

xisoftware

Xi-Text Release 6
Installation and Administration Guide

Table of Contents

1	Introduction	5
1.1	Documentation Standards and Program Usage	6
1.2	Command Line Program Options	6
2	Installation	7
2.1	Starting installation	8
2.2	Choosing the user program directory	8
2.3	Installing the system user	9
2.3.1	Creating user name manually	10
2.4	Selecting installation options	10
2.4.1	Network licence	10
2.4.2	Selecting the directories to be used	11
2.4.3	Final check prior to installation	11
2.4.4	CUPSPY	13
2.4.5	Installation of printers	13
2.4.6	Replacement of lp	13
2.4.7	Completion of installation	13
3	Defining and installing printers	14
3.1	Definition of printers	15
3.2	The standard formtype	15
3.3	Running Ptrinstall	16
3.4	Cloning printers	17
3.5	Hardware interfaces	17
3.6	Emulations	17
3.7	Old tools PTRDEFINE, PTRINSTALL, PTRUNINST and PTRUNDEF	18
4	Licence issues	19
4.1	Checking the status of a licence	19
4.2	Running xt-vwrite	20
4.3	Obtaining the licence directly from the fileserver	21
4.4	Emergency Licensing — xt-emerglic	21
5	Setting up networking	22
5.1	Editing the host file	22
5.1.1	Setting up the local address directly	23
5.1.2	Setting up the local address from a website	24

5.1.3	Setting up the local address from a connected host	24
5.1.4	Adding a host	25
5.1.5	Adding a fixed IP address Windows host	26
5.1.6	Adding a DHCP client	27
5.1.7	Quitting and saving	27
5.1.8	Editing the hosts file after installation	27
5.2	User map file	28
6	Overview of Xi-Text architecture	29
6.1	Daemon processes	29
6.2	System directories	29
6.3	IPC Facilities	30
6.4	Files used by Xi-Text	31
6.4.1	Internal spool files	31
6.4.2	Help and Message files	32
6.4.3	Printer Definitions and Setup Files	33
6.4.4	Configuration files	34
6.4.5	Configuration files held in /etc	34
6.4.5.1	Xi-Text Hosts and Clients File	34
6.4.5.2	Xi-Text Master Configuration File	34
6.5	Ports and Network Services	35
7	User administration	36
7.1	Adding New Users	37
7.2	Removing Users	37
7.3	Setting Privileges	37
7.4	Setting Default & User Priorities	38
7.5	Setting Classcodes	39
8	Other administration matters	40
8.1	Startup and shutdown of Xi-Text	40
8.1.1	spstart	40
8.1.2	sstop	40
8.2	spconn and spdisconn	41
8.3	ssuspend and srelease	41
8.4	Startup on boot	41
8.4.1	Starting all printers on boot	41
8.4.2	Cleaning and subdividing the spool directory	41
9	Backup of Xi-Text System	42
9.1	Jobs	42
9.2	Printers	43
9.3	User permissions	43
9.4	Upgrading from Earlier Releases of Xi-Text	43
9.4.1	Help and resource files	44
9.4.2	Carrying Information to the New Installations	44
10	Print jobs and form types	45

10.1 Print Jobs	45
10.2 Formtypes (Paper Types & Suffices)	45
11 Configuration options	47
11.1 Default Options	48
11.1.1 Setting Up Defaults	48
11.2 Setting views of the Job and Printer lists	49
11.2.1 Selecting Jobs and Printers by Printer Name	49
11.2.2 Selecting Jobs by Owner	50
11.2.3 Selecting Jobs by Title	50
11.2.4 A real example	51
11.3 Job and Printer list Formats	52
11.3.1 Adding New fields	56
11.4 Help & Error Messages	56
11.5 Keys & Commands	57
11.5.1 Specifying Different Keys	57
11.5.2 Disabling Commands	58
11.5.3 Customising Commands	58
12 Extensions to the toolkit	59
12.1 Printer Error Handling	59
12.2 Scripts	60
12.2.1 Printer Log Files	60
12.2.1.1 Viewing a Log File	60
12.2.1.2 Archiving / Purging Log Files	61
12.3 Macros	62
12.3.1 Publishing Documents for Printing on Demand	62
12.3.1.1 Enhancements	63
13 Advice & Trouble Shooting	65
13.1 Files & Directories	66
13.1.1 User Program Directory	66
13.1.2 The spd Internal Directory	67
13.1.3 The progs Internal Directory	68
13.1.4 The printers Internal Directory	70
13.2 IPC Facilities	70
13.2.1 Looking at IPC Facilities	70
13.2.2 Problem - Cannot Start Xi-Text	71
13.2.3 Deleting IPC entries owned by spooler	72
13.2.4 Processes	73
13.3 Messages about key clashes entering spq or spuser	73
13.4 Warning messages about unknown key name	74

Chapter 1

Introduction

Xi-Text 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
- A GTK+ Interface
- A PyQt interface
- An X-Windows Motif Toolkit Interface
- An API for use with C and C++
- An Interface for MS-Windows
- 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 **Xi-Text**
- Reference manual — a complete description of all components of the basic product.
- Administrator Guide — this manual. 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

Xi Software manuals use 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 **Xi-Text** or standard Unix facilities

Warnings and important advice

1.2 Command Line Program Options

Almost all of the programs that make up **Xi-Text** 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

```
spr -c 3 -f invoices
```

or it may be left out as in

```
spr -c3 -finvoices
```

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

```
spr -mwsv
```

or separated, as in

```
spr -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

```
spr +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.

Chapter 2

Installation

A simple installation procedure is adopted for **Xi-Text** using the Perl scripting language which is now distributed as standard on most UNIX systems and all Linux systems.

Some of the installation binaries which are generally useful, such as the `xt-cjlist` and `xt-cplist` utilities, are now installed together with the user-level binaries, prefixed by “`xt-`”. (The prefix is to avoid any confusion with similarly-named utilities for other Xi Software products).

The product uses some configuration files in the `/etc` directory, together with 5 other directories:

1. The spool directory, in which jobs and their attributes are saved, together with logfiles, pending messages etc. This can in principle get large.
2. The printers directory, in which control files and scripts for various printers are held in subdirectories of the same name as the printer (possibly with symbolic links for similar printers).
3. The internal programs directory.
4. Help files directory for files with help and error messages, menus etc.
5. User binaries directory, where the programs used directly by users are located.

In most installations the internal programs and help messages directory are the same, but they don't have to be. However the other directories should all be distinct.

If all the default options are selected, then the locations of the directories are as follows:

Spool directory	<code>/usr/spool/spd</code>
Printers directory	<code>/usr/spool/printers</code>
Internal programs directory	<code>/usr/spool/progs</code>
Help files directory	<code>/usr/spool/progs</code>
User binaries	<code>/usr/local/bin</code> if it exists otherwise <code>/usr/bin</code>

It is common to make the first four directories subdirectories of a base directory, in the default case `/usr/spool`, but this is by no means compulsory.

The user binaries should be put in a directory accessed by the `PATH` environment variable of all users of the product. On many systems `/usr/local/bin` is suitable for putting binaries local to the system not installed with the system, hence this is the default.

To install **Xi-Text**, we suggest you copy and extract the media to a suitable directory. The product binaries are located in a subdirectory given by the name of the platform you are using, for example `Linux-mint13_64`.

2.1 Starting installation

Within the product directory will be a file called `INSTALL`, which you can run (as `root`) by just typing:

```
./INSTALL
```

(You probably need the “./” in front as the `root` user will probably have the current directory excluded from the search path).

The following style of output should appear:

```
Welcome, Acme Systems, to Xi Software's distribution.

We intend to install the following products intended for the
platform Linux (Intel Architecture) running Mint Rel 13 (64 bit).

Xi-Text Release: 23.485 Serial: 123456
Xi-Util utilities Release: 1.62 Serial: N/A

Before we go any further, please check that the above is correct.
OK to Continue?
```

Please check this carefully to ensure that the platform and versions are what you expect before proceeding. If you have any doubts, please reply `N` and contact us.

2.2 Choosing the user program directory

If you do reply `Y`, you should get the following:

```
I need to ask you which directory you want to put user-level
binaries for Xi Software products in.

This should normally be the PATH for each user.

Here are the directories on your PATH (but note
this is the root path and may have unusual directories in).

/bin
/sbin
```

```

/usr/bin
/usr/local/bin
/usr/local/sbin
/usr/sbin

```

Directory to use:

At this point you should type in the directory to use.

If this does not exist or is not on the `PATH`, this will be queried. If you do choose a directory which is not on the `PATH`, then you will be reminded in various places. Versions of operating systems differ as to the appropriate place to set this up, so this must be left to the user.

2.3 Installing the system user

After selecting the user program directory, you should receive the following message.

```

Xi-Batch installation
=====

```

```

Welcome to Xi-Batch installation.

```

```

Note that in most places you can press F1 or ? to get help,
F2 or ENTER to continue (and accept default answer)
F8 or Q (capital) to abort.

```

```

[Please press ENTER to continue]

```

Most of the operations from this point are wholly interactive, where options such as directory names are offered, you can type space or tab to cycle through alternatives. In the case of directory names tab cycles through options at the same level of directory, space at lower levels. If an option appears in brackets, normally to a "yes or no", question pressing Enter will select this. Other questions offer the default or current setting where the user types. To accept this, just press Enter.

We now move on to installation of the system user, `spooler`, unless this has already been done, with the message

```

Installation of user spooler
-----

```

```

I need to install user "spooler".

```

```

This will own the files and directories used by Xi-Text spooler.

```

```

OK to continue?

```

Most internal files, and IPC facilities such as shared memory segments and semaphores, are owned by a user `spooler`. This user name is like a "sub-super-user" for **Xi-Text** facilities. It is not intended that the `spooler` user name should be used extensively, however it (hopefully) serves to distinguish **Xi-Text** files from other files.

The only files which are not owned by `spooler` are the scheduler process `spsched` and the network server process `xtnetserv`, which have to be owned by `root`, as they have to be able to transform to other users.

Before proceeding with the installation, it will be necessary to create user `spooler`, if it does not exist.

2.3.1 Creating user name manually

If there is a “create user” program on your system which the installation routine understands, it will be used, otherwise the installation will abort so you can do this yourself.

If you do have to create it manually, we normally recommend a user id of 50 (i.e. in the "system users" below 100), and your "create new user" routine may need a special option to install this. It is not entirely necessary, however, but we do not suggest that all the standard user-level login directory and contents be included, as the user name is not intended for general use.

When setting up the default group for user `spooler`, we suggest that a group such as `lp`, `daemon`, or `sys` is selected. This only matters, however, if the web browser interfaces to **Xi-Text** and **Xi-Batch** are to be used on the same machine, when the user names `batch` and `spooler` should have the same default group.

2.4 Selecting installation options

After checking and if necessary creating the `spooler` user, the next message you should receive is:

```
Installation Requirements
=====
```

```
Do you want a full licence? [Yes]
```

If you are just making a 30-day trial, then reply `n` to this. Otherwise, if this is to be a production copy, reply `y` (or just press Enter to get the default of Yes).

If you say that you want a full licence, you should then get the question

```
Do you want to try to download the codes? [No]
```

It is possible to connect directly to the Xi Software website and download a licence key without typing codes in. However it may not be allowed in your installation if it is not connected to the Internet or some firewall is in place. Say `y` if you want to try to do this (it won't be catastrophic if it fails, you'll just have to get the codes and type them in).

2.4.1 Network licence

Xi-Text can run “standalone”, or it can share jobs and printers with one or more other Unix or Linux computers. If you are doing this, you need a “network licence” to enable this facility. It also requires certain service names to be added to the system and a list of host names to be set up.

You also need a network licence if you are planning to use the MS Windows client interface, the API (Unix or Windows) or to use the remote version of the browser (which uses the API). You do not need it if all you are doing is printing from Windows Clients.

You will be asked the question:

```
Do you need a network licence? [Yes]
```

Reply either *y* or *n* as required. Do not specify this if you don't need it. I leaves TCP ports open which if not used are a potential security risk in addition to consuming resources.

2.4.2 Selecting the directories to be used

The next question concerns the directories to be used:

```
Installation directories
=====
```

The installation directories are currently set up as follows.

SHLIBDIR	Location for shared libraries	/usr/lib/xi
INCPATH	Include directory for API header	/usr/include/xi
USERPATH	Location for user programs	/usr/local/bin
SPROGDIR	Location of internal programs, data	/usr/spool/progs
SPHELPDIR	Location of help files	/usr/spool/progs
SPOOLDIR	Spool directory for pending print jobs	/usr/spool/spd
SPOOLPT	Location of printer setup files	/usr/spool/printers

```
Do you need to change any of those [No]?
```

If you opt to change any of the directories, you will be taken through a menu of options to do this, with short cuts for where you want all the parts placed in a common directory.

2.4.3 Final check prior to installation

After you have finished selecting the directories, you will get the message something like the following

```
Summary
=====
```

```
Just to summarise the options:
You are installing a full licence
You are going to try to download the codes from the Xi Software Website
You are enabling full networking.
```

```
Installation directories are as follows
```

SHLIBDIR	Location for shared libraries	/usr/lib/xi
INCPATH	Include directory for API header	/usr/include/xi
USERPATH	Location for user programs	/usr/local/bin NB not in PATH
SPROGDIR	Location of internal programs, data	/usr/spool/progs

```
SPHELPPDIR Location of help files          /usr/spool/progs
SPOOLDIR   Spool directory for pending print jobs /usr/spool/spd
SPOOLPT    Location of printer setup files      /usr/spool/printers
```

```
Are you happy with those options [Yes]?
```

If you agree to these, it will ask again if you are sure, and then install the software.

Finally, if you have not agreed to download the codes (or if it tried and could not), it will ask

```
In a moment you will be asked for some codes.
```

```
You may have been given them, or you may want to
fetch them from the Xi Software Website at
http://www.xisl.com/onlinlic.php
```

```
The time and date on your machine is set to 15:04 on 10/08/2012
If this is correct, please press RETURN, otherwise interrupt and reset:
```

You should check the time and date are correct, abort and restart if necessary, and then press ENTER to be asked for your name.

```
Please type your Organisation name.
This should be exactly as you would like it to appear on
your output. (1 line, Max 80 chars)
```

You should key in your organisation name as you want it to appear later. You should then be asked for your serial number (it will suggest the one it thinks is correct) thus

```
Organisation: Acme Systems
Serial number: 12346
```

```
Is that ok?
```

If that is OK, then you will be prompted for the codes if it not downloaded.

If the codes are successfully entered or downloaded, then it should say

```
Your copy is licensed from 10/08/2012 without limit
Network version (if applicable)
Licence OK
spstart - restarting Xi-Text scheduler process
```

Before finishing, it will attempt to install startup files in the appropriate places before asking

```
Do you want to be a Xi-Text super-user?
```

This is whether you want the the user you logged in as (if you did not log in as `root` and did `su` or `sudo`) to be given all possible permissions. You probably will want to do this.

