you may forget to reset the environment variable to something more appropriate. Or you may log out and come back after lunch and forget to set it again. Hence we provide an alternative mechanism using `.gnuspool` files.

You can plant a `.gnuspool` file in the current directory to imply options for the relevant programs appropriate to files in that directory. The format of the file is similar to setting environment variables, but without the quotes or export, for example:

```
GSPL_PR=-f a4.lh -c3
```

(The file may contain comment lines commencing with `#`. In fact any lines not understood are silently ignored).

As with environment variables, you can override the `.gnuspool` file on the command line.

This technique of planting `.gnuspool` files in directories is appropriate when, as you move from directory to directory, you want to specify options suitable for the files in that directory; e.g. `invoice` form type in one directory, `letterhead` in another etc.

If you want a given option to apply whatever happens, you can put a `.gnuspool` file in your home directory. For example if you want to reorganise the screen of gspl-pq to provide for 6 printers to be displayed at once, rather than specifying `-P6` every time, you can put in your home directory a `.gnuspool` file containing the line:

```
GSPL_PQ=-P6
```

You could instead set an environment variable `GSPL_PQ` in your `.profile` or `.login` file; the choice is up to you.

If you specify options in `.gnuspool` files in your home directory and current directory, and in an environment variable, then they are handled in the following order:

1. Any system-wide defaults are taken (e.g. the user's default form type).
2. The home directory `.gnuspool` file is processed.
3. The environment is processed.
4. The current directory `.gnuspool` file is
5. Options on the command line are processed.

Conflicting options encountered later completely override what came first, so that options specified on the command line will take priority whatever else was encountered. As mentioned above, there is a "reset" type option for every "set" type option, so for example the option `-r` resets the gspl-pr option `-s` to suppress the banner page.

As described below, you can re-assign the functions of the option letters using alternative help message files specified using the `.gnuspool` files.

11.1.1 Environment variable or keyword names

For all programs the environment variable name is the same thing in upper case letters and with non-alphabetic characters replaced by “_” (underscore).

To set the environment variable, the format is just the same as for the options in the corresponding command. If you are using the Bourne or Korn shells, type (for example)

```
$ GSPL_PR='-f letterhead -c2 -wi'
$ GSPL_PQ='-A -P2 -p'
$ export GSPL_PR GSPL_PQ
```

(The quotes are required if spaces are included, which they usually are).

With the C shell type (for example)

```
% setenv GSPL_PR '-f letterhead -c2 -wi'
% setenv GSPL_PQ '-A -P2 -p'
```

There isn't any hard and fast rule about whether to use home or current directory `.gnuspool` files, or environment variables.

In practice people put "comfort" type options, such as help display options and the display of job numbers in the home directory, options specific to files in a given directory, such as form type or retention on the queue, in that directory and transient requirements in the environment. The intention, however, is to give you some flexibility.

Finally note that the search order for options may be changed as described later.

11.2 User reconfiguration

To allow maximum flexibility, all strings, such as screen headers, error and help messages and prompts used in GNUspool are taken from a set of files, and are thus re-definable

a. On a system-wide basis by editing the files.

b. On an individual basis for each user by allowing each user to have access to his or her own version or versions of the files.

All the keystrokes understood by gspl-pq and gspl-user are "soft"; the functions may be redefined as required.

The following are examples of the kind of applications these facilities have:

1. Producing customised versions of the product incorporating site names, help messages etc. on the basic screen formats.
2. Providing "seamless joins" between GNUspool and other software with different function key sets.
3. Providing interfaces appropriate to different terminals, in particular taking advantage of function keys provided on those terminals.
4. Providing support for national languages; possibly with different languages on different terminals on the same machine.

11.2.1 Message files

The standard message files all live in the help files directory, by default `/usr/local/share/gnuspool/help`. The files involved are:

`spuser.help`	The configurable information for gspl-user
`spq.help`	The configurable information for gspl-pq
`splpd.help`	The configurable information for LPD interface programs.
`int-config`	containing the configuration information for the programs internal to GNUspool: (spd, spdinit, spwrite, spshed, xtnetserv).
`xmspq.help`	containing the configurable information for gspl-xpq and gspl-xmpq
`xmspuser.help`	containing the configurable information for gspl-xuser and gspl-xmuser
`rest.help`	Which contains all the configurable information for all the rest of the user-accessible utilities and the CGI programs for the web browser interface (this needs to be available for the remote version of the browser interface).

The files are found by default from `/usr/local/share/gnuspool/help` as specified.

If you want to specify an alternative file, then use the configuration file or environment variable mechanism previously described. The following table lists the environment variables and/or keywords used for various user programs:

Environment variable or keyword	Description
`SPQCONF`	gspl-pq message file
`SPUSERCONF`	gspl-user message file
`SPLPDCONF`	gspl-lpq and gspl-lprm message file
`XMSPQCONF`	gspl-xmpq or gspl-xpq message file.
`XMSPUSERCONF`	gspl-xmuser or gspl-xuser message file.
`SPRESTCONF`	message file for other utilities including CGI programs for Web Browser interface

For example if you had an alternative message file for gspl-pq, you could specify its use by means of the environment variable setting:

```
SPQCONF=`pwd`/my-gspl-pq-file
export SPQCONF
```

It is important to specify the full path name, otherwise the file will be searched for in whatever directory you move to.

Alternatively you can specify the message file in a configuration file (`.gnuspool` file) located in your home directory or in the current directory. If it is located in the home directory, it will apply regardless of what directory you are in, otherwise it will only apply whilst you are in the current directory.

The format of lines in the `.gnuspool` files is similar to that to set environment variables in the shell. For example:

```
SPQCONF=$HOME/lib/myspq.help$TERM
```

Note that environment variable names are also expanded here, so in this example the user is intending to specify a different file according to the setting of the `TERM` environment variable. Environment variables which contain further environment variable names (prefixed with `$`) will be recursively expanded up to 10 times.

There are 3 facilities in the expansion of these lines intended to assist the user to supply defaults etc.:

Firstly, as with the shell, sequences of the form `${VAR-default}` are replaced by the value of environment variable `VAR` if it exists, and otherwise the default string specified.

Secondly the sequence `$0` is replaced by the name (or the last component of the file name) by which the program was invoked, so if you were to put

```
SPRESTCONF=$HOME/lib/helps/$0.help
```

in your `.gnuspool` file, then the file searched for would be `gspl-start.help` if you were running gspl-start, `gspl-qlist.help` if you were running gspl-qlist, etc. You could likewise provide a completely different-looking version of gspl-pq with a different name which automatically selected a different message file.

Finally, if it cannot find the file specified, then an attempt will be made to find a file name constructed by deleting the last part of the file name from the path and substituting the standard name (`spq.help`, `rest.help` etc.), before giving up. For example in the above case, if the files `gspl-start.help` or `gspl-qlist.help` could not be found in the directory `$HOME/lib/helps`, then a further try would be made with the standard name `rest.help` to give the file name `$HOME/lib/helps/rest.help`. (Note that the order of search may be changed, see later).

As for the argument keywords and environment variables, the current directory `.gnuspool` file takes precedence over environment variables, if any, which in turn take precedence over the home directory `.gnuspool` file. However once the keyword is found in one of those places, the remainder are not searched; this means that if a non-existent file is specified, say in the current directory `.gnuspool` file, then the program will abort without looking in the other places.

11.2.2 File format

It is easiest to work on these files by copying one of the supplied ones and starting from there. The files are text, and you can use your favourite editor to amend it.

In general a line which doesn't make sense is ignored. However we recommend that you adopt the convention of using comments denoted by the `#` symbol. Blank lines are ignored everywhere.

