

xisoftware

Xi-Text Release 23
System Reference Manual

Table of Contents

1 Introduction	21
1.1 Documentation Standards and Program Usage	22
1.2 Command Line Program Options	22
2 Overview	23
2.1 Directory and File Structure	24
2.1.1 Internal Directories	25
2.1.2 Printer Definitions and Setup Files	26
2.1.3 Help and message files	27
2.1.4 Spool directory files	28
2.1.5 Internal programs	29
2.1.6 User Programs	29
2.1.7 Configuration files held in /etc	29
2.1.7.1 Xi-Text Hosts File	29
2.1.7.1.1 Multiple IP addresses	31
2.1.7.2 Xi-Text Master Configuration File	31
2.1.7.3 User Mapping file	32
2.2 Standard Exit Codes	32
3 User Administration	35
3.1 Privileges	35
3.2 Priorities	38
3.3 Charging	39
3.4 Class codes	39
3.4.1 Treatment of class codes	39
3.4.2 User access to class codes.	40
4 Printer Definitions	42
4.1 Definitions	42
4.2 Form Types	43
4.3 Location and naming of setup files	44
4.3.1 Changing the name of the default file	45
4.3.2 The formname "standard"	45
4.4 Printer help files	45
4.5 Naming of files in the printer subdirectory	46
4.6 The .device file	46
4.7 Reading of setup files	46

4.8	What setup files provide	47
4.9	Format of setup files	47
4.10	Keywords in setup files	48
4.10.1	Keywords specific to devices	48
4.10.1.1	outbuffer keyword	49
4.10.1.2	open keyword	49
4.10.1.3	close keyword	50
4.10.1.4	postclose keyword	50
4.10.1.5	canhang keyword	50
4.10.1.6	reopen keyword	51
4.10.1.7	portsetup keyword	51
4.10.1.8	stty keyword	52
4.10.2	Keywords specific to serial devices	52
4.10.3	Keywords specific to network devices	54
4.10.3.1	Network keyword	54
4.10.3.2	fberror keyword	55
4.10.3.3	logerror keyword	55
4.10.3.4	closekill keyword	56
4.10.3.5	exit keyword	56
4.10.3.6	signal keyword	56
4.10.4	Keywords selecting spooling options	57
4.11	Setup strings	59
4.11.1	How to define setup strings	61
4.11.2	Executing strings	61
4.11.3	Format of strings	62
4.11.4	Symbolic names for strings	62
4.11.5	Default string setting	63
4.12	Filter commands	63
4.13	Header or banner pages	64
4.13.1	Whether header pages are printed	64
4.13.2	Format of banner pages	65
4.13.3	Tuning the standard banner	66
4.14	Alignment pages	67
4.15	Settings relating to page breaks	68
4.15.1	Delimiters	68
4.15.2	Windback pages	68
4.15.3	Inclpage1 keyword	69
4.16	Miscellaneous other facilities	69
4.16.1	Add Carriage Return	69
4.16.2	Single job mode	69
4.16.3	Retention of jobs	70
4.16.4	Onecopy	70
4.16.5	Charge keyword	70
4.16.6	Record	70
4.16.7	Log Files	71
4.17	Environment variables passed to commands	71
4.18	Interaction of filter and setup strings.	72

4.19	Avoiding a terminating formfeed	73
4.20	Conditions	73
4.21	Real setup files	75
5	Internal Programs and files	81
5.1	Core Internal Programs	81
5.1.1	Spshed	81
5.1.1.1	Files used by spshed	82
5.1.1.2	Environment variables relevant to spshed	82
5.1.1.3	IPC facilities used by spshed	82
5.1.1.4	Internet Ports used by spshed	83
5.1.1.5	Diagnostics from spshed	83
5.1.2	Xtnetserv	83
5.1.2.1	Files used by xtnetserv	83
5.1.2.2	Environment variables relevant to xtnetserv	84
5.1.2.3	Internet ports used by xtnetserv	84
5.1.2.4	Diagnostics from xtnetserv	84
5.1.3	Spd	84
5.1.3.1	Files used by spd	84
5.1.3.2	Environment variables relevant to spd	85
5.1.3.3	Diagnostics from spd	85
5.1.4	Spdinit	85
5.1.4.1	Files used by spdinit	85
5.1.4.2	Environment variables relevant to spdinit	85
5.1.4.3	Diagnostics from spdinit	86
5.1.5	Spjobdump	86
5.2	Message Handlers	86
5.2.1	Spmdisp	86
5.2.1.1	Files used by spmdisp	86
5.2.1.2	Environment variables relevant to spmdisp	87
5.2.2	Spwrite	87
5.2.2.1	Files used by spwrite	87
5.2.2.2	Environment variables relevant to spwrite	87
5.2.3	Dospwrite	87
5.2.3.1	Files used by dospwrite	87
5.2.3.2	Environment variables relevant to dospwrite	88
5.2.4	Spexec	88
5.3	File Formats	88
5.3.1	/etc/Xitext-config	88
5.3.1.1	Important notes for /etc/Xitext-config	88
5.3.2	/etc/Xitext-hosts	88
5.3.2.1	Host name	89
5.3.2.2	Alias name	89
5.3.2.3	Flags	89
5.3.2.4	Timeout	90
5.3.2.5	Local address	91
5.3.3	User map file	91