At the conclusion it should say

```
Xi-Text is now running!
=====
```

```
Xi-Text is now available for use.
```

```
Looking at system startup files.....
```

```
System startup file exists - replace it? Yes
```

2.4.4 CUPSPY

We provide an interface which emulates CUPS for programs which expect to run it and route data to printers under the control of **Xi-Text**.

If this is installed, any printers you subsequently install will be installed into the CUPSPY data files.

If you do install this you will be invited to confirm that you want to stop CUPS if it is running.

2.4.5 Installation of printers

After completion of installation of the program files, you will be invited to add or update the installed printers controlled by **Xi-Text**. You can do this now if you wish, or you can do this later.

The procedure is described in the next section.

2.4.6 Replacement of lp

On System V-style systems, you will be invited to replace the `lp` client programs with **Xi-Text** equivalents which take the same options.

If you so this, the original `lp` commands will be saved with the `.OLD` suffix.

2.4.7 Completion of installation

When you reach the end of the installation procedure, you should receive the message

```
It looks like everything was installed correctly.  
We wish you the greatest of success with our products.
```

```
Yours sincerely
```

```
Xi Software Ltd.
```

Chapter 3

Defining and installing printers

In order to use **Xi-Text** for actual printing, printers have to be *defined* and then *installed*.

Defining a printer includes:

- Specifying the hardware interface details, Serial, Parallel, USB, Network and network protocol used by the printer.
- Specifying the emulation, i.e. the control language used by the printer.
- Specifying options like turning on or off headers permanently.
- Providing conditional variations such as double-sided, landscape etc under the control of "form type suffixes".

Installing a printer involves matching the definition to a physical device and putting the printer online for use by **Xi-Text**. If the copy of **Xi-Text** is networked, the printer will become immediately available for use by other computers running **Xi-Text**.

In previous versions of **Xi-Text**, the two steps were performed manually using `PTRDEFINE` then `PTRINSTALL`.

We now provide one extensive Perl script for defining printers, installing and de-installing them, also updating CUPSPY files for CUPS emulation. The script is called `Ptrinstall`, and is in the user programs directory, however it must be run by `root`.

Note that whilst a wide range of hardware types and emulations are supported, some cases will require custom management. Some lesser-used features of **Xi-Text**, such as alignment pages and record files are not supported by `Ptrinstall`. These will have to be hand-edited later.

Also please note that if you upgrade the copy of Xi-Text to be networked and you want the existing printers to be "visible" on the network, you must turn off the "local only" flag which will be set on all printers for a non-networked copy. You can do this by de-installing and re-installing the printer.

After installing the printer, the printer must be set running to start jobs being printed. `Ptrinstall` can do this, or this can be done on other hosts on the network, or via the windows interface etc.

Note that **Xi-Text** is assumed to be running by most of `Ptrinstall`, however it is possible to define new printers if it is not running.

`Ptrinstall` contains extensive help facilities and we aim to continually enhance and improve it, so its details are not set out extensively here, just explain some of the major points.

3.1 Definition of printers

When a printer is defined, we do the following:

1. We think of a unique name for **Xi-Text** to refer to it.
2. We define how the computer talks to it hardware-wise
3. We define the control language – material other than text, commonly PCL or PostScript.
4. We set up spooling options selectable by suffix, or site-specific options for printing header pages or otherwise.

(One option allows a "clone" of an existing printer to be created, similar to the existing printer but with a different name and accessed using a different IP address or device name).

The printer installation routines create a subdirectory in the printer directory, by default located in `/usr/spool/printers`, with the same name as the printer, so for example if the printer is called `print10`, the directory containing setup files etc for the printer will be `/usr/spool/printers/print10`.

Two files are created by `Ptrinstall` in this directory these are

1. `.device`, which holds information relating to the hardware interface parameters.
2. `default`, which hold emulation and spooling parameters.

In fact, these files are concatenated together in order, so in principle the data could be mixed, however this is not recommended, especially as a different file from `default` can be selected for a different form type and the `[.device]` is always read, so it should contain data independent of the form type.

Do remember that different form types may be handled differently. When a job is printed, the a form type such as `a4.ps` is split into the "paper type", in this case `a4` and the "suffix", in this case `ps`.

The actual file read (apart from the `.device` file, if it exists, which is always read first) is a file, if it exists, of the same name as the paper type (`a4` in the above example), or failing that the `default` file. It is a mistake not to have at least one of those form files, either that of the same name as the paper type, or the `default` file.

3.2 The standard formtype

When **Xi-Text** is first installed, each user is given a default form type of `standard`. There is nothing magic or special about this name, it is a placeholder for users to change to their own in-house default form types, such as a well-known form type like `a4` or `letter`, or some in-house name they might want.

We do suggest this is changed, and `Ptrinstall` offers to change it if it finds the form type is set to `standard`. Early on you may get the message:

```
Standard form type
-----
```

We think this is probably the first time you have run this.

When Xi-Text is installed the users are set up with a "standard" as the default form type.

You probably don't want this, but probably some standard name for the size, such as "a4" or "letter", or some other name like "plain" of "80gsm".

You probably want to set all other users to use this by default.
Do you want to change this [Yes]?

We suggest that you do agree to change this, either to a standard form name like `a4` or `letter`, or to some in-house name. The `Ptrinstall` script will step you through this.

3.3 Running Ptrinstall

To define and/or install printers, run the script `Ptrinstall` from the command line as `root`, for example

```
# /usr/local/bin/Ptrinstall
```

In this example the user path directory was not in `root`'s path, so the full name had to be quoted and the script will say:

```
***Warning: User Path /usr/local/bin not in PATH - adding it temporarily
```

Note that the program is completely interactive. Whilst ENTER will usually terminate an entry or accept the current one, several characters are interpreted immediately.

<code>y</code>	Reply yes to question
<code>n</code>	Reply no to question
<code>F1</code> or <code>?</code>	Ask for help
<code>F2</code>	Alternative to ENTER
<code>ESC</code>	Abort the current level
<code>F8</code>	Abort completely
space or tab	Choose next option

There is a distinction between space and tab when selecting directories. TAB suggests alternatives at the same level of the tree, whilst space suggests alternatives at the next level down.

Don't use the F8 for abort unless you have to, in case you are left with half-written files.

3.4 Cloning printers

When you run `Ptrinstall`, you will be asked early on if you want to make a printer a "clone" of an existing printer.

This is implemented by means of a symbolic link.

Only make a printer a clone of another if it is exactly the same interface and emulation of the printer being cloned, with exception of the device or IP address used. Changes to the main printer parameters will of course affect the clone.

There is an option to "break" the clone and make the cloned printer a separately-defined printer in its own right.

3.5 Hardware interfaces

We currently support the following interfaces with `Ptrinstall`. If you have other ones you may need a "filter" type program to send data to it using a customised `.device` file.

If you do edit the `.device` file, don't forget to change the interface type to `Custom` at the top of the file to avoid `Ptrinstall` mistakenly trying to parse or rewrite it.

Paralell	The original "Centronics" parallel port
Serial	The original RS232 serial port with baud rates and parity
USB	USB ports
Network	Various kinds of TCP network with a variety of protocols

Under the category of "network" printers, `Ptrinstall` supports the following TCP protocols.

LPD	The original LPD protocol with a "C-file" and a "D-file" and handshakes between blocks
Telnet	So-called "Reverse Telnet" protocol with no handshakes
FTP	FTP protocol – files are printed by copying a file to a printer as if it were being uploaded to an FTP server
HPNPF	Similar to "Reverse Telnet" except that the printer status is interrogated after every block of data using SNMP requests

In most cases the LPD protocol is sufficient and is recommended, however the HPNPF protocol provides more extensive feedback of error conditions which can be displayed on various status indications, such as the `spq` screen.

3.6 Emulations

We currently support the following printer emulations with `Ptrinstall`. If you have other ones you may need to create your own `default` or other paper-type file, perhaps based on one of the standard ones.

If you do edit the `default` file, don't forget to change the emulation type to `Custom` at the top of the file to avoid `Ptrinstall` mistakenly trying to parse or rewrite it.

Plain	No changes - data taken as plain text
Epson	Epson-style printer and escape sequences
IBM	IBM Proprinter emulation
PCL	Hp LaserJet emulation
PS	PostScript printer
PCL/PS	Printer accepts either PCL or PostScript data and prints accordingly or printer is a PCL printer and <code>Ghostscript</code> is being used to translate to PCL. This is recommended where possible as it provides the most extensive facilities
PJL	Printer accepts PJL (Printer Job Language) commands to select either PCL or PostScript data under the control of the formtype suffix

We try to implement a consistent scheme for "form type suffixes" selecting the handling of printed data.

3.7 Old tools `PTRDEFINE`, `PTRINSTALL`, `PTRUNINST` and `PTRUNDEF`

The tools `PTRDEFINE`, `PTRINSTALL`, `PTRUNINST` and `PTRUNDEF` have been replaced in newer versions of **Xi-Text** with one tool named `Ptrinstall` which performs all of the operations of the old tools in a more intuitive way. Especially, it combines the common case of defining then immediately installing a printer into one operation, by offering to do the installation after you've defined the printer.

Chapter 3 describes how to use `Ptrinstall` and starts on page 14.

Chapter 4

Licence issues

We have a licensing system partly to regulate use of the product, partly to enable or disable features, notably networking, which customers may or may not need and partly as a check on invalid installations.

Please note that the licence incorporates the date, which may be changed if the product is recovered from backup.

When the product is first installed, it often has a trial licence valid for a number of days, typically 30 days.

This can be extended any number of times, or a full, indefinite licence installed.

The installation script takes you through the initial licence. For further licences you will need to run the program `xt-vwrite`, which can be found amid the user-level binaries. However it should only be run by `root`.

You may find two other programs here useful:

Program	Use
<code>xt-checklic</code>	Check and report licence status
<code>xt-emerglic</code>	Emergency licence

4.1 Checking the status of a licence

You can run the `xb-checklic` program at any time to check the status of the licence.

If valid, it should say something like:

```
Licensed to:      Xi Software Ltd
Serial is:        1
Start time:       24/01/2005
End time:         No limit
Validated for networks
```

The last line will be omitted if it is not a networked licence.

If the licence is not valid, it will say something like

```
Checksum fail calc = 3cacfe92 read = 1dc5d31a
```

and return a non-zero exit code.

Xi-Text can still be running when this command is run.

4.2 Running xt-vwrite

`xt-vwrite` is used to install a new licence. It usually accepts a key supplied by us. You will need to get the validation code from us, either by calling us or accessing the web page www.xisl.com/online-licencing, clicking where indicated. Optionally it can try to download a licence

To run `xt-vwrite`, **Xi-Text** must be stopped. You can normally run it from the command line with no arguments:

```
xt-vwrite
```

The first thing it will do is display something like:

```
The time and date on your machine is set to 11:59 on 17/05/2003
If this is correct, please press RETURN, otherwise interrupt and reset:
```

Please do check that the time and date are set correctly as the codes are date-sensitive.

```
Please type your Organisation name.
This should be exactly as you would like it to appear on
your output. (1 line, Max 80 chars)
```

Here you should enter your organisation name as you want it displayed in certain places (such as on the default banner page). You should then be asked:

```
Organisation: Acme Solutions
Serial number: 566
Is that ok?
```

The serial number displayed is incorporated in the distribution files, but it is possible that there may be a discrepancy between that and the version issued. If there is, type `n` and re-enter your organisation name and the correct serial number.

You should then be asked:

```
Please enter the security code given to you by your supplier.
It is only valid for a very limited period so please obtain it afresh
if necessary.
>>>>>>
```

The validation code is 4 blocks of 8 hexadecimal digits (every fourth digit is a modulo-17 check digit, so a "G" is possible in those positions) which may be entered using upper or lower case letters when requested. If you make a mistake in your typing, the location of your error will probably be pointed out to you.

The validation code given incorporates the network enabling and the duration of the licence.

4.3 Obtaining the licence directly from the fileserver

If you can access our fileserver directly over the Internet, you may be able to install the licence without any key typing. This can only be done if you have a registered customer code and serial number allocated.

To attempt this, run `xt-vwrite` with two arguments, one of which indicates whether a network licence is required:

To install a non-networked licence, use the command:

```
xt-vwrite -I -L
```

To install a networked licence, use the command

```
xt-vwrite -I -N
```

The “-I” argument selects the Internet access to get the key, and the “-L” or “-N” select the non-networked or networked licence respectively.

If the attempt to fetch the licence key fails, `xt-vwrite` will drop back to prompting for it.

4.4 Emergency Licensing — `xt-emerglic`

Occasionally the licence codes (validation keys) can become corrupted, for example, following a restore operation on the main disc drive and it is not possible to install a new licence quickly.

The command `xt-emerglic` is provided which will licence the software for 7 days. This will allow the software to run allowing time to arrange for a permanent licence. A warning message is displayed when various programs, such as `spq`, are started.

To install an emergency licence, run `xt-emerglic` thus:

```
xt-emerglic -L
```

for a non-networked licence or

```
xt-emerglic -N
```

for a networked licence.

Without the argument, `xt-emerglic` will guess as to the likely requirement and prompt for confirmation.

Chapter 5

Setting up networking

Xi-Text can run with jobs and printers shared between 2 or more Unix hosts, possibly different platforms. Additionally, there is the MS Windows interface and the API.

To use any or all of these things, **Xi-Text** must be run in network mode.

Three things have to be done:

1. Some standard "service" names need to be inserted in the `/etc/services` file.
2. Hosts may need to be configured in a hosts file `/etc/Xitext-hosts`.
3. The product needs to be given a "network licence".

When you install **Xi-Text** with a network licence using the standard install script, these items are all attended to.

If you have already installed a non-network licence, you may find it easiest to just re-install by running the installation script, but answering the question differently.