Example lines within the files might be:

```
K400:?
1K500:o,*
100P:Ok to delete (Y/N)
E100:Unknown command - expecting job queue control
H400:Please reply Y or N
```

The general form of the lines is:

1. An optional *state code*, denoting the state of the program in which the line applies.
2. A key letter, denoting the function, i.e. key definition, prompt, error or help message etc.
3. A code number for which the program looks when it requires a particular line.
4. A colon preceding the definition of the item.

The state and code numbers are arbitrary and compiled into the relevant programs. Hopefully the context and the comments in the supplied files will make the situations in which they are used clear enough.

The types of entries given by the key letters are as follows:

`K`	a key mapping
`H`	a "help" or similar type message
`E`	an error or other information message
`P`	a "prompt" or other screen field
`N`	a numeric parameter
`A`	an option definition

In addition there are screen heading lines, which have a slightly different format using other letters, defined later.

11.2.2.1 Key definitions

Key definitions are read in when the program is first started. As a result conflicts, where a key is defined for two or more different uses, and other errors are detected before further processing is done.

There are two types of key definition; *global* definitions which apply everywhere within the program, and *local* definitions, which only apply in a particular context, enabling you to use the same key in another context for a different purpose.

A global key definition looks like this:

```
K400:?
```

whilst a local definition looks like this:

```
1K500:o
```

The first number is referred to as a *state code*. The number after the `K` gives the internal code into which the key sequence is translated and used to select the program action.

In all the contexts where state codes apply, there is a help message supplied with the same code as the state code. For example state 1 corresponds to the state where the cursor is in the job queue, keys relevant to this state only are prefixed with this state code, and the help message lines for this state start with `H1:`.

The balance of the key definition gives the character sequences which make up the key definition; these will be translated to the internal code. In the above examples the character `?` is globally translated to the internal code 400 and `o` to 500 in state 1.

You can represent control and non-printing characters using `\` and `^` as described for setup files together with the extensions described below. In particular you must represent comma as `\,` and spaces and tabs as `\s` and `\t` respectively.

To give two or more keys for a given command, separated them by commas, as follows:

```
K400:?,*
```

It is because of this that you must represent comma using `\,` as just mentioned.

Now many keys, particular cursor and function keys, generate multi-character sequences. The sequences can be just written out thus:

```
K406:\e[A
```

The whole of the sequence up to the end or to a comma will be searched for and translated to the given internal code when it occurs. The arrival of the characters will be timed so it should be possible to distinguish between key sequences with similar prefixes (often the case with cursor and function keys) and the effect of typing the component characters separately. This particularly affects the escape character, which on its own is commonly used to abort input, but which also invariably prefixes function and cursor key sequences.

You can if necessary tune the timing of the arrival of characters by means of the environment variable `KEYDELAY`. which is set to the number of 1/10ths of a second to wait between the characters of a suspected function key.

If another character is not received within this time, the character will be assumed to be a single key, otherwise it is taken as the start of a function key. The default for `KEYDELAY` is 3/10ths of a second, which should work correctly for the vast majority of terminals.

If this value is too low, then function keys may not always be recognised, whilst if it is too high, then the response to a single escape character, when required, will appear to be slow.

11.2.2.1.1 Special key sequences.

As well as the control and non-printing characters which may be inserted using the sequences `\n` and `^c` etc., additional constructs introduced by `\k` are provided to import the user's choice of keys for erase, kill etc., and the *termcap* or *terminfo* definitions of some of the special function keys on the terminal.

There are two problems which can arise when you do this:

1. Choices of single-character keys often conflict with existing uses of the same key, but not in a consistent way for each user or terminal. For example:
    a. The character control-U (`^U`) is defined in the supplied files as "scroll-half up". Many people use this as a kill character.
    b. Some terminals have cursor keys which generate a single character. A notable example is Wyse 60 terminals on which the left cursor key generates a backspace (*ctrl-h*) character which therefore cannot be distinguished from a "real" backspace character (or indeed *ctrl* + h).
2. *Termcap* and *terminfo* files often tell lies about what sequences are generated by cursor and function keys. This may not always be the fault of the files themselves if the user has extensively used soft key-setup facilities on the terminal.

If you do successfully get past these problems, then the way to import these values is to use the sequence `\k` followed by the symbolic name of the required item. The items available are:

User defined keystrokes

Symbolic Name	**Meaning**
`\kINTR`	user's interrupt key
`\kKILL`	user's kill key
`\kERASE`	user's erase key
`\kQUIT`	user's quit key

which are looked up on entry to gspl-user or gspl-pq from the *termio* or *stty* settings, as set by stty, and:

Terminal keys

Symbolic Name	**Meaning**

`\kUP`	string sent by up arrow key
`\kDOWN`	string sent by down arrow key
`\kLEFT`	string sent by left arrow key
`\kRIGHT`	string sent by right arrow key
`\kHOME`	string sent by home key
`\kF`*nn*	string sent by function key F*nn*

which are looked up in the *terminfo* or *termcap* database for the terminal type specified in the environment variable `TERM`.

To use these sequences, put the relevant one in place in the key definition for example:

```
K400:?,\kF1
K406:k,\kUP
```

If something does go wrong due to one of the two problems given above, you will find that either gspl-pq or gspl-user does not start, displaying a message such as:

```
State 5 setup error - clash on character 08 with previously-given
value 420 and new value 530.
```

In this case you can pinpoint the problem by looking for a key definition with 420 in (this is the code for erase last character) and with 530 in (left cursor in some fields). This particular message often appears on Wyse 60 terminals where left cursor is the same as backspace, in other words `\kLEFT` and `\kERASE` produce the same result - control-H or hexadecimal 08.

The other thing which might happen is that the function keys do not work properly or the message "Undefined key sequence" is displayed. This is because you have an incorrect *termcap* or *terminfo* file, or because you have reset function keys.

A common problem on VT100-based terminals is that there are four settings depending upon which combination is selected of "normal" or "application" cursor keys and "numeric" or "application" keypad. You can be reasonably sure that your *terminfo* file will assume one of the combinations whilst the terminal will be set to one of the other three.

This particularly affects cursor keys which are often defined in the *terminfo* file to be `\eOA`, `\eOB`, `\eOC` and `\eOD` (application keys) for up, down, right and left cursor respectively whilst the terminal when switched on generates the "normal" cursor keys `\e[A`, `\e[B`, `\e[C` and `\e[D` respectively. If you run into this problem we suggest that you spell out the combinations explicitly, thus for example

```
K406:k,\e[A,\eOA
K407:j,\e[B,\eOB
```

If you do manage to traverse all these snags and catches, we do encourage you to move over to function keys rather than printing characters as printing characters are thereby "lost" for the purpose of typing strings etc.

Don't forget to adjust your help and error messages to reflect the keys you have set up.

11.2.2.1.2 Help and error messages

Help and error messages are treated almost identically. The only difference really is that they are displayed slightly differently.