6	User Programs	92
6.1	Option Syntax	92
6.1.1	Configuration files	93
6.1.2	Option path	94
6.1.3	Message files	95
6.1.4	Location of message files	95
6.1.5	Path to locate message files	96
6.2	Submitting print jobs	97
6.2.1	spr	97
6.2.1.1	Options	97
6.2.1.1.1	-? or +explain option	97
6.2.1.1.2	-a or +mail-attention option	97
6.2.1.1.3	-A or +write-attention option	98
6.2.1.1.4	-b or +noatt-message option	98
6.2.1.1.5	-c or +copies option	98
6.2.1.1.6	-C or +classcode option	98
6.2.1.1.7	-d or +delimiter-number option	99
6.2.1.1.8	-D or +delimiter option	99
6.2.1.1.9	-E or +external-system option	99
6.2.1.1.10	-f or +formtype option	100
6.2.1.1.11	-F or +post-proc-flags option	100
6.2.1.1.12	-h or +header option	100
6.2.1.1.13	-i or +interpolate option	100
6.2.1.1.14	-l or +no-interpolate option	101
6.2.1.1.15	-j or +job-wait-time option	101
6.2.1.1.16	-l or +local-only option	101
6.2.1.1.17	-L or +network-wide option	102
6.2.1.1.18	-m or +mail-message option	102
6.2.1.1.19	-n or +delay-for option	102
6.2.1.1.20	-N or +delay-until option	102
6.2.1.1.21	-o or +originating-host option	103
6.2.1.1.22	-O or +odd-even-flags option	103
6.2.1.1.23	-p or +priority option	104
6.2.1.1.24	-P or +printer option	104
6.2.1.1.25	-q or +retain option	104
6.2.1.1.26	-Q or +host option	104
6.2.1.1.27	-r or +banner option	105
6.2.1.1.28	-R or +page-range option	105
6.2.1.1.29	-s or +no-banner option	105
6.2.1.1.30	-t or +printed-timeout option	105
6.2.1.1.31	-T or +not-printed-timeout option	106
6.2.1.1.32	-u or +post-user option	106
6.2.1.1.33	-U or +originating-user option	106
6.2.1.1.34	-V or +toggle-verbose option	106
6.2.1.1.35	+verbose option	107
6.2.1.1.36	+no-verbose option	107
6.2.1.1.37	-w or +write-message option	107

6.2.1.1.38	-x or +no-message option	107
6.2.1.1.39	-z or +no-retain option	107
6.2.1.1.40	-Z or +job-size-limit option	108
6.2.1.1.41	+freeze-current option	108
6.2.1.1.42	+freeze-home option	108
6.2.1.2	Environment variables used by spr	108
6.2.1.3	Important Note	108
6.2.1.4	Messages via terminal and e-mail	109
6.2.1.5	Queue Timeouts	109
6.2.1.6	Delay times	110
6.2.1.7	Interaction of spr and rspr	111
6.2.2	rspr	111
6.2.2.1	Options	111
6.2.2.1.1	-? or +explain option	112
6.2.2.1.2	-a or +mail-attention option	112
6.2.2.1.3	-A or +write-attention option	112
6.2.2.1.4	-b or +noatt-message option	112
6.2.2.1.5	-c or +copies option	112
6.2.2.1.6	-C or +classcode option	113
6.2.2.1.7	-d or +delimiter-number option	113
6.2.2.1.8	-D or +delimiter option	113
6.2.2.1.9	-f or +formtype option	114
6.2.2.1.10	-F or +post-proc-flags option	114
6.2.2.1.11	-h or +header option	114
6.2.2.1.12	-i or +interpolate option	114
6.2.2.1.13	-l or +no-interpolate option	115
6.2.2.1.14	-j or +job-wait-time option	115
6.2.2.1.15	-l or +local-only option	115
6.2.2.1.16	-L or +network-wide option	115
6.2.2.1.17	-m or +mail-message option	116
6.2.2.1.18	-n or +delay-for option	116
6.2.2.1.19	-N or +delay-until option	116
6.2.2.1.20	-O or +odd-even-flags option	117
6.2.2.1.21	-p or +priority option	117
6.2.2.1.22	-P or +printer option	117
6.2.2.1.23	-q or +retain option	118
6.2.2.1.24	-Q or +host option	118
6.2.2.1.25	-r or +banner option	118
6.2.2.1.26	-R or +page-range option	118
6.2.2.1.27	-s or +no-banner option	119
6.2.2.1.28	-t or +printed-timeout option	119
6.2.2.1.29	-T or +not-printed-timeout option	119
6.2.2.1.30	-u or +post-user option	119
6.2.2.1.31	-V or +toggle-verbose option	120
6.2.2.1.32	+verbose option	120
6.2.2.1.33	+no-verbose option	120
6.2.2.1.34	-w or +write-message option	120