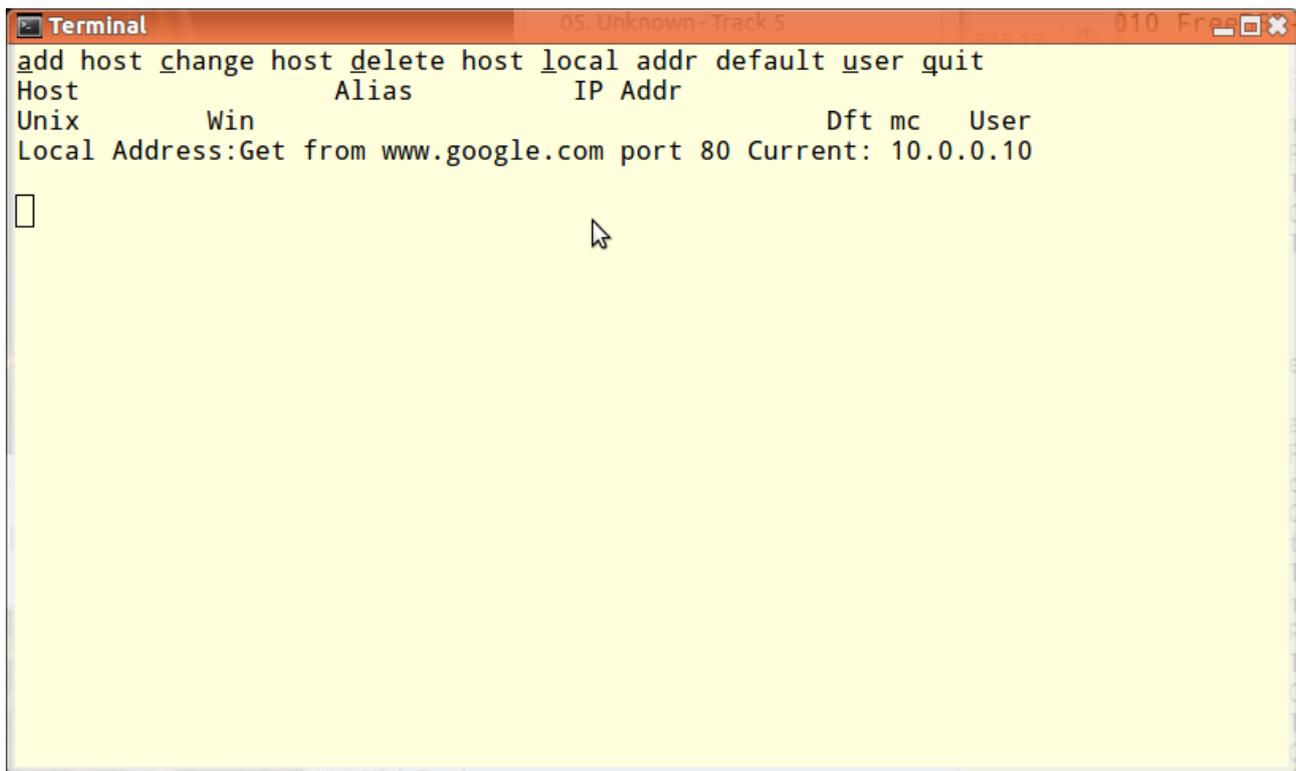
The services will be set up appropriately, and a network licence installed.

Recent versions of **Xi-Text** are initialised so as to pick up the local address automatically. Other hosts may be connected "manually" using `spconn`, but you may want to set up aliases and hosts to be connected automatically at start up. There are also facilities for setting up Windows clients.

5.1 Editing the host file

Unlike earlier versions of **Xi-Text**, you probably only need to edit the host file if:

1. You need to set up a "local address", the IP address by which the host you are installing on is known to the outside world. This is needed as jobs and variables are identified partly by that IP address.
2. You want to give aliases to other hosts you inter-connect with.
3. You want to denote some fixed IP addresses as clients only, not expecting to receive information about jobs and variables (typically for Windows clients).



```

Terminal
05 Unknown - Track 5
010 Fre
add host _change host _delete host _local addr default _user quit
Host      Alias      IP Addr
Unix      Win
Local Address: Get from www.google.com port 80 Current: 10.0.0.10

```

The former use, of declaring mappings from Windows user names to UNIX user names, is still supported, but deprecated in favour of the use of the user map file for this purpose.

A special host editor program `hostedit` (plus a GUI version of it `xhostedit`) is available to edit the host file. This currently still understands and maintains deprecated facilities. These are now put with the other user-level binaries.

To run this enter:

```
hostedit -I /etc/Xitext-hosts
```

The “-I” denotes that the file should be edited in place.

With a new installation, the display will take the following form:

The installation script sets up the hosts file to obtain the local IP address by connecting to `www.google.com`

5.1.1 Setting up the local address directly

Without a hosts file present, **Xi-Text** will try to discover the local address by applying `gethostbyname` to the result of `gethostname`. However this often yields a “loopback” IP such as `127.0.0.1` rather than something identified to other hosts.

To overcome this, you can set up in the host file:

1. An explicit IP address or host name to use or

2. A host name or IP address to connect to (usually using the HTTP port) and get the local IP address from `getsockname`.

The latter (more recently introduced) is better, especially on hosts which may have a variable DHCP-assigned IP address.

To set up a fixed local address, type `l`.

If there is a local address you will be asked:

```
Local address set, do you want to unset it? [N]
```

Type `y` to unset this prior to setting a new one by typing `l` again.

If there is no local address set, or you have just unset it, you should get the message

```
Do you want to set a local address? [N]
```

Type `y` and you will get the prompt

```
Local address:
```

Now you can type the host name or IP address to use.

5.1.2 Setting up the local address from a website

Rather than typing in a local address, you can obtain the local address by connecting to a website and noting the local address.

You can do this once, or you can do it every time (this is best if the IP address may change, due to it being set by DHCP).

Type `w` to select this option, and receive the message:

```
Use Google? [N]
```

If you reply `N` to this, you will be prompted for another host. If you reply `Y`, then www.google.com will be used.

In either event (after checking that the host is valid and has an HTTP server on port 80), you will get the prompt:

```
Always get host that way? [Y]
```

If you reply `Y` to this, it will set up to always obtain the local address from that host when **Xi-Text** starts, otherwise it will just set the local address IP as if it had been explicitly entered.

5.1.3 Setting up the local address from a connected host

If you have already set up a local host, you can use one of those to set the local address and possibly always do so by moving the cursor to it and typing `L`.

A variety of possible ports and services are tried and the first one to which a connection can be made is used.

Again you will be asked if you always want to get the host that way.

5.1.4 Adding a host

You should add a host:

1. If you want to give a name to an IP address by which the host is referred to.
2. If you want to give a more convenient name to a host rather than a fully-qualified domain name.
3. If you want to arrange for the host to be automatically connected on startup.
4. If you want to say that a given host is only to be used for clients.

We don't recommend that you set up Windows client user names any more, instead use the User Map file.

Press "a" to add a host, you should get the prompt:

```
Is new entry a Unix host(Y) or Client(N)? [Y]
```

Just press ENTER. Next you should get the prompt.

```
Unix host name:
```

Type the name of the host, e.g. on our site, one is called "jessie". You should then get the prompt:

```
Any alias for jessie:
```

Unless you require an alias, just press ENTER. Next you will get the prompt:

```
Probe (check alive) before connecting? [Y]
```

Again press ENTER (this checks that the host is on-line before attempting a TCP connection, however in some cases a Firewall etc prevents UDP messages from getting through and this must be avoided).

```
Trust host with user information? [Y]
```

The function of this has now been deprecated, just accept it.

```
Manual Connections only? [N]
```

This concerns whether you want the connection to the other Unix host to be attempted as soon as **Xi-Text** is started, or manually by means of `spconn`. Reply `y` if you require this.

```
Default timeout value OK? [Y]
```

Press ENTER unless you want a variation in the connection timeout. The display should now look like this:

This should be repeated for each host (and should also be repeated on each host so configured). To make any amendments, move the cursor to the relevant host name and type `c` or to delete it type `d`.

```

Terminal | 2012-08-10 14:02:08 To = From
add host change host delete host local addr default user quit
Host      Alias      IP Addr
Unix      Win
Local Address: Get from www.google.com port 80 Current: 10.0.0.10
jessie    10.0.0.2   probe     trusted

```

5.1.5 Adding a fixed IP address Windows host

Press `a` to add a host name, and this time type `n` to indicate that a Windows host is being added. The prompts are as follows:

```
Specific Windows Host (Y) or `roaming user' (N)? [Y]
```

Press ENTER to denote a specific IP address.

```
Windows client host name:
```

Type the name of the hosts, e.g. "kim".

```
Any alias for kim:
```

Again, press ENTER unless you want to give an alias.

```
Default user:
```

This is prompting for a default user name to be used for access — the normal UNIX user of that PC, although other users can use it, specifying a password. This can be omitted if required.

```
Password-check user(s)? [N]
```

This can be turned on to force password-checks on all users if required.

```
Default timeout value OK? [Y]
```

Again, this can be left as the default or not (it is slightly more important, as if the password checks are enabled, it gives the time after which the user will have to log in again with his password if he has not done anything in the meantime).

5.1.6 Adding a DHCP client

Please note that this has been deprecated in favour of using a user mapping file, but is included for compatibility with previous versions. If both are specified, the hosts file, which is read second, will take priority.

This is where the IP may change each time and the recognition is by the user name (or possibly a Windows user name matched to a Unix user name).

Again, press *a* to add a host name and *n* to indicate that a Windows host is being added, and now type *n* again to select DHCP clients.

```
Unix user name:
```

This is the user id under which activities will be run on the Unix host. Be careful to make the case of characters correct (usually all lower case).

```
Windows user name:
```

This is the windows user name, or blank if the same as the Unix user name. It is not case-sensitive.

```
Do you want to specify a default machine name? [N]
```

This enables the user to specify the (Unix) host name of the usual PC at which the user works. He or she will be able to "log in" at other hosts with the password.

```
Password-check user(s)? [N]
```

You can set this to insist on a password in all cases.

```
Default timeout value OK? [Y]
```

Again this gives the timeout value. After this time of inactivity, the user will have to log in again.

5.1.7 Quitting and saving

Type *q* to quit and save the newly-created hosts file in `/etc/Xitext-hosts`.

5.1.8 Editing the hosts file after installation

The utility `hostedit` (part of the standard utilities) may be used to edit the hosts file.

We recommend that **Xi-Text** be halted and restarted after this has been completed.

To invoke it, use the following command:

```
hostedit -I /etc/Xitext-hosts
```

The `-I` argument denotes that the file is to be edited in place.

5.2 User map file

The user map file provides a mapping between external names, usually Windows user names, and UNIX names.

The file is in `/etc/xi-user.map`, and consists (apart from comments introduced by the `#` character) of lines of the format

```
unix-user:windows-user
```

For example:

```
# User mapping file
jmc:john collins
sec:sue collins
guest:default
```

The final entry gives a default user if a named user is not found in the file.

UNIX users not found on the host are silently ignored.

Chapter 6

Overview of Xi-Text architecture

This is a brief overview of **Xi-Text** for the assistance of administrators. Please see the System Reference Manual for a more detailed discussion.

6.1 Daemon processes

Xi-Text relies on one or three continuously-running "daemon processes" for its operation, plus one process for each running printer.

If the product is non-networked (no linked Unix host or hosts, no Windows interface or API), then there is an `spsched` process which allocates jobs to printers.

If the product is networked:

- A further `spsched` process monitors and processes network traffic between other Unix hosts and the Windows Interface.
- A process `xtnetserv` handles incoming jobs from the Windows Interface and manages API sessions. (It may "fork off" additional copies of itself for each API session).

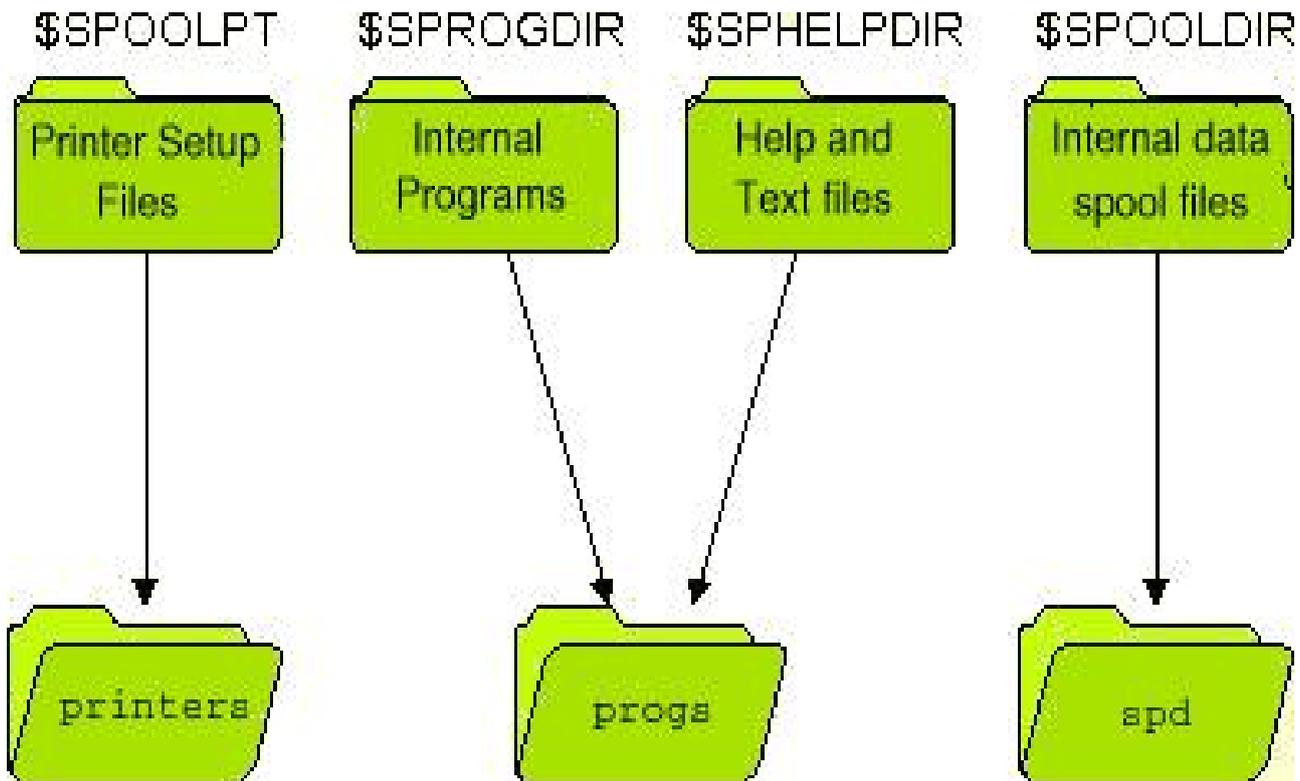
In addition there is an `spd` process for each running printer.

These processes are started by running `spstart` and terminated by using `sstop`. Please do not attempt to start **Xi-Text** in any other way.

Should some system crash ever require you to kill any of the `spsched`, `spd` or `xtnetserv` processes, please do not use "kill -9" initially, as this will force immediate termination. Instead use just "kill", which will give `spsched` a chance to release system resources first.

6.2 System directories

Xi-Text uses four logical directories to hold the internal programs and data. These are usually mapped onto three physical directories. A default installation would look like this:



These directories may be relocated by assignment to the four environment variables: `SPOOLPT`, `SPROGDIR`, `SPHELPDIR` and `SPOOLDIR`. These environment variable assignments may be placed in the master configuration file, `/etc/Xitext-config`, to ensure consistency. The default directories are as follows:

Default location	Environment variable	Function
<code>/usr/spool/spd</code>	<code>SPOOLDIR</code>	Jobs and other data.
<code>/usr/spool/printers</code>	<code>SPOOLPT</code>	Printer description files
<code>/usr/spool/progs</code>	<code>SPROGDIR</code>	Internal programs
<code>/usr/spool/progs</code>	<code>SPHELPDIR</code>	Help and message files

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

The system help files and the system programs have traditionally shared the same directory, by default `/usr/spool/progs`, mainly because there aren't very many files there. They don't have to, and you might want to have a separate file of customised help files to keep backed-up separately.