The text to be displayed is given on as many lines as required with the same prefix, for example:

```
E100:Unknown command - expecting job queue control
E100:Type ? for help
H1:? - help (this file)
H1:^L - redraw screen
```

Now in many cases there are specific variables or strings which it is helpful for the help or error message to quote in the message. These can be inserted using "parameters" introduced by `%` as follows:

Sequence	Meaning
`%%`	"%"
`%P`	program name (last component)
`%F`	full path name of message file
`%p`	process id as a decimal number
`%E`	*errno*, interpreted like perror
`%U`	user name corresponding to effective UID
`%R`	user name corresponding to real UID
`%un`	argument *n* (0<n<9) interpreted as a UID
`%gn`	argument *n* (0<n<9) interpreted as a GID
`%G`	group name corresponding to effective GID
`%H`	group name corresponding to real GID
`%dn`	decimal representation of argument *n* (0<*n*<9)
`%xn`	hex representation of argument *n* (0<*n*<9)
`%on`	octal representation of argument *n* (0<*n*<9)
`%cn`	character (or hex if non-printing) value of argument *n* (0<*n*<9)
`%s`	string argument
`%t`	secondary string argument

Some of the information (program name, process ID, error code, real and effective user & group IDs) is always available. The rest of the information (the 2 string arguments, the 10 parameters) is set up as required for each error message by the calling program. The only way to discover what has been passed is to examine the error message as given in the supplied and to use the same parameters, possibly with a different format.

11.2.2.1.3 Option syntax

There are two alternatives to using the standard single letter command-line options:

1. You can invoke options with keywords such as `+priority` or `--priority` instead of or as single-letter options such as `-p`.

2. Alternative option letters and keywords may be specified by use of different message files.

Option syntax is defined by means of lines of the form

```
A119:p,priority
```

As with other parts of the file, there is an internal state code, in this case 119, to denote the action to take. Alternatives are specified after the colon and separated by commas. They may consist of either:

1. Any single printing character, as for `p` in the above example, which denotes a "`-`" type argument, thus `-p`. The character is taken exactly as given, and can be any printing character (a comma or `\` must be escaped using a `\` thus `\,` and `\\` but these, especially the latter, are very strongly discouraged, for obvious reasons).
2. A keyword consisting of alphanumeric characters, "`-`" and "`_`" (underscore) starting with a letter, denoting a `+keyword` or `--keyword` type argument, thus `+priority` `--priority`. Letters in the keyword are case insensitive.

You do not have to define either single-character or keyword type options for any function. If you leave one out then only the other type will be available, or the option will not be available at all.

If you leave out option syntax definitions out of the file altogether, then a default set of options consisting only of the single-character variants of the options as defined in this manual for each program will be supplied.

If you change the options for gspl-pr, then when you unqueue a job from gspl-pq, the program spjobdump will be run and will pick up message files as for gspl-pr starting with configuration files in the current directory from which gspl-pq was invoked, using the options specified to generate arguments for a gspl-pr command. The same considerations apply to options saved using `+freeze-current` and `+freeze-home` options to various commands.

Of course this means that if you have completely different option definitions in various places, this may be inappropriate and you, and the software, will get totally confused! We recommend some care in using this facility, which was mainly intended for producing national-language versions of GNUspool, with appropriate option names.

11.2.2.1.4 Prompts

Strings introduced with sequences such as `100P:` are *prompts*. These are transient messages which are generated in various places such as the state codes on printers and the messages at the bottom of the screen indicating so many jobs or printers below.

Some prompts have parameters in (such as the "n jobs below" message), which are introduced with a `%` sequence such as `%s` or `%d` very much like the C printf function (which it uses).

A special set of prompts is provided in the internal program file `/usr/local/libexec/int-config`. These provide hexadecimal strings from which the bitmaps for the large characters in the banner page are constructed. These may be adjusted as required, also to provide for additional characters - see the comments in the file.

Another prompt is defined to make standard the initial form type assigned everywhere when GNUspool is first installed. If you should want to change this, please remember that it appears in all 4 files and should be changed consistently; the first program to discover that the User control file `/usr/local/var/gnuspool/spufile23` does not exist will recreate it with this default form type for each user.

11.2.2.1.5 Numeric parameters

There are some sequences of the form

```
1000N1001
```

in the files. These are used in 3 places:

1. In the gspl-pq "job options" screen, to configure the order of prompts and the prompt to start with.
2. In the gspl-user display or set privileges options, to configure the order to present the privileges.
3. In the "save options" screens in both gspl-pq and gspl-user to configure the order in which the prompts are presented.

Please see the comments in the supplied files for how to adjust these parameters.

11.2.2.1.6 Titles

The screen titles used by gspl-pq and gspl-user are handled slightly differently. Although they can be multi-line, each line of the title is taken from a different message number. For example, `D`*n* is the format for the *n*th line of the title for the gspl-pq printers list. (The `D` stands for "devices"). The default is:

```
D1: Printer line form status jobno user
```

Similarly the title for gspl-pq's job list is taken from `J`*n*, the default is:

```
J1:Seq jobno User file form posn size cps pty printer
```

Perhaps you might want to make this more elaborate and incorporate your company name as follows:

```
J1:Bloggs & Sons Ltd -- Print queue --- ? for help
J3:Seq jobno User file form posn size cps pty printer
J5:====================================================
J6:
```

The height of the title is given by the highest-numbered line specified - 6 in the above case (this will give a blank line). Note that lines 2 and 4 will be generated as blank lines also. There is also an optional footer section, using the same principle, with lines denoted by `T`.

The title for the `gspl-user -i` screen is taken from `T`*n* in the same way, and the trailer from `U`*n*.

It is permissible to omit any or all of these titles completely if desired, leaving more space on the screen for jobs and printers, or users, as the case may be.

11.2.2.1.7 Enhancements and line drawing in headers

It is possible to specify screen enhancements and line drawing characters within headers. To do this, the following sequences are interpreted within headers lines.

`\B`
: Set bold mode

`\D`
: Set dim mode

`\F`
: Set flashing mode

`\U`
: Set underlined mode

`\I`
: Set inverse video mode

`\S`
: Set standout mode

The effect of the above sequences are cumulative and apply until the end of the line or until a terminating `\N` to reset enhancements in encountered.

For example the line

```
\IExample\N Header
```

would be displayed as

```
Example Header
```

Do note that not all *terminfo* files define all of the various enhancements, also that "standout" mode is usually some arbitrary combination of the others.

The line-drawing set, if available, is separately invoked by use of sequences thus;

`\L`
: Turn on line drawing mode and display upper left corner line drawing symbol.

`\l`
: Turn on line drawing mode and display lower left corner line drawing symbol.

`\R`
: Turn on line drawing mode and display upper right corner line drawing symbol.

`\r`
: Turn on line drawing mode and display lower right corner line drawing symbol.

`\|`
: Turn on line drawing mode and display vertical edge line drawing symbol.

`\-`
: Turn on line drawing mode and display horizontal edge line drawing symbol.

`\+`
: Turn on line drawing mode and display internal intersection line drawing symbol.

`\<`
: Turn on line drawing mode and display intersection of horizontal line and left edge 'T' line drawing symbol.

`\>`
: Turn on line drawing mode and display intersection of horizontal line and right edge 'T' line drawing symbol.

`\^`

Turn on line drawing mode and display intersection of vertical line and top edge 'T' line drawing symbol.

`\V`

Turn on line drawing mode and display intersection of vertical line and bottom edge 'T' line drawing symbol.

Once line-drawing mode is set, then any of the above characters which immediately follow are interpreted without needing to have a " `\`" in front, so for example

```
\-------
```

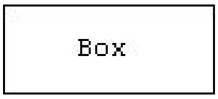
would generate a horizontal line.

Any character not in the above set of line-drawing characters will cancel line-drawing mode. So that you can put a line drawing character in a box next to a line, then the sequence

```
\.
```

cancels line-drawing mode without inserting a period.

For example to produce a display like this;



use the sequence

```
T1: \L---------R
T2: \| Box    \|
T3: \l---------r
```

This applies to all headers of screens in gspl-pq and gspl-user.