6.2.2.1.35	-x or +no-message option	121
6.2.2.1.36	-z or +no-retain option	121
6.2.2.1.37	-Z or +job-size-limit option	121
6.2.2.1.38	+freeze-current option	121
6.2.2.1.39	+freeze-home option	122
6.2.2.2	Environment variables used by rspr	122
6.2.3	lpcover	122
6.2.3.1	Options	122
6.2.3.1.1	-c option	122
6.2.3.1.2	-d option	122
6.2.3.1.3	-f option	123
6.2.3.1.4	-H option	123
6.2.3.1.5	-m option	123
6.2.3.1.6	-n option	123
6.2.3.1.7	-o option	123
6.2.3.1.8	-P option	123
6.2.3.1.9	-q option	124
6.2.3.1.10	-S option	124
6.2.3.1.11	-s option	124
6.2.3.1.12	-T option	124
6.2.3.1.13	-t option	124
6.2.3.1.14	-w option	124
6.2.3.1.15	-y option	125
6.2.3.2	Environment variables used	125
6.2.3.3	Diagnostics	125
6.3	Managing the spooler	125
6.3.1	spstart – for system	125
6.3.2	sstop	125
6.3.2.1	Options for sstop	126
6.3.2.1.1	-y option	126
6.3.3	spconn	126
6.3.4	spdisconn	126
6.3.5	ssuspend	126
6.3.5.1	Options	127
6.3.5.1.1	-y option	127
6.3.6	srelease	127
6.3.7	hostedit	127
6.3.7.1	Options	128
6.3.7.1.1	-o option	128
6.3.7.1.2	-s option	128
6.3.7.1.3	-l option	128
6.3.7.2	Commands	128
6.4	Managing print jobs (command line)	129
6.4.1	sqlist	129
6.4.1.1	Options	129
6.4.1.1.1	-? or +explain option	130
6.4.1.1.2	-C or +classcode option	130

6.4.1.1.3	-D or +default-format option	130
6.4.1.1.4	-e or +no-page-counts option	130
6.4.1.1.5	-E or +page-counts option	131
6.4.1.1.6	-F or +format option	131
6.4.1.1.7	-H or +header option	131
6.4.1.1.8	-l or +local-only option	131
6.4.1.1.9	-N or +no-header option	132
6.4.1.1.10	-n or +no-view-job option	132
6.4.1.1.11	-P or +view-pages option	132
6.4.1.1.12	-p or +printed-jobs option	132
6.4.1.1.13	-q or +list-only option	133
6.4.1.1.14	-r or +remotes option	133
6.4.1.1.15	-t or +just-title option	133
6.4.1.1.16	-u or +just-user option	133
6.4.1.1.17	-V or +view-job option	133
6.4.1.1.18	-Y or +all-jobs option	134
6.4.1.1.19	-y or +unprinted-jobs option	134
6.4.1.1.20	-z or +include-null option	134
6.4.1.1.21	-Z or +no-include-null option	134
6.4.1.1.22	+freeze-current option	134
6.4.1.1.23	+freeze-home option	135
6.4.1.2	Patterns	135
6.4.1.3	Format codes	135
6.4.2	sqdel	137
6.4.2.1	Options	137
6.4.2.1.1	-? or +explain option	137
6.4.2.1.2	-C or +classcode option	137
6.4.2.1.3	-d or +delete option	138
6.4.2.1.4	-D or +directory option	138
6.4.2.1.5	-e or +do-not-unqueue option	138
6.4.2.1.6	-J or +job-prefix option	138
6.4.2.1.7	-k or +do-not-delete option	138
6.4.2.1.8	-n or +keep-unprinted option	139
6.4.2.1.9	-S or +command-prefix option	139
6.4.2.1.10	-u or +unqueue option	139
6.4.2.1.11	-y or +delete-unprinted option	139
6.4.2.1.12	+freeze-current option	139
6.4.2.1.13	+freeze-home option	140
6.4.2.2	NOTES	140
6.4.3	sqchange	140
6.4.3.1	Options	140
6.4.3.1.1	-? or +explain option	141
6.4.3.1.2	-a or +mail-attention option	141
6.4.3.1.3	-A or +write-attention option	141
6.4.3.1.4	-b or +noatt-message option	141
6.4.3.1.5	-c or +copies option	142
6.4.3.1.6	-C or +classcode option	142