6.3 IPC Facilities

Xi-Text uses the "System V IPC facilities" to communicate with itself internally.

- One message queue is used to send requests for the main scheduler process, `spshed`.
- Two shared memory segments record jobs on the queue. These are periodically written out to the file `spshed_jfile` in the spool directory.

- One shared memory segment saves details of printers. This is likewise periodically written out to the file `spshed_pfile`.
- One shared memory segment is used to buffer pending requests as the size of messages which may be sent on message queues is limited on many systems.
- One set of semaphores controls access to the shared memory segments.

The IPC facilities can be displayed by running `ipcs`. The items in question are owned by `spooler` with a key of `0x58691xxx`.

The system utility `xt-ripc` provided in the installation directory may be used to inspect the IPC facilities especially after a crash and optionally to delete them.

The shared memory used for jobs and printers may have to be reallocated if the number of jobs or printers exceeds the initial amount allocated. On some systems this may be difficult if other software is in use which has exhausted the available shared memory space. If there are difficulties in this area, make sure to start **Xi-Text** with a reservation of the appropriate amount of space, as described under `spstart`.

On some releases of **Xi-Text** these shared memory segments may be replaced by memory-mapped files. On releases of **Xi-Text** subsequent to November 2004, file locks are usually used instead of semaphores.

6.4 Files used by Xi-Text

The following sections list the files used and created by **Xi-Text**.

6.4.1 Internal spool files

The following **Xi-Text** internal files are held in the spool directory, which by default is `/usr/spool/spd`:

File	Function
<code>spufile</code>	File of user permissions
<code>spcharges</code>	File of user charges
<code>sputmp</code>	Temporary file used when reformatting spufile
<code>spshed_reps</code>	Log file recording system start-up, shutdown and errors
<code>spshed_jfile</code>	File used for recording the job queue
<code>spshed_pfile</code>	File used for recording the printer list
<code>SPnnnnnnnn</code>	File holding the text of a spool job
<code>PFnnnnnnnn</code>	Holds page offsets where the page delimiter is not a form feed.
<code>ERnnnnnnnn</code>	Used to hold information to be mailed back to a user.
<code>sppmm_jobi</code>	<i>If using memory-mapped files</i> live job hash file.
<code>sppmm_jobd</code>	<i>If using memory-mapped files</i> live job data file.
<code>sppmm_ptr</code>	<i>If using memory-mapped files</i> live printers file.
<code>sppmm_xfer</code>	<i>If using memory-mapped files</i> live buffer file.

The above files are owned

by `spooler`. Unused copies of the last four kinds of files may safely be deleted. The `nnnnnnnn` component of the file name is derived from the print job number.

One of the main product options is to subdivide the spool directory into several smaller subdirectories, making the main directory more manageable.

Please note in particular the file `spshed_reps`. This is the system log file and processes such as `spshed`, `spd` and `xtnetserv` send messages to it in the event of error conditions arising. If you have any support questions, please look in this file.

6.4.2 Help and Message files

In common with other Xi Software products, **Xi-Text** reads all of its messages from a series of text files (Apart from the "help I cannot find the message file" messages). The user may adjust these to tailor the command interface, help and error messages to be suitable for the particular installation. These are *system-wide* message files. It is also possible to set up customised versions for individual users or applications.

The following files are, by default, owned by `spooler` and held in the directory `/usr/spool/progs`.

File	Function
<code>int-config</code>	Message file used by internal programs.
<code>rest.help</code>	Message file used by all other user programs.
<code>spq.help</code>	Message file used by <code>spq</code> .
<code>spuser.help</code>	Message file used by <code>spuser</code> .
<code>splpd.help</code>	Message file used by lpd interface.
<code>xmspq.help</code>	Message file used by <code>xspq</code> and <code>xmspq</code> .
<code>xmspuser.help</code>	message file used by <code>xspuser</code> and <code>xmspuser</code> .

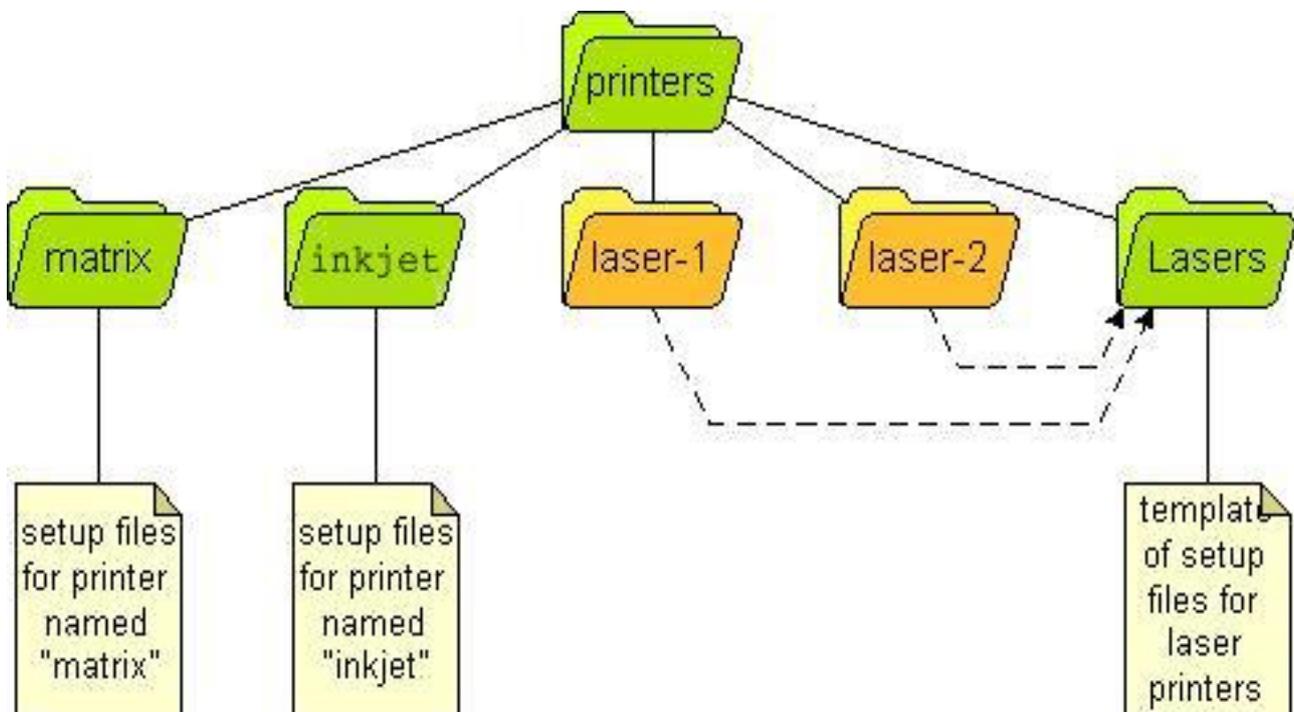
Refer to the chapters on Configuration and Extension tools of both the System Reference Manual and this Administrator Guide for details of how to modify these files as may be required.

Note that certain options in some of the programs `spq` and `spuser` may cause local copies of the files `spq.help` and `spuser.help` to be written out into user's areas, possibly under different names.

6.4.3 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 **Xi-Text** 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.



Symbolic links can also be used inside printer directories where just certain components are identical across two or more printers.

Refer to the chapter on Printers - Setting up and Administration for details of printer setup and their definitions.

6.4.4 Configuration files

Various programs allow sets of options to be written into local "configuration files" named `.xitext` so that further runs of those programs pick up a different selection of options from the defaults.

These are text files containing sets of options for various programs, which may be edited by the user, but are normally edited by the programs concerned.

6.4.5 Configuration files held in /etc

The following files are always held in the system directory `/etc`.

6.4.5.1 Xi-Text Hosts and Clients File

The file `/etc/Xitext-hosts` is used on networked installations of **Xi-Text** to denote details of the remote hosts and clients to which connection is to be made. This is fully described in the System Reference Manual.

6.4.5.2 Xi-Text 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 `/etc/Xitext-config`.

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. Any environment variables used by Xi-Text may be included in this file, not just those shown in the example.

For example:

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

Please note that the text to the right of the equal sign, `=`, is taken literally; there is no recursive expansion of `$name` constructs except for the message file names `SPQCONF`, `SPUSERCONF` and `SPRESTCONF` (for which up to 10 recursive expansions are applied).

6.5 Ports and Network Services

Xi-Text also uses 7 entries in the services file, `/etc/services`. When installed with the default values the entries will look like this:

```
xitext    2000/tcp # Connection port for Xi-Text
xitext    2000/udp # Probe port for Xi-Text
spq       2100/tcp # Feeder port for Xi-Text
xtnetsrv  2200/tcp # DOS client enqueue port for Xi-Text
xtnetsrv  2200/udp # DOS client enquiry port for Xi-Text
xtapi     2210/tcp # Xi-Text API library for Xi-Text
xtapi     2210/udp # Xi-Text API library for Xi-Text
```

The socket numbers can be changed, but all hosts must use the same numbers if **Xi-Text** connections are to be made. Please note the different spelling of the services entry, often limited to 8 characters, of `xtnetsrv`, the service, and the program `xtnetsrv`.

The installation script will set up entries for the **Xi-Text** TCP and UDP connections in the `/etc/services` file.

N.B. If you are using a name service like NIS instead of the services file, the entries must be copied to that service.

It is possible that the port/socket numbers used as standard by **Xi-Text** will conflict with another product. If this is the case they can be moved by editing the `/etc/services` file after installation. The entries are normally in the 2000 to 2300 range, if you move them it is simplest to add one or more thousand to each entry.

For example, if there are no other entries between 5000 and 6000 the standard could be changed to:

```
xitext    5000/tcp # Connection port for Xi-Text
xitext    5000/udp # Probe port for Xi-Text
spq       5100/tcp # Feeder port for Xi-Text
xtnetsrv  5200/tcp # DOS client enqueue port for Xi-Text
xtnetsrv  5200/udp # DOS client enquiry port for Xi-Text
xtapi     5210/tcp # Xi-Text API library for Xi-Text
xtapi     5210/udp # Xi-Text API library for Xi-Text
```

It is essential that the values used are the same on all **Xi-Text** hosts, Unix and PC, that are on the same network. Even if two hosts are never going to be connected, make the socket numbers the same.

Chapter 7

User administration

Xi-Text maintains a list of users which is generated from the password system (whether using the passwd file or NIS). Hence, each user must first have a Unix account in order to have a **Xi-Text** account.

There are 4 formerly 5) aspects to the **Xi-Text** user account:

Privileges Control access to usage and administration functions of the system. For example, the privilege to stop and start printers.

Charges (These are now deprecated).

Priorities When there is a back log of jobs to print the queue is sorted using a combination of priority and how long a job has waited.

Classcodes Group the users operationally. It is possible to put fire walls between different groups of users for security or convenience.

Restrictions To limit each user to a specific set of printers and/or formtypes.

The privileges, classcodes and printer restrictions 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. Printer restrictions can be used instead of or as well as the classcodes.

In addition to these features the user interface can be configured differently for different users and activities.

There are 6 programs for administering user accounts which are:

spcharge	(Deprecated)
spuchange	A command line utility for modifying user accounts
spulist	A command line utility for listing user accounts
spuser	An interactive tool for viewing and changing user accounts on character terminals.
xspuser	A GTK+ alternative to spuser .
xmspuser	A Motif GUI alternative to spuser .

The programs are described in detail in the System Reference Manual.

7.1 Adding New Users

No special action is required to add new users to **Xi-Text** unless the users have to have different permissions from the defaults.

7.2 Removing Users

No special action needs to be taken when users are deleted.

7.3 Setting Privileges

The available privileges are:

Privilege	Description
Edit admin file	May edit the administration file. This makes a user a full Xi-Text Administrator since the
Override class.	Allows user to override their own Classcode restrictions.
Stop spooler	Grants privilege to shutdown Xi-Text
Use other forms	Enables user to specify formtypes, other than their default.
Use other printers	Lets user override their default printer. Granted by default.
Change priority on Q	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 <code>spq</code>
Halt/restart printers	May start or stop printers
Add/delete printers	May add or delete printers on list
Set any priority on Q	May set any priority not just in range
Change own default prio/form	May run <code>spuser -c</code>
Unqueue jobs	May take jobs off the queue 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 queue options	May access the forms or screens behind the job list in <code>spq</code> and <code>xmspq</code> to modify param
Save new default options	May make and save changes to default parameters. Granted by default.

Unless otherwise stated in the table, when **Xi-Text** is installed these privileges are turned off. The currently specified default privileges are applied to all new users.

When **Xi-Text** is installed only the super-user, `root`, and user `spooler` are set up as system administrators. These two users are granted all of the privileges (and cannot lose them).

Changing users' privileges and the default settings can be performed with `spuser`, `spuchange`, `xspuser` or `xmspuser`.

When setting up **Xi-Text** on a machine one of the first Administration tasks should be deciding which extra Unix users to make administrators.

7.4 Setting Default & User Priorities

All print jobs have a priority in the range of 1 to 255. 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.

Each user has a default priority value which is given to jobs submitted by them, unless they specify otherwise. Users will usually be restricted to a smaller range between their individual minimum and maximum priorities. When it is installed **Xi-Text** sets the default values to be:

```
minimum 100
maximum 200
default  150
```

There is no particular reason for these values. They allow for non-standard users to set up jobs which have higher or lower priority than the normal user population.

When a new policy is decided the default values should be adjusted, then existing user accounts modified as required.

When a job is submitted, 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, for example the user may be prevented from submitting jobs at all by making the maximum priority less than the minimum.

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.

7.5 Setting Classcodes

Each user has a default set of classcodes. A user must be in at least one class in order to use **Xi-Text**. Only users with override class privilege can increase their set beyond their default set. All users can restrict their effective set to a sub-set of their default.

Changing users classcodes and the default settings can be performed with [spuser](#), [spuchange](#), [xspuser](#) or [xmspuser](#).

These are described in detail in the System Reference Manual.

Chapter 8

Other administration matters

8.1 Startup and shutdown of Xi-Text

The user-level programs `spstart` and `sstop` are provided for the startup and shutdown of **Xi-Text** and `spconn` and `spdisconn` are provided for attaching and detaching Unix hosts.

Two other administrative programs `ssuspend` and `srelease` are provided to temporarily suspend assignment of printers to jobs pending certain system-level activities.

Please try to use these utilities wherever possible rather than starting or halting internal processes yourself.