If you want an actual "`\`" in a header you should double it thus "`\\`"

11.2.2.2 Changing message files

A few tips may be useful:

- Tabs don't work in messages, since they may displayed anywhere on the screen; use multiple spaces instead.
- Keep the messages reasonably short, this avoids having to redraw large areas of the screen just to display a message and obliterate everything else going on.
- When re-defining keys, don't forget to adjust the help and error messages to correspond, and make sure that they are all consistently re-defined throughout all the states of all the programs.

Please note that there is nothing "magic" about the code numbers for global and local keystrokes; as distributed the state codes 400 upwards are assigned to global keystrokes, and 500 upwards to local ones, but if you want a different help key, for example, in different places by defining `1K400` etc. please go right ahead.

Likewise you can change around, for example, the order of access to screens in gspl-pq, by redefining local versions of the quit key (405) and selection of job and printer screens.

11.2.3 Local help files

The message file facility copes with most of the problems of customisation. One extra facility, however, is available to assist the operator/user in selecting form types and printers in gspl-pq.

Normally, if the operator/user asks for help while filling in a printer or form name, a list of names will be supplied by scanning the appropriate directory (either the *printers* directory for the printers list, or the specific directory in for the form type list for the given printer) and using the names contained therein to format a list of possible names. If this is not useful, either because there are other files being stored in the directory, or because more help is needed, a help file can be provided.

If a file called `-Help` exists in the appropriate directory, this will be assumed to contain a suitably formatted list of printer names and other help information to be output instead of the internally generated list.

11.3 Variation of search order

The order in which `.gnuspool` files are scanned, either to locate message files, or to read options, may be varied from the default if required.

A keyword, optionally placed in the Master configuration file `/usr/local/etc/gnuspool.conf`, in each case may be used to vary this.

11.3.1 Search order for message files

The default search order, for example with gspl-pq, which searches for a file named in the variable `SPQCONF`, is to look:

1. As specified in a `.gnuspool` file in the current directory.
2. As specified in the environment variable of that name.
3. As specified in a `.gnuspool` file in the user's home directory.
4. In a standard place, namely `spq.help` in the system help files directory, by default `/usr/local/share/gnuspool/help`.

This order may be respecified by assigning a "PATH variable" style value to the environment variable `GS_HELPPATH`, consisting of directory names separated by colons. The user's home directory may be denoted by "`~`" and any other user's home directory by "`~user`". Any environment variables are fully expanded.

If a directory name is relative (not starting with "`/`"), it is taken as being relative to the directory from which the program was started. The current directory should be represented as just "`.`".

Finally the environment is represented as a "`!`".

Thus the default search order is represented as:

```
GS_HELPPATH=".:!:~"
```

which causes the search to start in the current directory, then the environment, then in the home directory. Note that there is no way of suppressing the fall-back to the standard location in the system help files directory.

It might be relevant for example, for a French user, with copies of the help files in a separate directory, to set a different search path, thus:

```
GS_HELPPATH=".:!:/usr/spool/xihelp/Francais"
```

This will affect all GNUspool user-level programs, which search for a help file named by a keyword.

11.3.2 Search order for program options

An almost identical system is used for program options, with the keyword `GS_CONFIGPATH` except that the search is in the opposite order. Also the additional symbol "`-`" is used for command-line options.

The default is therefore:

```
GS_CONFIGPATH="~:!:.:-"
```

Indicating the search first in the home directory, then the environment, then the current directory, and finally the command line options.

Note that it is possible to turn off the interpretation of command line options, if required, in this way, by omitting the "`-`", limiting the command line arguments to, for example, just job numbers for gspl-qchange etc.

11.3.3 Freezing options

The `+freeze-home` and `+freeze-current` options in the user level commands and equivalent on-screen facilities in gspl-pq and gspl-xmpq are not affected by these options, the software does not attempt to "backwards-interpret" these paths. In particular please note that these facilities save all of the current values of these options regardless of whether they are default values or where they were read from, so you may want to "prune" the values of the options saved.

12 Extensibility

There are various mechanisms for enhancing or extending the functionality of GNUspool, which go beyond customisation of the user interface. Some of these mechanisms are separate products with their own documentation. These are the Connectivity option, the 'C' programmer's API for Unix and the API for Windows PCs.

This chapter covers the facilities which are built into the basic product and the Motif GUI option as standard. They consist of hooks where GNUspool can invoke custom built user, administrator and internal programs. Such programs are usually shell scripts, but compiled programs can be used just as easily.

12.1 Message Handling

An extra level of indirection exists with message handling (i.e. job completion messages etc by console message or mail), enabling the user to more easily specify alternative message handlers than the mail program, or console write program, and/or alternative message files to display messages in a foreign language.

It was previously possible only to change the message content or format "system-wide" and to reassign the mailer program etc by specifying variables such as `MAILER` in `/usr/local/etc/gnuspool.conf`.

This mechanism still works as before, only the printer name is inserted where appropriate. To do this, the scheduler now invokes an intermediate program spmdisp to generate and distribute the messages. Knowing which user was responsible for the job, spmdisp can be told to select an alternative message file for the given user by searching the user's home directory for a file `.gnuspool` containing an assignment to the variable `SYSMESG`.

```
SYSMESG=/my/message/file
```

The message file should be based upon the contents of `/usr/local/share/gnuspool/help/int-config`, although probably only the messages about job completion etc are necessary.

At the same time, the .gnuspool file in the user's home directory may contain any of three additional lines as follows:

```
MAILER=/alternative/mail/program
WRITER=/alternative/console/write/program
DOSWRITER=/alternative/dos/write/program
```

(The "programs" may alternatively be shell scripts). The only argument passed to the mail or Unix write program is the user name. The DOS write program is passed the host name to send to and the user name. The programs are run under the identify of the originating user.

In addition to the arguments, the message generated from the message file, plus any error log reports, is sent on standard input, and the following environment variables (as similar as possible to those supplied to filter programs etc by spd) are passed.

Variable	Meaning
`SPOOLJOB`	Job number

`SPOOLHDR`	Job title
`SPOOLUSER`	Numeric user id
`SPOOLJUNAME`	User name submitting job
`SPOOLPUNAME`	User name job "posted to"
`SPOOLFORM`	Form type
`SPOOLFLAGS`	Post-processing flags
`SPOOLPTR`	Printer name printed on
`SPOOLHOST`	Host name on which printed

The only difference between this and spd is an "orthogonal" one; `SPOOLHOST` gives the host name on which the job was printed on here, and spd sets it to the host name the job came from.

The variables `SPOOLHDR`, `SPOOLPTR` and `SPOOLHOST` are not set if there is no job title, or the message does not involve a printer, or does not involve a printer on a remote host respectively.

12.2 Customising printer error handling

It is possible to insert a custom-defined action to be automatically invoked whenever a printer local to the host goes into error or offline.

This is done by putting a reference to (usually) a shell script in the master configuration file, `/etc/gnuspool.conf`, with the keyword `SPPTRMSGS`, for example:

```
SPPTRMSGS=/usr/local/libexec/ptr_alert
```

The command is invoked with 3 arguments:

1. Name of printer.
2. Device name.
3. Description of printer (as a single argument).

12.3 Macros for Interactive User Programs

Macro command facilities are available in gspl-pq, gspl-user, gspl-xmpq and gspl-xmuser.

They are also available in gspl-xpq but on a per-user basis and can be set up interactively and saved to the user's `.gnuspool` file.

Up to 9 macro commands relating to jobs and printers in gspl-pq and gspl-xmpq, can be set up. Similarly up to 9 macro commands can be defined for users in gspl-user and gspl-xmuser. In addition you can prompt for any arbitrary command in all of those places. The command doesn't have to be anything to do with the context of jobs, printers, variables or users. It will, however, always be passed the identity of the currently-selected item, or user as a parameter or parameters.