6.4.3.1.7	-f or +formtype option	142
6.4.3.1.8	-F or +post-proc-flags option	142
6.4.3.1.9	-h or +header option	143
6.4.3.1.10	-l or +local-only option	143
6.4.3.1.11	-L or +network-wide option	143
6.4.3.1.12	-m or +mail-message option	143
6.4.3.1.13	-n or +delay-for option	144
6.4.3.1.14	-N or +delay-until option	144
6.4.3.1.15	-O or +odd-even-flags option	144
6.4.3.1.16	-p or +priority option	145
6.4.3.1.17	-P or +printer option	145
6.4.3.1.18	-q or +retain option	146
6.4.3.1.19	-r or +banner option	146
6.4.3.1.20	-R or +page-range option	146
6.4.3.1.21	-s or +no-banner option	146
6.4.3.1.22	-S or +set-classcode option	147
6.4.3.1.23	-t or +printed-timeout option	147
6.4.3.1.24	-T or +not-printed-timeout option	147
6.4.3.1.25	-u or +post-user option	147
6.4.3.1.26	-w or +write-message option	148
6.4.3.1.27	-x or +no-message option	148
6.4.3.1.28	-z or +no-retain option	148
6.4.3.1.29	+freeze-current option	148
6.4.3.1.30	+freeze-home option	149
6.4.3.2	Notes	149
6.4.3.3	Examples	149
6.4.4	Lpstatc	149
6.5	Managing printers (command line)	150
6.5.1	splist	150
6.5.1.1	Options	150
6.5.1.1.1	-? or +explain option	150
6.5.1.1.2	-C or +classcode option	151
6.5.1.1.3	-D or +default-format option	151
6.5.1.1.4	-F or +format option	151
6.5.1.1.5	-H or +header option	151
6.5.1.1.6	-l or +local-only option	152
6.5.1.1.7	-N or +no-header option	152
6.5.1.1.8	-q or +list-only option	152
6.5.1.1.9	-r or +remotes option	152
6.5.1.1.10	-S or +sort-ptrs option	153
6.5.1.1.11	-U or +no-sort-ptrs option	153
6.5.1.1.12	+freeze-current option	153
6.5.1.1.13	+freeze-home option	153
6.5.1.2	Format codes	153
6.5.2	spstart	154
6.5.2.1	Options	155
6.5.2.1.1	-? or +explain option	155

6.5.2.1.2	-l or +device-name option	156
6.5.2.1.3	-C or +classcode option	156
6.5.2.1.4	-f or +force-all option	156
6.5.2.1.5	-n or +no-force option	156
6.5.2.1.6	-W or +wait-complete option	156
6.5.2.1.7	-E or +no-wait option	157
6.5.2.1.8	+freeze-current option	157
6.5.2.1.9	+freeze-home option	157
6.5.3	sphalt and spstop	157
6.5.3.1	Options	158
6.5.3.1.1	-? or +explain option	158
6.5.3.1.2	-l or +device-name option	158
6.5.3.1.3	-C or +classcode option	158
6.5.3.1.4	-f or +force-all option	158
6.5.3.1.5	-n or +no-force option	159
6.5.3.1.6	-W or +wait-complete option	159
6.5.3.1.7	-E or +no-wait option	159
6.5.3.1.8	+freeze-current option	159
6.5.3.1.9	+freeze-home option	159
6.5.4	spchange	160
6.5.4.1	Options	160
6.5.4.1.1	-? or +explain option	160
6.5.4.1.2	-N or +network-device option	160
6.5.4.1.3	-L or +line-device option	160
6.5.4.1.4	-s or +local-only option	161
6.5.4.1.5	-w or +network-wide option	161
6.5.4.1.6	-l or +device-name option	161
6.5.4.1.7	-C or +classcode option	161
6.5.4.1.8	-D or +description option	161
6.5.4.1.9	-v or +new-device option	162
6.5.4.1.10	-S or +set-classcode option	162
6.5.4.1.11	-f or +force-all option	162
6.5.4.1.12	-n or +no-force option	162
6.5.4.1.13	-W or +wait-complete option	162
6.5.4.1.14	-E or +no-wait option	163
6.5.4.1.15	+freeze-current option	163
6.5.4.1.16	+freeze-home option	163
6.5.5	spadd and spdel	163
6.5.5.1	Options	164
6.5.5.1.1	-? or +explain option	164
6.5.5.1.2	-N or +network-device option	164
6.5.5.1.3	-L or +line-device option	164
6.5.5.1.4	-s or +local-only option	165
6.5.5.1.5	-w or +network-wide option	165
6.5.5.1.6	-l or +device-name option	165
6.5.5.1.7	-C or +classcode option	165
6.5.5.1.8	-D or +description option	165