8.1.1 `spstart`

Xi-Text can be started by just running the program `spstart`. However there are options to `spstart` which should be worth noting:

If `spstart` is given two numeric options, then shared memory (or memory-mapped file) space will be reserved for the given number of jobs and printers respectively. This is often necessary on installations where it is not possible to reallocate shared memory after **Xi-Text** has been started, perhaps because other applications have taken all the available shared memory.

To start **Xi-Text** reserving space for 5,000 jobs and 200 printers, use the command:

```
spstart 5000 200
```

8.1.2 `sstop`

`sstop` should be used to halt **Xi-Text**. Any jobs running will be aborted. This may only be run by a suitably-privileged user.

If there are problems with it and some of the processes continue to run and you need to kill `spshd` and other processes, try first with just "kill" and not "kill -9" as the latter will not give a chance for the software to delete the IPC facilities.

8.2 spconn and spdisconn

`spconn` is used to connect to Unix hosts set up as for "manual connection" in the hosts file, or disconnected using `spdisconn`. `spdisconn` disconnects from other Unix hosts previously connected.

Either are invoked with the host name or alias required. They should return immediately, although the network startup or shutdown may take longer (or not at all if the other machine is unavailable).

8.3 ssuspend and srelease

Prior to connecting additional hosts, network traffic can be held off by invoking `ssuspend` to prevent printing being started (this generates a lot of network messages). `ssuspend` takes an argument in seconds for which it applies or it can be cancelled with `srelease`.

8.4 Startup on boot

The installation routine installs a startup routine, if one doesn't exist already. It contains an `spstart` command which includes values of the space for jobs and printers to allocate, as described on page 40). Please edit the values, especially that for job space, to avoid allocating too much shared memory on the one hand or too little on the other.

If the job or printers run out of shared memory space, additional space is allocated, but this is not always possible later. Hence it is advisable to allocate space initially for the maximum number of jobs which are likely to be held in the queue.

8.4.1 Starting all printers on boot

We don't recommend that this be done as a general rule, but many of our users like all the printers (local to that machine) to be restarted on boot. If you want to do this, uncomment the lines given in the startup file.

8.4.2 Cleaning and subdividing the spool directory

Over time various "dead wood" files may be left in the spool directory. There may be jobs listed in the job list file (`spshed_jlist`) which don't have any counterpart data files (`Spnnnnnnnn`) or vice versa.

There is a utility program `setspdir` which may be used to tidy up the spool directory and either requeue orphaned job files or to delete them.

As a further option, where there are a large number of job files, it is often helpful to share them out between subdirectories of the main spool directory, so no directory has a huge number of files in it. `setspdir` lets you set or change that.

Chapter 9

Backup of Xi-Text System

Certain utilities are provided to back up the **Xi-Text** system, by saving the current jobs, printers and user accounts. You may want to do this when you have set up a complicated list of printers. Alternatively you may want to do this if you are upgrading to a new major release of **Xi-Text**. We make efforts to keep a common format between minor releases, so it should not be necessary to do this.

All the backup options create a single shell script which, if run, would recreate the relevant jobs, printers or user list. This may be edited if required.

Jobs, printers and user profiles should be saved in any order, but it is probably best to restore jobs last as they may depend on some of the other items.

9.1 Jobs

To convert the job list to a shell script, the utility `xt-cjlist` is used. This may be run at any time, even when **Xi-Text** is running, but we would suggest that not too much activity is in progress at the time.

`xt-cjlist` may be found in the user-level programs directory (we used to keep it in the distribution directory but too many people wanted it quickly found).

You will need to create a directory into which the job data files are saved.

Here is a suggested routine for saving the jobs:

1. Create a backup directory, for example `/usr/textsave/May05`.
2. Create a subdirectory within that directory for the scripts, for example you might choose `/usr/textsave/May05/Scripts`.
3. Run `xt-cjlist` on the existing job files.

Thus:

```
mkdir -p /usr/textsave/May05/Scripts
cd /usr/textsave/May05
xt-cjlist() -D /usr/spool/spd spshed_jfile Jcmd Scripts
```

The shell command to recreate the jobs will be put in `Jcmd`, and the job script files in `Scripts`.

For more details, and especially the handling of errors, please see the documentation of [xt-cjlist](#).

[xt-cjlist](#) only considers and saves jobs on the machine on which it is run.

To restore the jobs, just run the shell script `Jcmd`.

9.2 Printers

Printers are saved in a similar manner to jobs, except no additional directory is required, using the program [xt-cplist](#). Only one file is generated, a single shell script of `spadd` commands to recreate the printers.

Assuming that the `/usr/textsave/May05` directory described above has already been created and selected to save jobs, the following command will save the printers:

```
xt-cplist() -D /usr/spool/spd spshed_pfile Pcmd
/usr/spool/spd
```

When restoring jobs and printers, you will almost certainly need to restore printers first.

Please see the documentation of [xt-cplist](#) for more information, particularly regarding error handling.

9.3 User permissions

User permissions may be saved in a similar manner to jobs and printers, using the command [xt-spuconv](#).

As with [xt-cplist](#) a shell command is saved which when run will reset the user permissions and defaults for all users.

To continue the examples above for jobs and command interpreters, the following will create a command file capable of recreating the user permissions.

```
xt-spuconv -D /usr/spool/spd btufile6 Ucmd
```

When restoring the whole system, it is not necessary, but probably desirable, to restore the user permissions prior to restoring jobs and printers.

Please see the documentation of [xt-spuconv](#) for more information, particularly regarding error handling.

9.4 Upgrading from Earlier Releases of Xi-Text

This section describes the preparations to be made before and actions to take when using the installation kit to upgrade from an earlier version of **Xi-Text**. The main topic is include how to bring forward the relevant information.

Release or version numbers consist of two parts:

1. The *major release Number*, which precedes the decimal point.
2. The *minor version Number*, which follows the decimal point.

Thus if the full release number is 23.170, then it has a major release number of 23 and a version number of 170.

The saved file structures only change between Major Release Numbers, so if you are, for example, upgrading from version 23.43 to 23.361, you should not have to save and restore any data files. You should be able to stop the old version, install the new binaries, and restart the new version.

On the other hand, internal interfaces may change slightly, so you should install all the binaries of the new release.

It is only when you move between major releases, such as from Release 22.x to Release 23.y, that you need to take additional action to preserve the status of jobs, printers etc.

9.4.1 Help and resource files

Help files in the help file directory (`/usr/spool/progs` by default) will be replaced when the software is upgraded. We suggest that you make a copy of the existing versions before you begin and if necessary consult us regarding integrating your customisations into the new release. In most cases the changes will be correction of spelling mistakes and clarifications of existing error messages.

Please note also that in some cases, such as when users save a customised set of screen formats for `spq` or `spuser`, a local copy of the files `spq.help` etc are saved in the user's home directory. These should probably be deleted and recreated as necessary.

The X programs `xmspq` and `xmspuser` have information kept in a file called `XI` in an `"app-defaults"` directory. When a user tailors their user interface, a local copy of the `XI` file is created in their home directory, which may not reflect all the changes in the new versions of the software.

9.4.2 Carrying Information to the New Installations

If you are upgrading a major release, e.g. from Release 22.x to Release 23.y, you may want to carry information such as jobs, printers and user permissions forward into the upgraded version. It should not be necessary at all if you are upgrading from the same major release, i.e. from 23.aaa to 23.bbb.

You may however want to use these procedures to take a "snapshot" of a configuration of jobs and printers, perhaps for backup purposes.

The installation scripts provide menu options to invoke some or all of these options, alternatively you may prefer to use some or all of the following procedures to fit in with your individual requirements.

Strenuous efforts are made to cater for data files from historical versions of **Xi-Text** and to appropriately convert them. To a limited degree, errors in data files are also catered for.

Chapter 10

Print jobs and form types

This is a summary of some of the major features of print jobs and form types appropriate for this administration manual. For full information, please see the System Reference Manual.

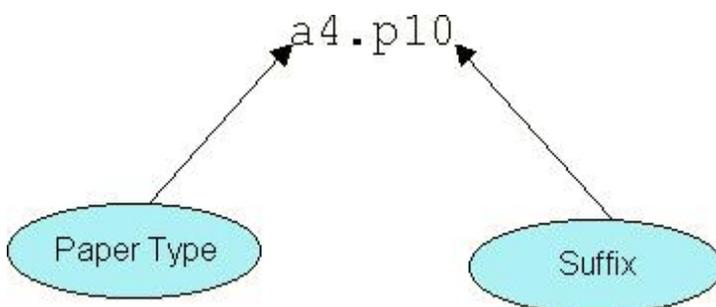
10.1 Print Jobs

A print job consists of a set of data which is to be output and the specification of how to output it. The data can be ASCII text, Postscript, Graphics or anything else that is understood by the output device. The specification controls things like how many copies to produce, what formatting to apply, which output devices to use and so on.

10.2 Formtypes (Paper Types & Suffices)

There are many different types of media that can be used in output devices like printers and plotters. There are also many different ways of formatting the data that is output on each media type. To support this idea **Xi-Text** has a concept of a formtype.

A form type consists of two parts, a "paper type" corresponding to the actual paper loaded and an optional "suffix" denoting variations of how to print on it, for example:



In order to change the paper type, the printer has to be stopped and started on the spooler. However the suffix can change automatically as different jobs are printed with different suffixes.

In many cases, there is no need for special treatment and no suffix is required at all and the form type can be something like labels or cheques.

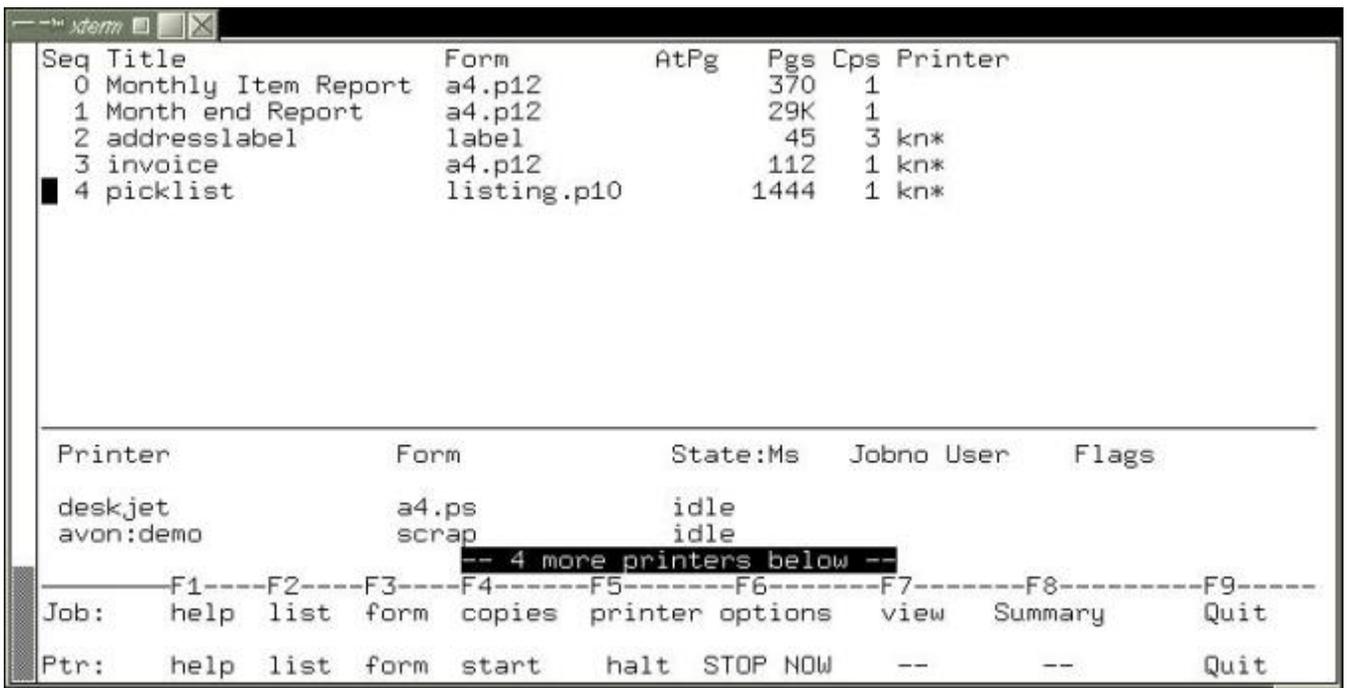
Various options in the printer setup file control actions to be taken at various stages of the print cycle. Generally speaking, there is a setup file for each paper type and for each printer. Suffix handling is done within the setup file.

Chapter 11

Configuration options

Xi-Text can be highly configured to restrict or enhance the scope and function of user activities. The System Reference Manual gives the basic information for configuring the user interfaces. This chapter describes some of the applications of this configurability and their implementation.

Here is an example that shows how program `spq` might look when configured to take advantage of function keys and show a different set of information:



```
Seq Title          Form          AtPg   Pgs Cps Printer
0 Monthly Item Report a4.p12         370   1
1 Month end Report  a4.p12        29K   1
2 addresslabel     label         45    3 kn*
3 invoice          a4.p12        112   1 kn*
4 picklist         listing.p10   1444  1 kn*
```

```
Printer          Form          State:Ms  Jobno User  Flags
deskjet          a4.ps         idle
avon:demo        scrap         idle
-- 4 more printers below --
```

```
F1----F2----F3----F4----F5----F6-----F7-----F8-----F9-----
Job:  help list form copies printer options view Summary Quit
Ptr:  help list form start halt STOP NOW -- -- Quit
```

This configuration could be specific to a particular user or activity. In this case it is taken from a real configuration belonging to a particular user when sending jobs to printers to a particular office. The screen display has been changed as follows:

- The `Jobno`, `User` and `Priority` fields in the job list have been removed.
- The `device` and `Lim` fields in the printer list have also been removed.

- The `Title` field is widened to display more text.
- The footer is expanded to include function key reminders.
- The blank line under the Printer field column headings has been removed to allow more printers to be seen.

This is what the standard configuration of `spq` looked like when invoked on another terminal, at the same time:

```

xi-term
Seq Jobno User Title Form AtPg Pgs Cps Pri Printer
torres:243 jmc Monthly Item R a4.p12 370 1 164
torres:243 jmc Month end Repo a4.p12 29K 1 153
2 3476 jmc addresslabel label 45 3 155 kn*
torres:243 jmc invoice a4.p12 112 1 150 kn*
torres:243 jmc picklist listing.p10 1444 1 100 kn*

Printer Device Form State:Ms Lim Jobno User Flags
deskjet lp0 a4.ps idle
avon:demo <dh> scrap idle
-- 4 more printers below --

Xi-Text spq (c) Xi Software Ltd 2000 (? for help)

```

As well as changing the format, the view may be restricted so only particular printers and jobs for them are displayed, or possibly only jobs submitted by particular users.

The following sub-sections describe various ways in which the interface can be configured.