The actual commands are placed in the relevant help files, which by default are `spq.help`, `spuser.help` etc. in `/usr/local/share/gnuspool/help`.

12.3.1 Inserting the commands.

Firstly you must put the commands in the format:

```
27000P:Prompt for prompted command
27001P:cmd1
27002P:cmd2
27003P:cmd3
27010P:Prompt for prompted command
27011P:cmd1
27012P:cmd2
```

The `2700nP` prompts are the commands for fixed command n, n 1 to 9 for jobs in gspl-pq, gspl-xmpq, users in gspl-user and gspl-xmuser, the 0th entry being the prompt for the prompted-for command. The `2701nP` prompts are the commands for fixed command n, n 1 to 9.

The command involved is any arbitrary shell command which may take a list of jobs, printers, variables or users as the case may be. (The command may he given zero users by gspl-xmuser and gspl-user if none are selected).

The commands should be preceded by "`!`" where appropriate to indicate that they may produce output, and the screen should be refreshed when they are completed.

12.3.2 Binding the keys

This only applies to gspl-pq and gspl-user. Menu entries are automatically created in gspl-xmpq and gspl-xmuser. The names given to the pulldown entries can be set by editing the lines:

```
gspl-xmpq*menubar*macro1.labelString:
```

etc in the resources file.

To bind the keys, you need to assign key codes 600 to 609 in `spq.help` etc, for example:

```
K600:0
K601:\kF1
```

would assign the "0" key to the prompted-for command and F1 to "fixed" command 1.

12.3.3 Example

You want to provide 3 extra function keys in gspl-pq.

1. Reset the number of copies on a given print job to 3 by pressing F1.
2. "Hold" a print job by setting the printer name to "hold" by pressing F2.
3. "Release" a print job by setting the printer name to blank by pressing F3.

First we create 3 shell scripts, which must be in every user's path, these are `copies3`, `holdprint` and `relprint` and are one-liners as follows:

```
copies3: gspl-qchange -c3 $1
holdprint: gspl-qchange -Phold $1
```

```
relprint: gspl-qchange -P- $1
```

Next insert the following lines in `spq.help`

```
K601:\kF1
K602:\kF2
K603:\kF3
```

(you might want to make the key definitions specific to state 1 by giving them as `1K601:\kF1` etc)

```
27001P:copies3
27002P:holdprint
27003P:relprint
```

To do the same thing with gspl-xmpq you just have to add the last 3 lines to `xmspq.help`. Specify the full pathname to copies3 etc.

13 The LPD Interface

13.1 Introduction

Many external systems and terminal servers expect to send and receive data using the "LPD protocol". This is a method of submitting files for printing across a TCP connection to a spooler on a remote machine.

The LPD interface provides two complementary functions:

1. It provides for GNUspool to be able to send outgoing spool jobs, converting them into LPD-style jobs, and also provides emulation of the list and remove functions. This is provided for by means of the xtlpc program and associated programs gspl-lpq and gspl-lprm.
2. It provides for GNUspool to receive incoming spool jobs and convert them into GNUspool jobs, possibly sending back completion information. In addition, emulation of the list and remove functions of the LPD interface is provided. This is provided for by means of the xtlpd program and associated files.

The interfaces are intended to be flexible and easily tuned to the user's requirements in respect of the relationship between the different concepts of form types and printers as understood by GNUspool and LPD.

13.2 Limitations

The LPD interface is very different from GNUspool. Behind it lies another spooler. This means that all that the interface can do is pass data from one to another. "Printing" means that the files are being passed across. The LPD interface does not tell GNUspool about offline printers, page counts in jobs, which copy out of several has been reached, not does it expect to receive such information from GNUspool.

If you want to have the level of control which GNUspool normally offers, then this really is not the way to do it. If you want to achieve this and you can allow printers to be directly addressed without any further spooling or buffering, then the performance and facilities thereby offered will be vastly better.

13.3 Outline of LPD Protocol

The details of the LPD protocol are defined in the various RFCs. A more detailed description than that given here is given in the reference section of this document.

Spool jobs are made to consist of two files;

1. The D-file, containing the data to be printed.
2. The C-file, consisting of textual commands, one per line, indicating various information about the user etc and indicating how the corresponding D-file is to be printed.

On top of that, there is a protocol for sending either type of file between machines, and for transmitting listing and deletion requests.

It is expected that a terminal server or computer using the LPD protocol will convert the print request into appropriate C and D files in a spool area, transmit them to the receiving machine, which will itself buffer the files again before interpreting the C file to cause the document to be printed.

13.4 Form types

The LPD protocol provides for form types denoted only by a single letter, whereas GNUspool provides for a multiplicity of multiple letter form types. One of the objectives of the LPD interface is to provide for a flexible mechanism to translate between the two styles.

In each case the translation is specified by a control file which may be edited to suit the installation.

13.5 Using the XTLPC program

Xtlpc is run as a command, typically from the `network=` line of a printer setup file. We suggest that you make the outgoing host name or IP address the parameter in the `<net>` field of the GNUspool printer list.

xtlpc takes the following options:

Option	Meaning
`-f <ctrlfile>`	Specify location of control file
`-H <hostname>`	Specify destination host or IP address
`-N`	Forbid zero-length files
`-S <hostname>`	Specify source host or IP address if not that given by the hostname call.
`-P <printername>`	Override printer name in protocol, necessary for some servers which require a particular name to select the appropriate printer

Please note that the first two options, "-f" and "-H" must always be specified. The "-N" option is provided because some LPD servers treat zero-length spool files as an error.

The following environment variables are expected to be provided to the interface. GNUspool normally supplies these to any command run from the network= line.

Environment variable	Meaning
`SPOOLPTR`	Name of printer
`SPOOLDEV`	Contents of <net> field in printer list
`SPOOLHDR`	Job title
`SPOOLJUNAME`	User name of owner of job
`SPOOLPUNAME`	"Post to" user name of job
`SPOOLFORM`	Form type
`SPOOLJOB`	Job number
`SPOOLCPS`	Number of copies

To use this from within GNUspool, it should be invoked as follows from the printer setup file:

```
network=/usr/local/libexec/xtlpc -f /usr/local/share/xtlpc-ctrl -H $SPOOLDEV -N
```

However you might wish to test this from the command line by setting appropriate values to the above environment variables and running the command (without the `network=` of course), supplying a small file as input.

We also recommend that you put the following keywords in your setup file:

```
onecopy
fberror
```

The first causes only one spool job to be generated when multiple copies are printed; the number of copies is passed on. The second feeds back error reports on standard error so that any errors are displayed on gspl-pq etc screens.

13.6 Description of LPD Protocol

Each LPD job consists of one or more D-files of job data, and a C-file. This consists of lines of text referred to as "control cards". The first character of the control card gives the action, such as "`f`" to print the file name following, "`H`" to set a host name etc.

The following is an example of a C-file:

Control Card	**Explanation**
`Hxisl.co.uk`	Set host name to xisl.co.uk
`Pjmc`	Set user's login name
`Jpasswd`	Set job name
`Cxisl.co.uk`	Set class name
`Ljmc`	Set user's name for banner
`fdfA058xisl.co.uk`	Print file name given
`UdfA058xisl.co.uk`	Unlink (delete) file name
`N/etc/passwd`	Name of file

Some of these constructs do not have direct equivalences in GNUspool, for example the file name is not relevant, and the class name does not mean the same thing.

The form type (or equivalent) is selected by substituting some other letter for "`f`" in the print command. Copies are specified by duplicating the print command the appropriate number of times. There are options for indentation, and the automatic invocation of "`pr`" with appropriate arguments.