6.5.5.1.9	-W or +wait-complete option	166
6.5.5.1.10	-E or +no-wait option	166
6.5.5.1.11	+freeze-current option	166
6.5.5.1.12	+freeze-home option	166
6.5.6	spinter	166
6.5.6.1	Options	167
6.5.6.1.1	-? or +explain option	167
6.5.6.1.2	-l or +device-name option	167
6.5.6.1.3	-C or +classcode option	167
6.5.6.1.4	-f or +force-all option	168
6.5.6.1.5	-n or +no-force option	168
6.5.6.1.6	-W or +wait-complete option	168
6.5.6.1.7	-E or +no-wait option	168
6.5.6.1.8	+freeze-current option	168
6.5.6.1.9	+freeze-home option	169
6.5.7	spok and spnok	169
6.5.7.1	Options	169
6.5.7.1.1	-? or +explain option	170
6.5.7.1.2	-l or +device-name option	170
6.5.7.1.3	-C or +classcode option	170
6.5.7.1.4	-f or +force-all option	170
6.5.7.1.5	-n or +no-force option	170
6.5.7.1.6	-W or +wait-complete option	171
6.5.7.1.7	-E or +no-wait option	171
6.5.7.1.8	+freeze-current option	171
6.5.7.1.9	+freeze-home option	171
6.5.8	spstat	171
6.5.8.1	Standard state names for spstat	172
6.5.8.2	Options	172
6.5.8.2.1	-? or +explain option	172
6.5.8.2.2	-l or +device-name option	173
6.5.8.2.3	-C or +classcode option	173
6.5.8.2.4	+freeze-current option	173
6.5.8.2.5	+freeze-home option	173
6.5.9	Ptrinstall	173
6.5.10	netwcmd	174
6.5.11	getsntp	174
6.5.11.1	Options	175
6.5.11.1.1	-h option	175
6.5.11.1.2	-H option	175
6.5.11.1.3	-p option	175
6.5.11.1.4	-C option	175
6.5.11.1.5	-d option	175
6.5.11.1.6	-T option	176
6.5.11.1.7	-f option	176
6.5.11.1.8	-N option	176
6.5.11.1.9	-l option	176

6.6	Interactive job and printer administration	176
6.6.1	spq	176
6.6.1.1	Options	177
6.6.1.1.1	-? or +explain option	177
6.6.1.1.2	-A or +no-confirm-abort option	177
6.6.1.1.3	-a or +confirm-abort option	177
6.6.1.1.4	-B or +no-help-box option	178
6.6.1.1.5	-b or +help-box option	178
6.6.1.1.6	-C or +classcode option	178
6.6.1.1.7	-D or +printed-jobs option	178
6.6.1.1.8	-d or +dont-care option	178
6.6.1.1.9	-E or +page-counts option	179
6.6.1.1.10	-e or +no-page-counts option	179
6.6.1.1.11	-H or +keep-char-help option	179
6.6.1.1.12	-h or +lose-char-help option	179
6.6.1.1.13	-I or +include-all option	179
6.6.1.1.14	-j or +jobs-screen option	180
6.6.1.1.15	-l or +local-only option	180
6.6.1.1.16	-M or +no-error-box option	180
6.6.1.1.17	-m or +error-box option	180
6.6.1.1.18	-n or +follow-job option	180
6.6.1.1.19	-P or +number-printers option	181
6.6.1.1.20	-p or +printer-screen option	181
6.6.1.1.21	-q or +list-only option	181
6.6.1.1.22	-R or +refresh-time option	181
6.6.1.1.23	-r or +remotes option	182
6.6.1.1.24	-S or +sort-ptrs option	182
6.6.1.1.25	-s or +keep-cursor option	182
6.6.1.1.26	-t or +just-title option	182
6.6.1.1.27	-u or +no-sort-ptrs option	182
6.6.1.1.28	-u or +just-user option	183
6.6.1.1.29	-V or +all-jobs option	183
6.6.1.1.30	-y or +unprinted-jobs option	183
6.6.1.1.31	-Z or +no-include-null option	183
6.6.1.1.32	-z or +include-null option	183
6.6.2	xspq and xmspq	184
6.7	User administration and charging	184
6.7.1	spulist	184
6.7.1.1	Options	184
6.7.1.1.1	-? or +explain option	184
6.7.1.1.2	-D or +default-format option	184
6.7.1.1.3	-d or +default-line option	185
6.7.1.1.4	-F or +format option	185
6.7.1.1.5	-H or +header option	185
6.7.1.1.6	-N or +no-header option	185
6.7.1.1.7	-n or +numeric-sort option	186
6.7.1.1.8	-S or +no-user-lines option	186