11.1 Default Options

All of the programs in the Base Product can be given or require options on the command line. Default options and values can be specified to save typing or to enforce their use. The defaults can be set up globally, per user, by current working directory or by activity.

11.1.1 Setting Up Defaults

Default options for each command line utility, like `spr` and `sqlist`, is specified using an environment variable. The environment variable for `spr` is called `SPOOL_OPTS`. All of the other programs use a variable of the same name as the program, in upper case. This holds a string of one or more options and any associated arguments.

Each program additionally uses a further environment variable to which may select an alternative message file. There is a rather more restricted set as programs, particularly non-interactive ones, share message files.

The program descriptions in Chapter 6 of the System Reference Manual list the keywords at the start of each entry. For example for `spq`, the description shows a keyword of `SPQ` for options and `SPQCONF` for an alternative help file.

The chapter on Configurability in the System Reference Manual explains the theory. Here are some examples of how defaults can be used for `spq` :

```
SPQCONF=/usr/spool/progs/spq/whouse.help
```

This tells `spq` to load the file `whouse.help` in the specified directory instead of `spq.help` from the `progs` directory. It is a good idea to use meaningful file names, in this case `whouse.help` could be the configuration for "Warehouse Jobs".

If all of these "Warehouse jobs" are going to specific printers in the warehouses the `-q` option can be used to restrict the view accordingly. For example:

```
SPQ=-Z -q whlist*,whlabel*
```

This just selects jobs whose printer name starts with `whlist` or `whlabel`. The `-Z` option excludes jobs which have no printer specified.

11.2 Setting views of the Job and Printer lists

There are facilities for selecting which jobs and printers to display. Whilst these can be used for security purposes they are also just as useful for selecting logical views of the print spooling system.

Users can have logical views imposed upon them or be allowed to select their own. A user can only affect a job or printer if they can see it.

11.2.1 Selecting Jobs and Printers by Printer Name

Programs `spq`, `sqlist`, `xspq` and `xmspq` have options for printer name selection. The first two use Keywords for setting default values and the Motif program

`PrXmspq` uses two resources in the `XI` file. For example to select only jobs for printer `Ljet`:

Program Keyword / Resources

<code>spq</code>	<code>SPQ=-Z -q Ljet</code>
<code>sqlist</code>	<code>SQLIST=-Z -q Ljet</code>
<code>xmspq</code>	<code>xmspq.onlyPrinter: Ljet</code>

11.2.2 Selecting Jobs by Owner

Programs `spq`, `sqlist`, `xspq` and `xmispq` have options for selection by user name. The first two use Keywords for setting default values and the Motif program `xmispq` uses resources in the `XI` file. For example to select only jobs owned by user `fred`:

Program	Keyword / Resources
<code>spq</code>	<code>SPQ=-ufred</code>
<code>sqlist</code>	<code>SQLIST=-u fred</code>
<code>xmispq</code>	<code>xmispq.onlyUser: fred</code>

11.2.3 Selecting Jobs by Title

The output from `sqlist` and the job lists in `spq` and `xmispq` can be reduced to show only jobs whose Title field match a pattern. The pattern is specified in the same way as it is for printer names and users. From the command line the Title may be specified by the options:

```
sqlist -t selection
spq -t selection
xmispq -xrm "xmispq.onlyTitle: selection"
```

An alternative to the `-t` option is the `+just-title` keyword. The master `XI` file and user's copies can be edited manually to include an `xmispq.onlyTitle` line.

A new selection can be specified from within `spq` by pressing the `=` key from either the printer or job list on the main screen. This brings up the Program options screen, which has the title parameters on the third line down. In `xmispq` the equivalent window is opened using `View options` from the `Options` menu.

The jobs may be selected by a simple string, a pattern or a list of strings and patterns for text in the Title. Patterns and strings in a list are separated by commas. It may be necessary to put quotation marks around the Title specification when invoked from a shell command.

The wild-card options for pattern matching are:

<code>*</code>	Matches anything
<code>?</code>	Matches one character
<code>[a-m]</code>	Matches one character in list or range
<code>[!n-z]</code>	Matches one character not in list or range

They may be given as a comma-separated list of alternatives, including the use of shell-style wild-cards. For example

<code>Sales*</code>	The string <code>Sales</code> selects all jobs whose Title starts with the word <code>Sales</code> . It does not select any jobs which have one or more characters before the word <code>Sales</code> , or which are spelt with a lower case <code>s</code> at the beginning.
<code>Sales*, *Rental</code>	Selects jobs starting with the string <code>Sales</code> or ending in the string <code>Rental</code> .
<code>*Sale*</code>	More than one asterisk can be used. This example selects any Title containing the string <code>Sale</code> . It does not have to be a separate word, i.e. the words <code>Sales</code> and <code>Salesman</code> will also match.
<code>[S,s]ales</code>	Use the list or range wild card to cope with similar patterns like words with lower case and capital letters.

The wild cards can be used in combinations like this:

`*[R,r]eport*` will select all the jobs with the word `Report` or `report` in their title.

To invoke `spq` with the job list restricted using the `-t` option from the command line could look like this:

```
spq -t "[M,m]onth*[R,r]ep"
```

This invokes `spq` with the job list restricted to "`Monthly Reports`" whose titles match the pattern. These will include titles like:

```
Monthly Sales Report
End of month defect reps
```

It does not match titles where the pattern `rep` or `Rep` comes before the pattern `Month` or `month`, such as

```
Sales report at month end
```

To cope with this extra requirement the selection would have to be a list of two patterns looking something like

```
spq -t "[M,m]onth*[R,r]ep*, *[R,r]ep*[M,m]onth"
```

11.2.4 A real example

The Jobs in the example screen for `spq` at the beginning of this Chapter were selected by invoking `spq` with the command:

```
spq -Z -q kn* -u wally
```

This could have been set up as the default in a `.xtext` file using a line like this:

```
SPQ=-Z -q kn* -u wally
```

Alternatively it could have been set up in an environment variable using the commands (assuming the Bourne or Korn shell are in use):

```
SPQ="-Z -q par* -u wally"
export SPQ
```

If these commands are being executed in a general shell script, the current user could be specified by replacing `wally` with `$LOGNAME`. For example:

```
SPQ="-Z -q par* -u $LOGNAME"
export SPQ
```

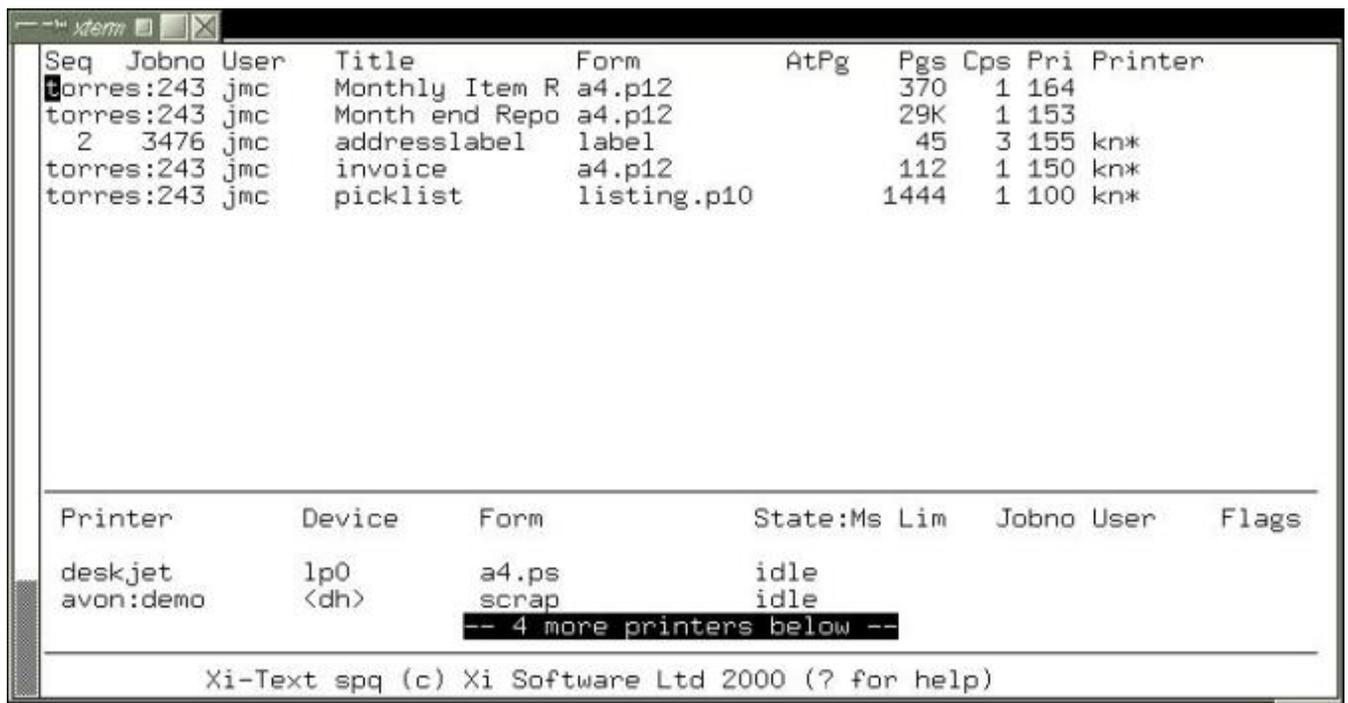
The `$LOGNAME` is evaluated by the shell not the **Xi-Text** programs.

11.3 Job and Printer list Formats

The job and printer list formats are described in the System Reference Manual. To quickly create a new layout the format can be re-specified and tested from within `spq` or `xmspq`. New formats can be saved at any time, for use or further development later.

To illustrate how to achieve the format given at the start of this chapter, let us show the steps.

Firstly create a working directory and change to it. Then start `spq` to get the standard screen:



```

xi-term
Seq Jobno User Title Form AtPg Pgs Cps Pri Printer
torres:243 jmc Monthly Item R a4.p12 370 1 164
torres:243 jmc Month end Repo a4.p12 29K 1 153
2 3476 jmc addresslabel label 45 3 155 kn*
torres:243 jmc invoice a4.p12 112 1 150 kn*
torres:243 jmc picklist listing.p10 1444 1 100 kn*

Printer Device Form State:Ms Lim Jobno User Flags
deskjet lp0 a4.ps idle
avon:demo <dh> scrap idle
-- 4 more printers below --

Xi-Text spq (c) Xi Software Ltd 2000 (? for help)

```

From the job list enter the `,` command. This opens the Job list formats screen, showing the current specification:

```

Job list formats
  Width Code
█      3 n Number in queue
"      6 N Job number
"      6 u User submitting job
"      14 h Job title
"      13 f Form type
"      5 Q Page reached
"      5 S Size in Pages
"      3 c Copies
"      3 p Priority
"      14 P Printer

```

Each row in the display corresponds to a column in the job list, showing the width and the code letter which selects the job attribute to be displayed. Many of the fields are separated by spaces.

First let us delete the job number. To do this, move the cursor to the job number line, and press *D*. The display will look like the following:

```

Job list formats
  Width Code
"      3 n Number in queue
█      "
"      6 u User submitting job
"      14 h Job title
"      13 f Form type
"      5 Q Page reached
"      5 S Size in Pages
"      3 c Copies
"      3 p Priority
"      14 P Printer

```

We don't want two separators, so press *D* again to delete the second separator, *D* again to delete the user name and *D* again to delete the third separator. The screen should now look like this:

```

xterm
Job list formats
      Width  Code
      3  n  Number in queue
" "
█ "      14  h  Job title
" "
      13  f  Form type
      5  Q  Page reached
      5  S  Size in Pages
" "
      3  c  Copies
" "
      3  p  Priority
" "
      14  P  Printer

```

Next we can increase the width of the job title field. To do this type `w` and key in a new value, say `24`.

Finally, delete the priority and following space by moving to the field and typing `D` twice. The display should now show:

```

xterm
Job list formats
      Width  Code
      3  n  Number in queue
" "
      24  h  Job title
" "
      13  f  Form type
      5  Q  Page reached
      5  S  Size in Pages
" "
      3  c  Copies
█ "
      14  P  Printer

```

Type `q` to quit the screen. The following prompt will appear

```
Save format codes?
```

Type *y* to get the prompt

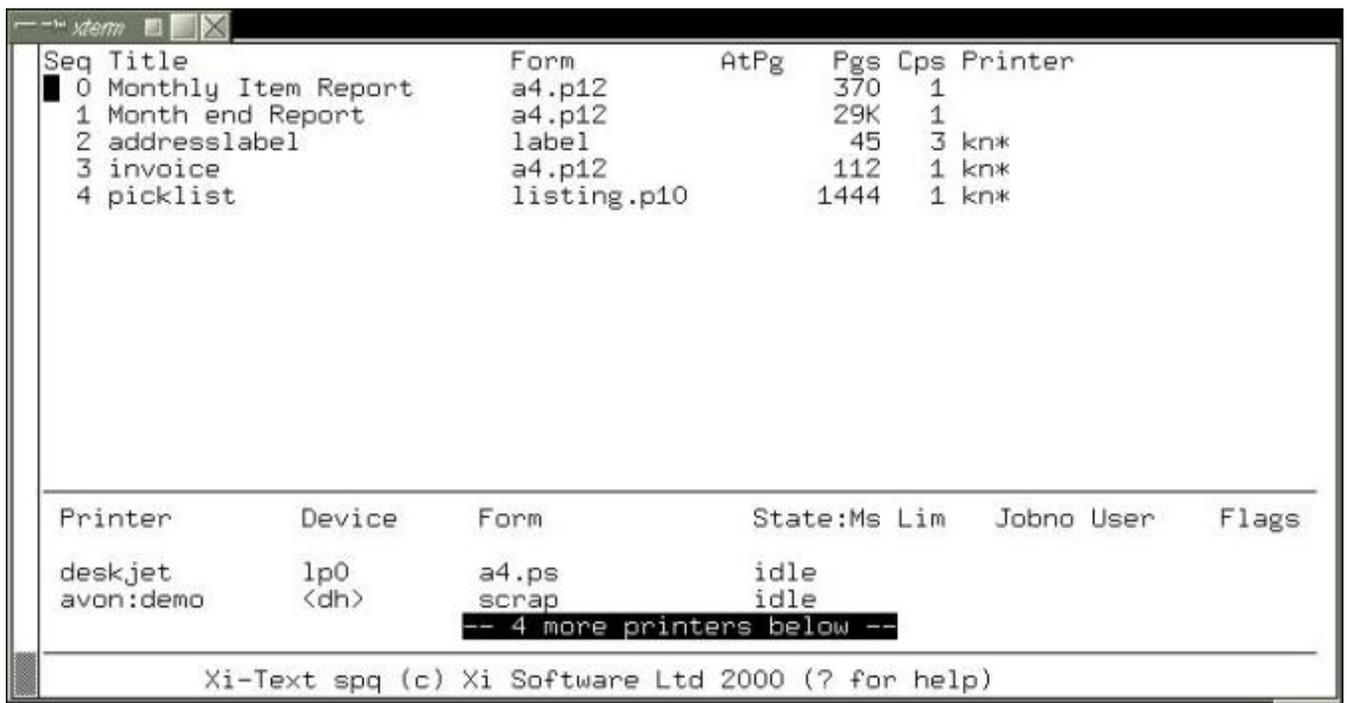
```
Save in current directory?
```

Type *y* to get the prompt

```
What file:
```

Type in a file name, e.g. *myhelp*, press RETURN and the formats are saved for next time you enter *spq*.