Each file is sent across by means of a protocol of lines of text including a single-byte code denoting the action, and the name and size of the file being sent.

13.7 Format of XTLPC control file

The control file specifies 3 types of operation (apart from comments introduced with `#`).

1. Assignments, which are implemented immediately.
2. Control-card specifications, introduced by a "`:`", which specify output to the

C-file, with appropriate variable substitutions.

3. Protocol specifications, introduced by `<`, `>`, `<` and `*`, which specify control lines to be sent to the remote lpd

There are options to make the first two of these conditional or repeated.

Prior to executing the control file, the environment is read in and treated as a series of assignments. Many of these assignments may be irrelevant, but particularly important is many of the variables set up by GNUspool and in particular:

`SPOOLHDR`	job title
`SPOOLJUNAME`	owner of job
`SPOOLPUNAME`	posting user
`SPOOLCPS`	number of copies
`SPOOLHOST`	host owning job
`SPOOLJOB`	job number
`SPOOLFORM`	form type
`SPOOLPTR`	printer name

In addition, the following variables are set up:

`MYHOST`	host name being run on
`DESTHOST`	host name spooled to as in -H option

The following variables should be assigned to in the control file:

`XTLPCSPOOL`	Spooling directory
`PORTNAME`	Port name to use
`CFILE`	Format string for C-files
`DFILE`	Format string for D-files

The value assigned to `XTLPCSPOOL` should be an absolute pathname, writeable by user gnuspool.

The formats of `CFILE` and `DFILE` are special, and take the following formats:

```
CFILE=cf%c%.3dxisl.co.uk
```

The `%c` and `%.3d` parameters are used to generate unique names in the spool directory, using a similar method to lpr. A value of '`A`' is substituted for the `%c` and a new number between 1 and 999 for `%.3d` the latter incremented by one and the former incremented as far as '`Z`' until a unique number is generated.

13.7.1 C-File control cards

These are introduced by "`:`" and cause the following string of text to be output, with variable names substituted.

13.7.2 Protocol descriptions

The line

```
^B%$CFILE $CFILE^J
```

is interpreted by expanding the variable names, and substituting the appropriate control character for `^A`, `%B` etc (`^J` is newline), and the size of the file specified by `%$CFILE`.

The line

```
< ^@
```

is interpreted by expanding the line if necessary and attempting to read in the character requested. If it matches, then the next protocol entry is taken, otherwise the process is aborted. (In all cases the null character `^@` is expected).

The line

```
* $CFILE
```

causes the specified file to be sent literally down the socket.

13.7.3 Conditions

Conditions may be applied to C-file specifications and assignments.

Conditions take 5 basic types:

1. Execute remainder of line if specified variable is defined and non-null.
2. Execute remainder of line if specified variable equals a given string.
3. Execute remainder of line if specified variable does not equal a given string.
4. Execute remainder of line if specified variable pattern-matches a string.
5. Execute remainder of line if specified variable does not pattern-match a string.

Examples of each of these are as follows:

```
{ SPOOLHDR } : J$SPOOLHDR
{ SPOOLFORM == a4 } Form = f
{ SPOOLFORM != labels } Form=f
{ SPOOLFORM ~ *.ps } Form=c
{ SPOOLFORM !~ *.pr } Form=d
```

You can use this to set the variable `Form` to the appropriate control letter to insert later into a C-file control card, as in `: $Form$DFILE`

13.7.4 Repetitions

LPD makes multiple copies by repeating the relevant control card the appropriate number of times, whereas with GNUspool the number of copies is just an attribute of the job.

In principle, there is no reason to disturb this, each copy generated by GNUspool will become a distinct lpd job.

A provision is made for conversion, however. GNUspool passes the `SPOOLCPS` environment variable, giving the number of copies to print (this counts down with each copy).

If `onecopy` is included in the printer setup file, only one copy is generated, with `SPOOLCPS` set to the number of copies. To produce multiple copies of the "control card", this is put inside ()s in front, thus:

```
(SPOOLCPS): $Form$DFILE
```

In this way the command C-file card is output the number of times given by the numeric value of `SPOOLCPS`.

13.7.5 Example control file.

```
# Control file for lpd client
# We must define a spool directory and a port number
# or port name (printer in /etc/services is standard)

XTLPCSPOOL=/var/spool/outlpd
PORTNAME=printer

# We must define the format of the C-file and D-file.
# The %.3d gets filled in in the style of printf.

CFILE=cf%c%.3d$MYHOST
DFILE=df%c%.3d$MYHOST

# We start off by defining the form type by pattern
# matching the requested form type.
# first default if all else fails

Form=f

{ SPOOLFORM ~ *.ps }Form=l
{ SPOOLFORM ~ *.pr }Form=p

# If SPOOLHOST is defined and sensible, use that as host name,
# otherwise stick with MYHOST which gets set up before we start.

Host=$MYHOST
{ SPOOLHOST } Host = $SPOOLHOST

# Next we lay out the C-file
# Start off with standard stuff

:H$MYHOST
:P$SPOOLJUNAME
:J$SPOOLHDR
:C$Host

# Insert pr title if Form type p

{ Form == p } :T$SPOOLHDR

# Insert for each copy.....
```

```
( SPOOLCPS ) :$Form$DFILE

# Insert delete command

:U$DFILE

# Insert "N" line

:N$SPOOLHDR

# And now for the protocol to send it
# > means send something
# < means receive something
# * means send whole file

# First say we've got a job for the printer

> ^B$SPOOLPTR^J

# Wait for acknowledgement and then send D-file

< ^@
> ^C%$DFILE $DFILE^J
< ^@
* $DFILE
> ^@

# Wait for acknowledgement and then send C-file

< ^@
> ^B%$CFILE $CFILE^J
< ^@
* $CFILE
> ^@
< ^@
```

13.8 Listing and removing jobs with the LPD interface

Two additional user programs are provided with xtlpc, gspl-lpq, which lists jobs on the remote queue, and gspl-lprm which removes jobs, after the fashion of lpq and lprm.

Two internal programs, xtlpq and xtlprm provide support for these facilities.

An additional help file, `splpd.help` provides messages for gspl-lpq and gspl-lprm.

Communication with the remote queue is handled by xtlprm and xtlpq. gspl-lprm and gspl-lpq interrogate the printer setup files and supply the arguments to xtlprm and xtlpq respectively.

Although the intention is to interface with a remote lpd server, printers on any other GNUspool host, including the host on which the commands are run, will operate with identical effects provided that xtlpd is running on the relevant host. In this way usage is transparent regardless of whether the printer represents a remote lpd server or not.

Some lpd servers do not support the "list" and "remove" options for lpd. In such cases there is no way of reproducing the output. Also note that the format, in particular that of the list command, is entirely in the hands of the remote lpd server. When the emulation is in effect using xtlpd, the output is generated via the commands shortlist and longlist which may be tailored as desired.

13.8.1 Listing Jobs

The command gspl-lpq takes the following syntax:

```
gspl-lpq [-l] [-s] printer
```

The argument `-l` denotes "long" style listing. This may be cancelled by a `-s` option, and options supplied in `.gnuspool` files and the environment by means of the keyword `SPLPQ` The printer argument is used to determine which host etc is required and xtlpq invoked to obtain the listing output from the lpd server.

The keyword options `+long-list` (or `--long-list`) and `+short-list` (or `--short-list`) are available as alternatives for `-l` and `-s` respectively.

The printer name is that understood by GNUspool; this may be translated to a different printer name understood by the LPD server as below.