6.7.1.1.9	-s or +no-default-line option	186
6.7.1.1.10	-U or +user-lines option	186
6.7.1.1.11	-u or +user-sort option	186
6.7.1.1.12	+freeze-current option	187
6.7.1.1.13	+freeze-home option	187
6.7.1.2	Format argument	187
6.7.1.3	Privileges format	188
6.7.2	spuchange	188
6.7.2.1	Options	189
6.7.2.1.1	-? or +explain option	189
6.7.2.1.2	-A or +copy-defaults option	189
6.7.2.1.3	-c or +class option	189
6.7.2.1.4	-D or +set-defaults option	190
6.7.2.1.5	-d or +default-priority option	190
6.7.2.1.6	-F or +form-allowed option	190
6.7.2.1.7	-f or +default-form option	190
6.7.2.1.8	-l or +min-priority option	190
6.7.2.1.9	-m or +max-priority option	191
6.7.2.1.10	-N or +no-rebuild option	191
6.7.2.1.11	-n or +max-copies option	191
6.7.2.1.12	-O or +ptr-allowed option	191
6.7.2.1.13	-o or +default-ptr option	191
6.7.2.1.14	-p or +privileges option	192
6.7.2.1.15	-R or +rebuild-file option	192
6.7.2.1.16	-s or +no-copy-defaults option	192
6.7.2.1.17	-u or +set-users option	192
6.7.2.1.18	-X or +dump-passwd option	192
6.7.2.1.19	-Y or +default-passwd option	193
6.7.2.1.20	-Z or +kill-dump-passwd option	193
6.7.2.1.21	+freeze-current option	193
6.7.2.1.22	+freeze-home option	193
6.7.2.2	Users or default	193
6.7.2.3	Rebuilding the user control file	194
6.7.2.4	Dumping the password file	194
6.7.2.5	Privileges	194
6.7.3	spuser	195
6.7.3.1	Options	195
6.7.3.1.1	-? or +explain option	195
6.7.3.1.2	-B or +no-help-box option	196
6.7.3.1.3	-b or +help-box option	196
6.7.3.1.4	-c or +form-and-prio option	196
6.7.3.1.5	-d or +display option	196
6.7.3.1.6	-H or +keep-char-help option	196
6.7.3.1.7	-h or +lose-char-help option	197
6.7.3.1.8	-i or +administer option	197
6.7.3.1.9	-M or +no-error-box option	197
6.7.3.1.10	-m or +error-box option	197

6.7.3.1.11	-n or +numeric-sort option	197
6.7.3.1.12	-u or +user-sort option	198
6.7.4	xspuser and xmspuser	198
6.7.5	spcharge	198
6.8	Web browser interface support	198
6.9	Spool directory management	199
6.9.1	xt-cjlist	199
6.9.1.1	Options	200
6.9.1.1.1	-D option	200
6.9.1.1.2	-e option	200
6.9.1.1.3	-f option	200
6.9.1.1.4	-s option	200
6.9.1.1.5	-u option	200
6.9.1.1.6	-v option	201
6.9.2	xt-cplist	201
6.9.2.1	Options	201
6.9.2.1.1	-D option	201
6.9.2.1.2	-e option	202
6.9.2.1.3	-f option	202
6.9.2.1.4	-s option	202
6.9.2.1.5	-v option	202
6.9.3	xt-spuconv	202
6.9.3.1	Options	203
6.9.3.1.1	-D option	203
6.9.3.1.2	-e option	203
6.9.3.1.3	-f option	203
6.9.3.1.4	-s option	203
6.9.3.1.5	-v option	204
6.9.4	xt-ripc	204
6.9.4.1	Options	204
6.9.4.1.1	-A option	204
6.9.4.1.2	-D option	204
6.9.4.1.3	-d option	205
6.9.4.1.4	-f option	205
6.9.4.1.5	-n option	205
6.9.4.1.6	-o option	205
6.9.4.1.7	-P option	205
6.9.4.1.8	-G option	205
6.9.4.1.9	-r option	206
6.9.4.1.10	-S option	206
6.9.4.1.11	-x option	206
6.9.4.1.12	-B option	206
6.9.4.1.13	-N option	206
6.9.4.2	Example	206
6.9.5	setspdir	207
6.9.5.1	Options	207
6.9.5.1.1	-d option	207