The display will now look like this:



```

Seq Title                               Form      AtPg  Pgs  Cps  Printer
0 Monthly Item Report                   a4.p12    370   1
1 Month end Report                       a4.p12   29K   1
2 addresslabel                           label     45    3 kn*
3 invoice                                 a4.p12   112   1 kn*
4 picklist                                listing.p10 1444  1 kn*

Printer      Device      Form      State:Ms Lim  Jobno  User  Flags
deskjet      lp0         a4.ps     idle
avon:demo    <dh>       scrap     idle
-- 4 more printers below --

Xi-Text spq (c) Xi Software Ltd 2000 (? for help)

```

This has now completed the first task, that of setting the job list display.

To reset the printer list format, move to it, and go through the same steps, starting with the *,* command, as for jobs. However you will have to specify a different name for the message file, as the current file is not overwritten.

The configuration was saved in a new help file, which was then edited to give the finished result.

The lines in the middle and at the bottom of the screen were also edited using a text editor. By default the specification for the lines in the middle of the screen is this:

```

D1:=====
D2:p
D3:

```

and the specification for the footer lines at the bottom is

```

T1:=====
T2:                Xi-Text %P (c) Xi Software Ltd 2000 (? for help)

```

To lighten the dividing line between the jobs and printers the = signs were edited to be ordinary dashes, -. The blank line between the column headings and the printers was removed by deleting the D3: line. The result is:

```
D1:-----
D2:p
```

To put the Function key reminders at the bottom of the screen replace the two original lines with four that look like this.

```
T1:-----F1----F2----F3-----F4-----F5-----F6-----F7-----F8-----F9
T2:  job:  help  list  form  copies printer options  View  Summary Quit
T4:  ptr:  help  list  form  start  halt  STOP NOW  --  Quit
```

Setting up the function keys to actually do something is a separate exercise described later in this chapter.

11.3.1 Adding New fields

New fields, to display other job or printer attributes, and extra separators may be added to the display, in a similar manner to the change and delete options.

The relevant commands are:

<i>i</i>	Insert new field before current row
<i>a</i>	Insert new field after current row
'	Insert new separator before current row
"	Insert new separator after current row
<i>w</i>	Change field width
<i>c</i>	Change code (job or printer attribute to be selected)
<	Toggle left overflow flag (if set, over-long fields may overflow into previous field)
>	Toggle right overflow flag (if set, over-long fields may overflow into the next field).

Asking for help when setting the code will list possible fields. The width is initialised to a default setting for whichever field is selected.

11.4 Help & Error Messages

Almost all of the help and error messages output by **Xi-Text** programs are held in the help files. These messages can be edited with any ordinary text editor.

Each message can be made more informative or shorter. The terminology can be changed to reflect local standards or the whole file can be re-written in a different language. For example, in the `spq` job list, pressing a key with no associated command produces this error message:

```
Unknown command - expecting job queue control
```

The message is defined in the help file, in an error line which looks like this:

```
E3000:Unknown command - expecting job queue control
```

When one of the programs needs a message it looks down the help file for all lines beginning with a particular code. In this case `E3000` is the relevant error message.

To enhance the message it could be edited and or have additional lines appended. This version offers some constructive advice:

```
E3000:Unknown Key - Expecting a job related command
E3000:You probably pressed the wrong key by mistake
E3000:Enter a ? or press F1 to list commands & keys.
```

11.5 Keys & Commands

The keys and commands of the interactive tools `spq` and `spuser` can be re-configured. One or more keys is mapped to each command by an entry in the relevant help file which looks like this:

```
K400:?
K401:^L
K402:\e
K403:\s,\t
K404:\r
K405:Q,q
K406:k,\kUP
K407:j,\kDOWN
```

or this:

```
1K500:A
1K501:O
1K502:I
1K503:U
1K504:C
1K505:o
1K510:c
```

The entries which start with a letter `K` are global and those that start with a number apply to a specific sub-screen or option. The component before the `:` specifies the command. The component after the `:` is a comma separated list of key definitions.

11.5.1 Specifying Different Keys

An entry for a command may have one or more keys defined. The default key for requesting help is a question mark. In the `spq` help file it is specified like this.

```
K400:?
```

If the *terminfo* file defines the sequences for the various function keys correctly (not always the case!) the F1 key could be set up for getting help instead of the question mark. Change the entry to look like this:

```
K400:\kF1
```

It is possible that not all of your terminals will support function keys properly. In this case both keys can be specified, like this:

```
K400:?,\kF1
```

11.5.2 Disabling Commands

If there is no key defined for a command that command cannot be invoked. Deleting, or preferably converting to comment, the entry in a help file for a key effectively disables that command.

For example the delete job command for `spq` is specified by the help file entry:

```
1K500:A
```

To disable this command comment the entry out by prefixing it with a `#` like this:

```
# 1K500:A
```

Having disabled a command it is best to edit the help messages to reflect the change. The help associated with delete command for the job list will be adjacent to the key definitions in the help file.

In the case of the delete command the only line that needs changing is from top level help that displays the available commands. This is the line

```
H1:A Abort job
```

If this line contained information about other commands it would need editing. Since it only relates to the delete command it can be commented out:

```
# H1:A Abort job
```

11.5.3 Customising Commands

The function of a particular command can be changed by substituting a macro. Macros are described in the chapter on Extensions. First the existing command must be disabled as described above. Then a macro is set up which is invoked by the same key or keys as the original command.

For example the Delete command could be replaced by a macro which silently unqueues the job to an archive directory. This could be useful for accident prone users.

Chapter 12

Extensions to the toolkit

There are various mechanisms for enhancing or extending the functionality of **Xi-Text**, which go beyond customisation of the user interface. Some of these mechanisms are separate products with their own documentation. These are the 'C' programmer's API for Unix and the API for Windows PCs.

Chapter 8 of the System Reference Manual describes the facilities which are built into the basic product and the Motif GUI option as standard. This chapter gives some example applications of these facilities and how to set them up.

12.1 Printer Error Handling

The spooler can call a custom built script or a program whenever a locally defined printer goes into error. To define an error handler a one line entry must be added to the `/etc/Xitext-config` file. If the program is called `ptr_alert` and is held in the `/usr/spool/progs` directory the line would look like:

```
SPPTRMSGS=/usr/spool/progs/ptr_alert
```

The full path for the program must always be specified. This is because it is invoked by the spooler daemon, `spsched`, which does not usually have an appropriate `PATH` environment variable.

Any time that a local printer goes offline or into error the spooler invokes a copy of the program with the following 3 arguments:

1. Name of the printer.
2. Device name.
3. Description or comment.

The program could be an interface to an Alert Management Package or paging system. An example we have on a Sun machine running Solaris 2.5 uses the machine's sound card to give an audible warning. The warnings are spoken messages held in sound files.

The script looks something like this:

```
#
```

```

# Set up the path because there is none supplied
# by the spshed daemon.
#
PATH=/usr/sbin:/usr/bin
export PATH
#
# Put a message on the console.
#
echo "\nPrinter $1 needs attention!" > /dev/console
echo "===== \n" > /dev/console
#
# Play audio files.
#
audioplay -v 85 -p speaker /usr/spool/progs/Alert_begin
audioplay -v 85 -p speaker /usr/spool/progs/Alert/$1
audioplay -v 85 -p speaker /usr/spool/progs/Alert_end

```

The name for the middle file is derived from the printer name. There is a separate file for the name of each printer. If printer Ljet went into error the message might be:

```
<sound effect> Printer Laserjet needs attention
```

This script could be expanded to loop, repeating the message every so often, until the error has been cleared. It would check the printer status using splist or spstat.

12.2 Scripts

The command line programs for **Xi-Text** enable all job, printer and user information to be queried and / or modified. These can be built into new commands using shell scripts or used within user applications.

Here are two examples which use the command line utilities to view and manage the printer audit trail files.

12.2.1 Printer Log Files

Printer Definitions include an option to keep an audit trail of all jobs printed. This is specified in the setup file by the keyword logfile. A different audit trail can be kept for each papertype. The logfile specification looks like this:

```
logfile=.log
```

In this case the audit trail will be written to a file called .log in the Printer Definition directory. An absolute pathname can be specified for a different directory.

12.2.1.1 Viewing a Log File

Here is a shell script ,which given the printer's name, will open the currently active log file for viewing. It uses the splist command to get the formtype and from there find the active log file.

This example assumes that all printers have unique names. If they do not the script must be enhanced to take device names as well.

```
#
# Change to the Printer Definition directory
# (the path may be different on your machine)
#
cd /usr/spool/printers/$1
#
# Get the formtype for the printer and extract the
# papertype - which is the same as the setup filename.
#
PAPER=`splist -F "%f" $1 | cut -d "." -f1`
# Check to see if there is a formtype specific setup file
# and if not use the file named "default" instead. If neither
# file exist then give an error message and quit.
#
if [ ! -a $PAPER ] then if [ -a default ]
then PAPER=default
else
    echo Bad Printer Definition for $1
    echo -----
    echo Missing Setup Files for formtypes: $PAPER and default
    echo
    echo press return to continue
    read reply
    exit 0
fi
fi
# Extract the name of the logfile from the setup file
# and open it with a suitable editor or listing program.
#
LOGFILE=`grep "logfile=" $PAPER | cut -d "=" -f2`
view $LOGFILE
exit 0
```

Only users with sufficient permissions will be able to access the printer setup and log files. The script is suitable for use as a macro from within `spq`, `xspq` and `xmspq`.

12.2.1.2 Archiving / Purging Log Files

To prevent log files getting too big you may want to archive and/or purge them regularly. Before doing this the printer should be halted. This will stop any data being written to the file during the purge/archive process. If the original file is renamed, it also ensures that the printer daemon picks up the correct file when re-started.

A script could be produced to halt the printer if running, archive/purge the file then re-start the printer if it had been running. The script should allow the current print job to finish before manipulating the log file(s). The procedure might be something like this:

```
#!/bin/ksh -vx
# use splist to get the printer state and test for it being
```

```

# halted, in error or offline.
if [ `splist -F "%t" $1 | egrep '(halted|offline|error)'` ]
then
    RESTART=NO
else
    RESTART=YES
# Use sphalt to halt printer at end of job and use splist
# to wait for it to finish.
    sphalt $1
    while [ `splist-F"%t"$1|egrep'(halted|offline|error)'` ]
    do sleep 5
    done
fi
#
# **** Do log file operations here. ****
#
# Use spstart to re-start the printer if required.
if [ $RESTART = "YES" ]
then
    spstart $1
    echo starting $1 $RESTART
fi

```

This script could be run automatically at regular intervals or set up as a Macro in [spq](#), [xspq](#) or [xmspq](#) for System Administrators.

12.3 Macros

Amongst the different organisations which use **Xi-Text** there are a wide variety of functional requirements. To cater for some of the more individual needs a mechanism has been provided for programming new functionality and incorporating it into the standard queue management tools. These "programs" are called Macros.

Any program which takes a job id number, printer name or set of user names as an argument can be installed as a macro. They can be any executable program, whether a shell script or binary.

12.3.1 Publishing Documents for Printing on Demand

Xi-Text has a built in document viewer for plain text files. It can also invoke third party viewers for files using printer languages like postscript. This enables users to view a file before printing to see if they really want a hardcopy.

It is possible to submit a job to **Xi-Text** with a specification that prevents it from being printed. Users can view the job, then route it to a printer near them if they want a copy. This could be used as a mechanism for distributing documents.

The following script will take a copy of the original job and submit it for printing at the requesters printer. By taking a copy, conflicts between two or more users asking for a print at the same time are avoided. An environment variable called MYPRINTER must be set up in each users environment to specify their printer.

```
#!/bin/ksh
#
# Setup unique file name prefixes
CID=$$C
JID=$$J
# Get the title and formtype of the job
SPOOLOPTS=`sqliist -F '-h "%h" -f %f' $1`
export SPOOLOPTS
# Unque a copy of the job without deleting the
# original. Ignore the command file and use an
# spr command to submit a new copy.
sqdel -k -u -J $JID -S $CID $1
spr -P $MYPRINTER ${JID}*$1
# Tidy up all files used.
rm ${CID}*$1 ${JID}*$1
```

The environment variables `CID` and `JID` are used to provide unique filenames for the job and command files. `sqdel` produces files whose names begin with a specified prefix and end with the job number. The prefixes are specified using the `-J` and `-S` options for Job and Command files respectively. The number may have several leading zeros.

Using the process id number in the prefix avoids clashes between two users taking a copy of the job at the same time. Using an `*` pattern between the prefix and significant digits of the job number avoids having to work out how many leading zeros there are.

`SPOOLOPTS` is the environment variable for default options to `spr`. A single `sqliist` command is used to set up the formtype and title in `SPOOLOPTS`.

12.3.1.1 Enhancements

This script handles the basic mechanics of copying the job to a pre-defined printer. There are various ways in which it could be enhanced. For example:

1. The user could be invited to select a printer for output. The script should offer a choice of the most suitable printers and check the reply. The formtype and printer restrictions are important parameters for identifying suitable printers.
2. Logging copy requests. Each request for a copy could be logged in a database. Alternatively the originator could be sent an e-mail to notify them of each copy requested.
3. Page selection can be incorporated.
4. A special printer could be set up to work with the original document. This printer could be a program not a hardware device. When a job is submitted to the printer, instead of printing the job it would read the distribution list and e-mail each recipient with details.

5. The printer setup file should contain the retain keyword. This is better than having to remember to submit jobs with retention specified.

Chapter 13

Advice & Trouble Shooting

When encountering a problem the following steps should be taken:

1. Save what ever information is possible without endangering the system. Having the exact text of any error messages can be very helpful. Please let us have the last few entries in the system log file, which by default is to be found in `/usr/spool/spd/spshed_reps`.
2. Make the system safe.
3. Collect as much information as possible. This includes identifying everything involved, (software, hardware and human) what they were doing, when and where.
4. If **Xi-Text** was working and now it is not, look for what has changed. There is always something, it just might not seem relevant.
5. When altering the system to see what happens only change one thing at a time, document it and save the results.

How much information to collect is always a compromise between how much time is available and how severe the problem is. If a problem is hard to recreate or it is intermittent then it is worth the time to collect every possible piece of information.