13.8.2 Removing Jobs

The command gspl-lprm takes the following syntax:

```
gspl-lprm printer job1 [job2 job3 ... jobn]
```

The printer and job arguments are transmitted to the remote lpd server. Any output from the command is returned to the user.

The printer name is that understood by GNUspool; this may be translated to a different printer name understood by the LPD server as below.

13.8.3 Hostname Details

The programs gspl-lpq and gspl-lprm use the following method to obtain the hostname and other information to be passed in the requests to the remote server.

If the printer is a local printer, or not a local network printer (with `network` in the setup file) then it is interpreted using the local hostname and the printer name specified.

Otherwise the printer setup file for the printer and formtype currently specified are examined, and certain keywords or an `xtlpc` command is searched for. If the network command specified appears to refer to a shell script, then that may be searched further for an `xtlpc` command. The control file for xtlpc is also parsed looking for various assignments to `PORTNAME` or `SPOOLPTR`.

The parameters searched for are the destination hostname, printer name, port number or name and the sending host `SPOOLHOST`, for the benefit of systems with multiple IP addresses.

A reasonably brave effort is made to trace through shell scripts, but it is not omniscient.

13.9 Receiving incoming jobs with XTLPD

Xtlpd runs as a "daemon" process, to receive incoming jobs using the LPD protocol and submit GNUspool jobs from them.

A script, by default `xtlpd-ctrl`, in `/usr/local/share`, specifies the translation from LPD-style control cards to GNUspool job parameters, together with implementations of the long and short listing and remove commands.

To start xtlpd, issue the following command, possibly substituting your own control file for the argument.

```
/usr/local/libexec/xtlpd /usr/local/share/xtlpd-ctrl
```

This should start as a "daemon process" and return control to the command prompt.

You may wish to add this line to the system startup files to start up GNUspool when the system is booted.

13.10 Format of XTLPD control file

The control file consists of 2 basic types of line apart from comments, introduced by a # sign.

1. Variable assignments, looking very much like shell variable assignments, which are performed as the file is read. Some assignments provide additional options to the process, such as log file generation and debugging. Assignments have 3 variants, =, += and =+ which replace, append after and prepend before any existing values respectively.
2. Specifiers, which are "remembered" and interpreted when the time comes to read a C-file.

The following are examples of assignments:

```
LOGFILE=xtlpd-log
DEBUG=2
Prcmd = "pr -f -l66"
```

The following are examples of specifiers:

```
S<s>:
H<s>: sprflags += " -o <s>"
W<n>: propts += " -w<n>"
f<f>*: | gspl-pr $sprflags $cpsflags -f $Form
p<f>*: | $Prcmd $propts | gspl-pr $sprflags $cpsflags -f $Form
```

The first case describes the handling of an "`S`" control card. The rest of the line is a string, which we choose to ignore.

The next case describes the handling of an "`H`" control card. In this case we do an assignment, to append the given string to the variable sprflags. The `<s>` in the right hand side of the assignment is replaced by whatever string followed the "`H`" on the control card.

Next we describe how to handle a "`W`" control card. The `<n>` denotes that a number is expected, and is inserted in the right hand side of the assignment.

The next two lines give two alternative file handling commands. The `<f>` denotes that a file name is specified. The "`*`" indicates that the control card may be repeated to denote the number of copies required (we explain how this is handled shortly).

The pipe symbol "`|`" denotes that we now want to run a command with the relevant file or files "piped into" it. The various variables are substituted first. The file name is either the name read from the control card, or a list of space-separated file names taken from a variable name given in parentheses.

There are also two internal commands are `<indent>` and `<unlink>`. These are available to implement the `I` and `U` control cards respectively.

There may be more than one specifier for each control letter. Each is treated in the order given.

13.10.1 Assignment format

As previously stated, assignments may take the forms:

```
Variable = String
Variable += String
Variable =+ String
```

These commands assign a value to the variable, appends the string to the current value and pre-pends the string to the current value respectively.

The string may be "as is" (leading spaces are deleted), or delimited by single or double quotes. It may also, except in an assignment in a specifier, be a shell command enclosed in backquotes with the same meaning as with the shell. `$variable` constructs are expanded except where the string has internal quotes or characters escaped with a `\`. Note that, unlike the shell, this applies regardless of whatever kind of quotes the string has around it. The quotes just specify the limits of the string.

Further examples of assignments might be:

```
Form = "-f "
Form += `gspl-user -d|grep Default|sed -e 's/Default form \(.*\)
$/\1/'`
```

13.10.2 System Variables

The following are so-called "system variables", which the Control file may, and in some cases must specify.

Variable	**Meaning**
`XTLPDSPOOL`	Spooler directory. This must be specified.
`PORTNAME`	Name or number of tcp port name to use. This must be specified.
`LOGFILE`	Append messages to specified log file (relative to the spooler directory). If omitted, messages appear on standard error.
`DEBUG`	Debugging level. 0 = none, 1 = major ops, 2+ all operations.
`SHORTLIST`	(More or less compulsory). Shell command to execute to generate short listing of jobs.

`LONGLIST`	(More or less compulsory). Shell command to execute to generate long listing of jobs.
`REMOVE`	(More or less compulsory). Shell command to execute to delete jobs.

There are additional system variables which are redefined at the start of each job.

Variable	**Meaning**
`HOSTNAME`	Host name which the job came from.
`PRINTER`	Printer name which lpr thinks it's talking to.
`PERSON`	(In remove command) the "person" parameter
`CMDLINE`	(In list and remove commands) the users and jobs list.

The last three of these are "exported" to the environment.

Note that the host name assigned to `HOSTNAME` may not necessarily be the same as on the control cards, as the former comes from the `/etc/hosts file` on which xtlpd is run, and the latter from that from which the job originated. The host names which gspl-pr and other GNUspool programs use come from the file `/etc/gnuspool.hostss` and may be different again.

13.10.3 Handling Repeats

The C-files in the LPD protocol specify multiple copies by just repeating the f commands the requisite number of times. It that is good enough, and you want a separate GNUspool job for each repeat, then forget this section.

It is more efficient, however, to provide for the "`-cn`" parameter to gspl-pr to be specified. To do this what we do is to say that the specifier is repeatable, by use of the "`*`", thus

```
f<s>*: . . .
```

Next we specify a repeat specifier. This looks like this:

```
*<n>:cpsflags =-c <n>
```

The effect of this is that whenever a control card is repeated, the repeat specifier is executed first, and then the control card specifier is executed once. The `<n>` is of course replaced by the number of copies involved.

A refinement of this allows you to distinguish between repeats of more than one copy, and provide for code to be executed then:

```
*<n>: cpsflags=
*<n>*: cpsflags =-c <n>
```

The second "`*`" signifies work to be done when the repeat is more than one. The first line is thus always obeyed, setting `$cpsflags` to blank, and the second is only obeyed when the copies is greater than 1.

If you want to specify actions to take place at the end of the block of repeats, you should use the special "`.`" command thus:

```
.<n>:cpsflags=
```

13.10.4 Beginning and Ending actions

You may want to specify actions to take place at the beginning and end of each C-file, as several C-files may be passed across in each transmission and you may want to reset variables at the start, or to defer commands until the end.

The following specifiers provide for this:

```
!: . . .
@: . . .
```

Specify actions at the beginning and end respectively.

Again, multiple occurrences of these may appear and are treated in order.

13.10.5 Internal commands

Two internal commands are provided:

- `<indent>` followed by a number or <n> provides for indentation as lpr expects by prepending spaces to each line (empty lines do not have any spaces added). The number of spaces can be fixed, in which case a number should follow, or as read from the control card.
- `<unlink>` is used to implement the unlink command. This is followed by the file names to unlink.