6.9.5.1.2	-f option	207
6.9.5.1.3	-r option	207
6.9.5.1.4	-s option	208
6.9.5.1.5	-y option	208
6.9.6	xipasswd	208
6.9.6.1	Options	208
6.9.6.1.1	-u option	208
6.9.6.1.2	-p option	209
6.9.6.1.3	-f option	209
6.9.6.1.4	-d option	209
6.9.6.1.5	-F option	209
6.10	Licence, installation, version control	209
6.10.1	xt-vwrite	209
6.10.1.1	Options	210
6.10.1.1.1	-L option	210
6.10.1.1.2	-N option	210
6.10.1.1.3	-l option	210
6.10.1.1.4	-n option	210
6.10.2	xt-checklic	210
6.10.2.1	Diagnostics	211
6.10.3	xt-emerglic	211
6.10.3.1	Options	211
6.10.3.1.1	-L option	211
6.10.3.1.2	-N option	211
6.10.4	xt-triallic	211
6.10.5	whatvn	212
6.10.5.1	Options	212
6.10.5.1.1	-P option	212
6.10.5.1.2	-l option	213
6.10.5.1.3	-D option	213
6.10.5.1.4	-d option	213
6.10.5.1.5	-v option	213
6.10.5.1.6	-q option	213
6.10.5.1.7	-t option	213
6.10.5.1.8	-F option	214
6.10.6	zinstall	214
6.10.6.1	Options	215
6.10.6.1.1	-l option	215
6.10.6.1.2	-V option	215
6.10.6.1.3	-C option	215
6.10.6.1.4	-U option	215
6.10.6.1.5	-t option	215
6.10.6.1.6	-D option	216
6.10.6.1.7	-O option	216
6.10.6.1.8	-M option	216
6.10.6.1.9	-p option	217
6.10.6.1.10	-N option	217

6.10.6.1.11-S option	217
6.10.6.1.12-m option	217
6.10.6.1.13-u option	218
6.10.6.1.14-g option	218
6.10.6.1.15-f option	218
6.10.6.1.16-v option	218
6.10.6.1.17-q option	218
7 Text screen-based Programs	219
7.1 spq – interactive spool queue manager	219
7.1.1 Position of cursor on entry	219
7.1.2 Screen-based commands - cursor movement	219
7.1.3 Field entry	220
7.1.4 Set and save program options	221
7.1.4.1 Restricting the display by printer	221
7.1.4.2 Restricting the display by user	222
7.1.5 Job queue section.	222
7.1.5.1 Abort job	223
7.1.5.2 Change the number of copies	224
7.1.5.3 Change the form type	224
7.1.5.4 Priorities	224
7.1.5.5 Change the printer name	225
7.1.5.6 Unqueue job	225
7.1.5.7 Other options	225
7.1.5.8 View job file	227
7.1.5.9 Inspect system error log file	228
7.1.6 Reassign format of job list display	228
7.1.7 Printer list section of spq	230
7.1.7.1 Stop/start commands	232
7.1.7.2 Printer size limits	233
7.1.7.3 Restarting halted jobs	233
7.1.7.4 Abort/restart current job	233
7.1.7.5 Change printer name	234
7.1.7.6 Change form type	234
7.1.7.7 Delete printer	234
7.1.7.8 Add a printer	234
7.1.7.8.1 FIFOs as printers	235
7.1.7.8.2 Terminal servers as printers	235
7.1.7.9 Reset class code	235
7.1.7.10 Operator attention commands	235
7.1.8 Reassign format of printer list display	236
7.2 spuser – user administration tool	237
7.2.1 spuser enquiry	238
7.2.2 spuser -c	238
7.2.3 spuser -i	238
7.2.3.1 Resetting system-wide defaults	240
7.2.3.2 Setting default privileges	241

7.2.3.3	Setting default class	242
7.2.3.4	Setting individual user privileges	242
8	X/Motif Programs	244
8.1	xmspq – Motif GUI Spool Queue Tool	244
8.1.1	Options	244
8.1.2	Useful Resources	244
8.1.3	Examples	246
8.1.4	The Main Window	246
8.1.4.1	The Options Menu	248
8.1.4.2	The Action Menu & Buttons	248
8.1.4.3	The Jobs Menu & Buttons	249
8.1.4.4	The Printers Menu & Buttons	249
8.1.4.5	The Search Menu	250
8.1.4.6	Help	250
8.1.5	Setting the View Options	250
8.1.5.1	Setting the Confirmation level	251
8.1.5.2	Restricting the display	252
8.1.5.2.1	Restricting the display to the local host	252
8.1.5.2.2	Restricting the display by job status	252
8.1.5.2.3	Restricting the display by class code	252
8.1.5.2.3.1	Restricting the display by user & printer	253
8.1.5.3	Changing the fields displayed and their format	254
8.1.5.3.1	Changing the Job Display	254
8.1.5.3.1.1	Changing the Printer Display	256
8.1.5.4	Saving the Format Changes	256
8.1.5.5	Saving the View Options (excluding Format Changes)	256
8.1.6	Viewing a Print Job	257
8.1.7	Changing Job and Printer parameters	258
8.2	xmspuser – Motif GUI User Administration Tool	260
8.2.1	Options	260
8.2.2	Useful Resources	261
8.2.3	Examples	261
8.2.4	The Main Window	261
8.2.4.1	The Options Menu	262
8.2.4.2	The Defaults Menu	263
8.2.4.3	The Users Menu	263
8.2.4.4	The Charges Menu	264
8.2.4.5	The Help Menu	264
8.2.5	Selecting multiple users for menu options	264
8.2.6	Copying defaults to all users	265
8.2.7	Resetting a user to the default	265
9	Terminal Server support commands	266
9.1	Xtlpc	266
9.1.1	Options for xtlpc	266
9.1.1.1	-H option	266
9.1.1.2	-f option	267