Before contacting Xi Software Support or your distributor for help you should have the following information ready:

- Machine Model and Operating System
- Version numbers for the copy of **Xi-Text**. You can use the utility `whatvn` to display this, or `what` or `ident`.
- A copy of any error messages, especially the last few lines in the system log file `/usr/spool/spd/spshed_re`

The following sections describe some of the areas for investigation.

13.1 Files & Directories

One of the most common causes of problems for most products and platforms is changes to files, directory structures, ownerships or permissions. The effects of a missing, duplicate or modified file can be instantly fatal or very subtle.

Directories, files, permissions and ownerships should be as follows.

13.1.1 User Program Directory

The permissions of this directory and all of its parent directories must be sufficient for execution of the programs held in it. It is likely that many more files than just the **Xi-Text** user programs will be contained in this directory. The following `ls` command will display the required information for the relevant files:

```
ls -l sp* sq* sstop rspr xms*
```

Only the permissions, owner and file name are important. These should be:

Permissions	Owner	File
-rwsr-xr-x	spooler	rsprName
-rwsr-xr-x	spooler	sp.lpq
-rwsr-xr-x	spooler	sp.lprm
-rwsr-xr-x	spooler	spadd
-rwsr-xr-x	spooler	spchange
-rwsr-xr-x	spooler	spcharge
-rwsr-xr-x	spooler	spconn
-rwsr-xr-x	spooler	spdel
-rwsr-xr-x	spooler	spdisconn
-rwsr-xr-x	spooler	sphalt
-rwsr-xr-x	spooler	splist
-rwsr-xr-x	spooler	spnok
-rwsr-xr-x	spooler	spok
-rwsr-xr-x	spooler	spq
-rwsr-xr-x	spooler	spr
-rwsr-xr-x	spooler	spstart
-rwsr-xr-x	spooler	spstat
-rwsr-xr-x	spooler	spuchange
-rwsr-xr-x	spooler	spulist
-rwsr-xr-x	spooler	spuser
-rwsr-xr-x	spooler	sqchange
-rwsr-xr-x	spooler	sqdel
-rwsr-xr-x	spooler	sqlist
-rwsr-xr-x	spooler	sstop
-rwsr-xr-x	spooler	xmspq
-rwsr-xr-x	spooler	xmspuser

Some of these files may not be present if you are not using some options

13.1.2 The spd Internal Directory

The spd directory must be owned by user `spooler` and be accessible only by the users `spooler` and `root`. The permissions, owner etc look like this:

Permissions	Owner	File
drwx-----	spooler	spd

The contents of the `spd` directory must all belong to `spooler`, but the group ownership is not important. Permissions vary for each file and are very important. The files are:

Permissions	Owner	File
-r-----	spooler	SPnnnnnnnn
-rw-----	spooler	ERnnnnnnnn
-rw-----	spooler	PFnnnnnnnn
-rw-----	spooler	SPDEnnnnnnnn
-rw-----	spooler	spcharges23
-rw-----	spooler	sppm_jobd
-rw-----	spooler	sppm_jobi
-rw-----	spooler	sppm_ptr
-rw-----	spooler	sppm_xfer
-rw-----	spooler	spshed_jfile
-rw-----	spooler	spshed_reps
-rw-----	spooler	spshed_pfile
-rw-----	spooler	spufile23
drwx-----	spooler	xtlpc
drwx-----	spooler	xtlpd

The number on the end of the `spcharges` and `spufile` is the major version number for the installed copy of **Xi-Text**.

If any of these files have the wrong owner or permissions the spooler should be stopped using `sstop -y` before correcting them. Changing permissions or owners with the spooler running should only be done in consultation with Xi Software Support or your distributor.

13.1.3 The progs Internal Directory

The `progs` directory may be owned by user `spooler` or possibly user `batch` if **Xi-Batch** has also been installed, depending on which order they were installed.

Some of the **Xi-Text** files contained in the `progs` directory are owned by `spooler` and others by `root`. Having the correct owner and permissions is essential to the operation of the scheduler. They are:

Permissions	Owner	File
-rwxr-sr-x	spooler	dosspwrite
-rw-r--r--	spooler	int-config
-rwxr-xr-x	spooler	longlist
-rwxr-xr-x	spooler	psbanner
-rwxr-xr-x	spooler	ptr_alert
-rwxr-xr-x	spooler	remove
-rw-r--r--	spooler	rest.help
-rwxr-xr-x	spooler	sendudp
-rwxr-xr-x	spooler	shortlist
-rwsr-xr-x	spooler	spd
-rwxr-xr-x	spooler	spdinit
-rwsr-xr-x	root	spexec
-rwsr-xr-x	spooler	spjobdump
-rw-r--r--	spooler	splpd.help
-rwsr-xr-x	root	spmdisp
-rwsr-xr-x	root	sppwchk
-rw-r--r--	spooler	spq.help
-rwsr-xr-x	root	spshed
-rw-r--r--	spooler	spuser.help
-rwxr-sr-x	spooler	spwrite
-rwsr-xr-x	root	xilp
-rw-r--r--	spooler	xmspq.help
-rw-r--r--	spooler	xmspuser.help
-rwxr-xr-x	spooler	xtelnet
-rwsr-xr-x	root	xtlpc
-rw-r--r--	spooler	xtlpc-ctrl
-rwsr-xr-x	root	xtlpd
-rw-r--r--	spooler	xtlpd-ctrl
-rwsr-xr-x	root	xtlpq
-rwsr-xr-x	root	xtlprm
-rwsr-xr-x	root	xtnetserv

The files `xmspq.help` and `xmspuser.help` will only be present if the Motif GUI option is installed. The `.help` files can be owned by a user other than `spooler` but they must be readable by all users.

The spooler must be stopped with `sstop` before fixing owner or permissions of the executable files `spshed` and `spd`. The others can be fixed with the spooler running if it is absolutely essential, but you should check with Xi Software Support first. Using the commands `chmod`, `chgrp` and `chown` should be safe, but

any other operation may invalidate the license.

The `.help` files can be corrected without stopping the spooler.

13.1.4 The printers Internal Directory

The printers directory should have permissions and ownership looking like this:

```
Permissions Owner File
drwxr-xr-x spooler printers
```

The Printer Definition directories inside it should have the same permissions and ownerships like this:

```
Permissions Owner File
drwxr-xr-x spooler ljet
drwxr-xr-x spooler colour
```

They may have a different owner, but in this case user `spooler` must have sufficient permissions to access the files contained in them. The permissions can be removed for users other than `spooler` and `root` if this is required for security.

By default the files in the Printer Definition directories will be owned by `spooler` with read permissions for all users. For example:

```
Permissions Owner File
-rw-r--r-- spooler default
```

13.2 IPC Facilities

Xi-Text uses one message queue to communicate with the spooler process, `spshed`. Three shared memory segments (or memory-mapped files) are used to hold records of jobs and printers. These records are written out to the files `spshed_jfile` and `spshed_pfile` respectively, after changes occur. A further shared memory segment or memory-mapped file is used as buffer space for passing job details, as the size of messages which may be sent on message queues is limited on many systems. One semaphore set controls access to the shared memory segments.

On some versions of **Xi-Text** use of shared memory may be replaced by memory-mapped files and use of semaphores is usually replaced by file locking.

13.2.1 Looking at IPC Facilities

The IPC facilities can be recognised by running `ipcs`. The items in question are owned by `spooler` with a KEY of `0x58691xxx`. Using the command `ipcs -o` should produce an output listing resembling this:

```
# ipcs -o
IPC status from <running system> as of Thu Jan 4 12:27:52 1999
T      ID      KEY      MODE      OWNER      GROUP CBYTES  QNUM
```

```

Message Queues:
q      0 0x5869b000 -Rrw-----   batch    bin      0      0
q      1 0x58691000 -Rrw-----   spooler  staff    0      0

T      ID      KEY      MODE      OWNER     GROUP  NATTC
Shared Memory:
m      0 0x50018c83  --rw-r--r--  root     root    1
m      1 0x5869b003  --rw-----  batch    bin     4
m      2 0x5869b002  --rw-----  batch    bin     4
m      3 0x5869b100  --rw-----  batch    bin     4
m      4 0x58691002  --rw-----  spooler  staff   3
m      5 0x58691003  --rw-----  spooler  staff   3
m      6 0x58691004  --rw-----  spooler  staff   3

Semaphores:
s      0 0x5869b001  --ra-ra-ra-  batch    bin
s      1 0x5869b003  --ra-----  batch    bin
s      2 0x58691001  --ra-----  spooler  staff

```

In this example there are two third party software products using the IPC facilities, as well a shared memory area in use by the operating system. The lines of interest are the ones showing entries owned by user `spooler`. More recent versions of both products will not use shared memory or semaphores.

The entries owned by user `batch` are from another of our products, **Xi-Batch**. These are included to show that many users may be seen in the output from `ipcs`. If several products are using IPC facilities there will be more entries displayed than will fit on a screen. The entries appear in the order they were created, as time goes by existing entries will be removed and new ones created. Hence the entries will no longer be in easy to find blocks.

An easy way to see only the entries owned by `spooler` would be to pipe the output from the `ipcs` command through `grep`. For example:

```

# ipcs -o | grep spooler
q      1 0x58691000 -Rrw-----   spooler  staff    0      0
m      4 0x58691002  --rw-----  spooler  staff    3
m      5 0x58691003  --rw-----  spooler  staff    3
m      6 0x58691004  --rw-----  spooler  staff    3
s      2 0x58691001  --ra-----  spooler  staff

```

The type of each entry is shown in the first column. Message Queues have a letter "q" in the first column, shared memory areas have a letter "m" and semaphores a letter "s".

13.2.2 Problem - Cannot Start Xi-Text

When **Xi-Text** is halted normally it removes all of the IPC facilities which it is using. If terminated abnormally the spshd processes will still try to tidy up if possible.

The spooler cannot be restarted until all old **Xi-Text** processes have been killed and their IPC entries removed. An IPC entry cannot be deleted if there is still a process using or attached to it. This can be a program, like `xtnetserv`, `spq` or `splist`, not just `spshd`.

Possible Symptoms:

- Invoking `spshd` to start the spooler fails silently.

- Using `spstart` to start the scheduler fails with an error message about the IPC facilities, I/O or about files other than ones which are held in the spooler directory.

Investigation:

- If invoking `spshd`, run `spstart` instead to get an error message.
- Make sure that the spooler is stopped.
- Look for `spshd`, `xtnetd` and any user programs with a command like `ps`.
- Check the IPC facilities with the `ipcs -o` command.

Fix:

- Kill any `spshd`, `xtnetd` and user processes like `spq`. Do this first with just "kill", not with "kill -9".
- Check the IPC facilities, for entries owned by spooler, with the `ipcs -o` command again. Remove any that remain using the `ipcrm` command, or more simply by using the `xt-ripc` command
- With versions of **Xi-Text**, which use memory-mapped files, remove any outstanding undeleted ones with:

```
rm -f /usr/spool/spd/spmm*
```

- Run `spstart` to bring the spooler up, if this fails call Xi Software for support.

13.2.3 Deleting IPC entries owned by spooler

The `ipcrm` command may be used to delete Message Queues, Shared Memory Segments and Semaphores. It uses the same letters to identify Queues, Memory Segments and Semaphores as the `ipcs` command. If running `ipcs -o` gives a listing like this:

```
# ipcs -o | grep spooler
q      1 0x58691000 -Rrw----- spooler  staff      0      0
m      4 0x58691002 --rw----- spooler  staff      3
m      5 0x58691003 --rw----- spooler  staff      3
m      6 0x58691004 --rw----- spooler  staff      3
s      2 0x58691001 --ra----- spooler  staff
```

then suitable `ipcrm` commands to remove each entry would be:

```
# ipcrm -q 1
# ipcrm -m 4
# ipcrm -m 5
# ipcrm -m 6
# ipcrm -s 2
```

The flag in `ipcrm` indicates what type of IPC entry to delete, using the same letter as in the first column of the output from `ipcs -o`. A message queue is represented with a letter "q" and deleted with a `-q` option.

The number following the flag is an ID number for the queue, memory segment or semaphore. It is unique within the type of IPC, hence there can only be one Message Queue 0, but there can also be one Shared Memory Segment 0 and one Semaphore 0.

Most operating systems will allow more than one entry to be deleted with a single invocation of `ipcrm`. To delete all 5 entries in one go the command would be:

```
ipcrm -q 1 -m 4 -m 5 -m 6 -s 2
```

Be very careful not to delete the wrong IPC entry with `ipcrm`. It is very easy to make an error with system fatal results.

You may prefer to use the `xt-ripc` utility, which diagnoses problems as well as optionally deleting the shared memory facilities.

13.2.4 Processes

Some Unix installations have a very low figure specified for the maximum number of processes which may simultaneously be run by a user other than the super-user, `root`. The user `spooler` is subject to the same restrictions on numbers of processes as all other users apart from `root`. This may have an adverse effect on the number printers and associated processes which **Xi-Text** can run.

If this occurs, then **Xi-Text** will become unable to spawn new processes, such as start printers, or send emails or messages. In all cases appropriate messages should be written to the system log file, normally located in `/usr/spool/spd/spshed_reps`.

In such cases, please refer to your operating system manual for details of how to tune the limit on the number of processes per user.

As a rough guide to the number of processes allow 6 processes for Xi-text with all printers halted, plus:

- 1 process for each directly connected printer, plus transients when it is started up, or is running alignment process.
- 1 process for each filter command.
- 2 processes for each terminal server printer.

13.3 Messages about key clashes entering `spq` or `spuser`

If you receive a message like this:

```
State 5 setup error - clash on character ^H with previously-given value
420 and new value 530
```

It means that your help message file contains 2 definitions for the given character, in this case backspace (`^H`). You should look in the help message file (in the case of `spq` this will be by default `/usr/spool/progs/spq.help` and `spuser.help`) for lines of the form

```
K420: . . .
```

and

```
5K530: . . .
```

The "420", "5" and "530" are all parameters given in the message. You may find that the same character value is defined more than once; remember that (in this example) backspace could conceivably be specified from:

```
^H \b    An explicit character definition
\kERASE  Your terminal key settings for erase character etc
\kLEFT   The left cursor key as defined in terminfo
```

The installation procedure attempts to cover cases on the terminal on which you installed **Xi-Text**, but you may run into these difficulties if you use different terminals or key settings.

To overcome the difficulties you may want to rearrange the help message file, or possibly temporarily re-assign your terminal keys. Don't forget that you can arrange to automatically select different help message files depending upon the terminal you are using.

13.4 Warning messages about unknown key name

If you receive a message on entry to `spq` or `spuser` such as

```
Unknown key name 'khome' - ignored
```

then this means that your help message file contains a reference to a key not defined in your *terminfo* or *termcap* file, in this case the HOME key. The effect is harmless; however to get rid of the message you should either adjust the file (`spq.help` or `spuser.help`) to remove the reference, in this case to `\kHOME`, or to replace it with the correct character sequence, alternatively you should include the definition in your *terminfo* or *termcap* file.