Examples of these are:

```
I<n>:<indent><n>
U<f>:<unlink><f>
```

In the last case you might want to be sure of doing the "unlinks" only at the end thus:

```
U<f>:unlinks += " <f>"
@: <unlink> $unlinks
```

13.10.6 Example Control File

```
# Define variables used to configure the daemon process

PORTNAME=printer
XTLPDSPOOL=/var/spool/incoming-jobs
LOGFILE=xtlpd-log
DEBUG=2

# Commands to invoke to produce short and long lists

SHORTLIST=/usr/local/libexec/shortlist
LONGLIST=/usr/local/libexec/longlist
REMOVE=/usr/local/libexec/remove

# Initialise "pr" command options

Prcmd = "pr -f -l66"
```

```
# Get default form type

Form = `gspl-user -d|grep 'Default form'|sed -e 's/^.* \([^ ]*\)
$/\1/'`

# Control character interpretation
# First give things we want to initialise
# Note use of special gspl-pr options -E and -o, and further down -U

!: sprflags=-E BSD -o $HOSTNAME
!: propts=
!: cpsflags=
!: <indent>0
!: plainlist=

S<s>:
J<s>: header = "-h '<s>'"
C<s>:
L<s>: sprflags += " -U <s>"
T<s>: propts += " -h '<s>'"
H<s>:
P<s>:
W<n>: propts += " -w<n>"
I<n>: <indent><n>
1<s>:
2<s>:
3<s>:
4<s>:
N<s>: header = "-h '<s>'"
M<s>: sprflags += " -m"

# Pre-repeatable.
# Flush out any pending files from the last command
# We do it this way to absorb Ns which come after the files.

*<n>: (plainlist) | gspl-pr $sprflags $cpsflags $header -f $Form
*<n>: (prlist) | $Prcmd $propts | gspl-pr $sprflags $cpsflags -f $Form
*<n>: (pslist) | gspl-pr $sprflags $cpsflags $header -f $Form.ps
*<n>: (trofflist) | gspl-pr $sprflags $cpsflags $header -f $Form.troff
*<n>: (ditrofflist) | gspl-pr $sprflags $cpsflags $header -f
$Form.ditroff
*<n>: (dvilist) | gspl-pr $sprflags $cpsflags $header -f $Form.dvi
*<n>: (plotlist) | gspl-pr $sprflags $cpsflags $header -f $Form.plot
*<n>: (rasterlist) | gspl-pr $sprflags $cpsflags $header -f
$Form.raster
*<n>: (cifplotlist) | gspl-pr $sprflags $cpsflags $header -f
$Form.cifplot
*<n>: plainlist =
*<n>: prlist =
*<n>: pslist =
*<n>: trofflist =
*<n>: ditrofflist =
*<n>: dvilist =
*<n>: plotlist =
*<n>: rasterlist =
*<n>: cifplotlist =

# The next line sets the copies number to default
# The line after is only executed if the number of copies > 1
# and sets to the number of copies

*<n>: cpsflags =
*<n>*:cpsflags = " -c <n>"

# Repeatable commands

f<f>*:plainlist += " <f>"
p<f>*:prlist += " <f>"
l<f>*:pslist += " <f>"
t<f>*:trofflist += " <f>"
n<f>*:ditrofflist += " <f>"
d<f>*:dvilist += " <f>"
g<f>*:plotlist += " <f>"
v<f>*:rasterlist += " <f>"
c<f>*:cifplotlist += " <f>"
U<f>: unlinks += " <f>"

# End of file, flush out any pending files and unlink

@: (plainlist) | gspl-pr $sprflags $cpsflags $header -f $Form
@: (prlist) | $Prcmd $propts | gspl-pr $sprflags $cpsflags -f $Form
@: (pslist) | gspl-pr $sprflags $cpsflags $header -f $Form.ps
@: (trofflist) | gspl-pr $sprflags $cpsflags $header -f $Form.troff
@: (ditrofflist) | gspl-pr $sprflags $cpsflags $header -f
$Form.ditroff
@: (dvilist) | gspl-pr $sprflags $cpsflags $header -f $Form.dvi
@: (plotlist) | gspl-pr $sprflags $cpsflags $header -f $Form.plot
@: (rasterlist) | gspl-pr $sprflags $cpsflags $header -f $Form.raster
@: (cifplotlist) | gspl-pr $sprflags $cpsflags $header -f
$Form.cifplot
@: <unlink> $unlinks
```

13.10.7 Example short and long list commands

The following shell/awk scripts implement the short and long list commands referred to in the above example control file.

13.10.7.1 Lpq short format

```
# Turn gspl-qlist output into equivalent of lpq short format output

qlim=${PRINTER--}
count=1
```

```
gspl-qlist -z -q $qlim -l -F '%u %N %K %h' | awk '{
if (count == 0)
print "Rank  Owner Job  Files  Total Size"
count++
if (int(count/10)%10 == 1) th = "th"
else if (count%10 == 1) th = "st"
else if (count%10 == 2) th = "nd"
else if (count%10 == 3) th = "rd"
else th = "th"
num = sprintf("%d%s", count, th)
printf("%-7s%-11s%-5s%-38s%s bytes\n", num, $1, $2, $4, $3);

END { if (count == 0)
print "no entries"
}'
```

13.10.7.2 Lpq long format

```
# Turn gspl-qlist output into equivalent of lpq long format output

qlim=${PRINTER--}
count=1
gspl-qlist -z -q $qlim -l -F '%u %N %K %c %o %h' | awk '{
count++
if (int(count/10)%10 == 1) th = "th"
else if (count%10 == 1) th = "st"
else if (count%10 == 2) th = "nd"
else if (count%10 == 3) th = "rd"
else th = "th"
num = sprintf("%d%s", count, th)
printf("%s: %-35s[job %s%s]\n", $1, num, $2, $5)
if ($4 != 1)
num = sprintf("%-3d copies of %s", $4, $6)
else
num = $6
printf("  %-33s%s bytes\n\n", num, $3)
}
END { if (count == 0)
print "no entries"
}'
```

13.10.8 External Queue Specification

Three additional options are used in the the gspl-pr command to support xtlpd. Exceptionally, these may only be used by `gnuspool` and `root` regardless of the permissions setting via gspl-user. Attempts by other users to access them will be silently ignored, as will invalid parameters.

The option `-En` will set an external queue number of n. The queue number may be given as a number, or a name as specified in the external queue definition file `/usr/local/etc/gnuspool.ext`.

The option `-Uuser` overrides the authorship of the job, setting it as though it had been queued by the given user. (This is in distinction to the option `-uuser` which sets the user name on the banner page, but leaves the owner of the job unaffected).

The option `-ohostname` overrides the "originating host" name of the job, as though it had been submitted from the given host.

The purpose of these options is to enable gspl-pr to be run via daemon processes with the parameters set appropriately. None of these options are intended to be invoked by general users.

The external queue number has no effect other than changing the disposition of messages. There are currently no options to display this, although the number can be obtained from the field `apispq_extrn` in the job structure of the API.

13.10.9 External queue definition file

This file is a text file, with `#` comments, consisting of 4 fields per line.

1. An external queue number, 1 to 65535.
2. A name, e.g. BSD, Novell.
3. A "mailer" program or shell script.
4. A "writer" program or shell script.

The mailer or writer programs are invoked in circumstances where mail or write messages are called for. The arguments passed are the user name and the originating host name, together with the environment variables as specified for spmdisp.

The two programs should be given as absolute path names, or relative to `$SPROGDIR` (which defaults to `/usr/local/libexec/gnuspool`).

Either or both programs may be omitted, or given as "`-`" to completely suppress the relevant messages.