9.1.1.3	-S option	267
9.1.1.4	-P option	267
9.1.1.5	-F option	267
9.1.1.6	-N option	267
9.1.1.7	-U option	268
9.1.1.8	-I option	268
9.1.1.9	-L option	268
9.1.1.10	-I option	268
9.1.1.11	-O option	268
9.1.1.12	-R option	268
9.1.1.13	-s option	269
9.1.2	Exit codes from xtlpc	269
9.1.3	Format of control file	270
9.1.3.1	Description of LPD Protocol	270
9.1.3.2	Multiple copies	271
9.2	Sp.lpq	271
9.2.1	Options	272
9.2.1.1	-? or +explain option	272
9.2.1.2	-I option	272
9.2.1.3	-s option	272
9.2.1.4	+freeze-current option	272
9.2.1.5	+freeze-home option	272
9.2.2	sp.lprm	273
9.3	Xilp	273
9.3.1	Options	273
9.3.1.1	-H option	273
9.3.1.2	-p option	273
9.3.1.3	-h option	273
9.3.1.4	-r option	274
9.3.1.5	-# option	274
9.3.1.6	-u option	274
9.3.1.7	-P option	274
9.3.1.8	-C option	274
9.3.1.9	-J option	274
9.3.1.10	-F option	275
9.4	Xtelnet	275
9.4.1	Options	275
9.4.1.1	-h option	275
9.4.1.2	-p option	275
9.4.1.3	-I option	275
9.4.1.4	-L option	276
9.4.1.5	-t option	276
9.4.1.6	-s option	276
9.4.1.7	-d option	276
9.4.2	Exit codes	276
9.5	Xtlhp	277
9.5.1	Options	277

9.5.1.1	-h option	277
9.5.1.2	-p option	277
9.5.1.3	-f option	277
9.5.1.4	-c option	277
9.5.1.5	-l option	278
9.5.1.6	-b option	278
9.5.1.7	-d option	278
9.5.1.8	-D option	278
9.5.1.9	-c option	278
9.5.1.10	-T option	279
9.5.1.11	-S option	279
9.5.2	Configuration file	279
9.5.3	Control file format	279
9.5.3.1	Expressions	280
9.5.3.2	Constructs	280
9.5.3.2.1	Assignments	281
9.5.3.2.2	Comparisons and tests	281
9.5.3.3	Example	282
9.6	Xtftp	283
9.6.1	Options	283
9.6.1.1	-h option	283
9.6.1.2	-A option	283
9.6.1.3	-p option	283
9.6.1.4	-u option	283
9.6.1.5	-w option	284
9.6.1.6	-o option	284
9.6.1.7	-D option	284
9.6.1.8	-T option	284
9.6.1.9	-i option	284
9.6.1.10	-t option	284
9.6.1.11	-b option	285
9.6.1.12	-d option	285
9.6.2	Exit Codes	285
10	Configuration of user interfaces	287
10.1	Configuration files	287
10.1.1	Environment variable or keyword names	288
10.2	User reconfiguration	289
10.2.1	Message files	290
10.2.1.1	File format	291
10.2.1.2	Key definitions	292
10.2.1.2.1	Special key sequences	293
10.2.1.3	Help and error messages	295
10.2.1.4	Option syntax	296
10.2.1.5	Prompts	297
10.2.1.6	Numeric parameters	297
10.2.1.7	Titles	298

10.2.1.7.1 Enhancements and line drawing in headers	298
10.3 Changing message files	300
10.4 Local help files	300
10.5 Variation of search order	301
10.5.1 Search order for message files	301
10.5.2 Search order for program options	302
10.5.3 Freezing options	302
11 Extension of user interfaces	303
11.1 Message Handling	303
11.1.1 Overriding mail and write options	303
11.1.1.1 Global overrides	304
11.1.1.2 Per-user overrides	304
11.1.2 Customising printer error handling	305
11.2 Macros for Interactive User Programs	305
11.2.1 Inserting the commands	305
11.2.2 Binding the keys	306
11.2.3 Example	306
12 XTLPD	308
12.1 Format of XTLPD control file	308
12.1.1 Assignment format	309
12.1.2 System Variables	310
12.1.3 Handling Repeats	310
12.1.4 Beginning and Ending actions	311
12.1.5 Internal commands	311
12.2 Example Control File	312
12.3 Example short and long list commands	314
12.3.1 Lpq short format	314
12.3.2 Lpq long format	314
12.4 External Queue Specification	315
12.5 External queue definition file	315

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
- 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 - this manual. A complete description of all components of the basic product.
- Administrator Guide. Information about installation and customisation of the software.

Also available are:

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

1.1 Documentation Standards and Program Usage

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 an appropriate value)
 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
```

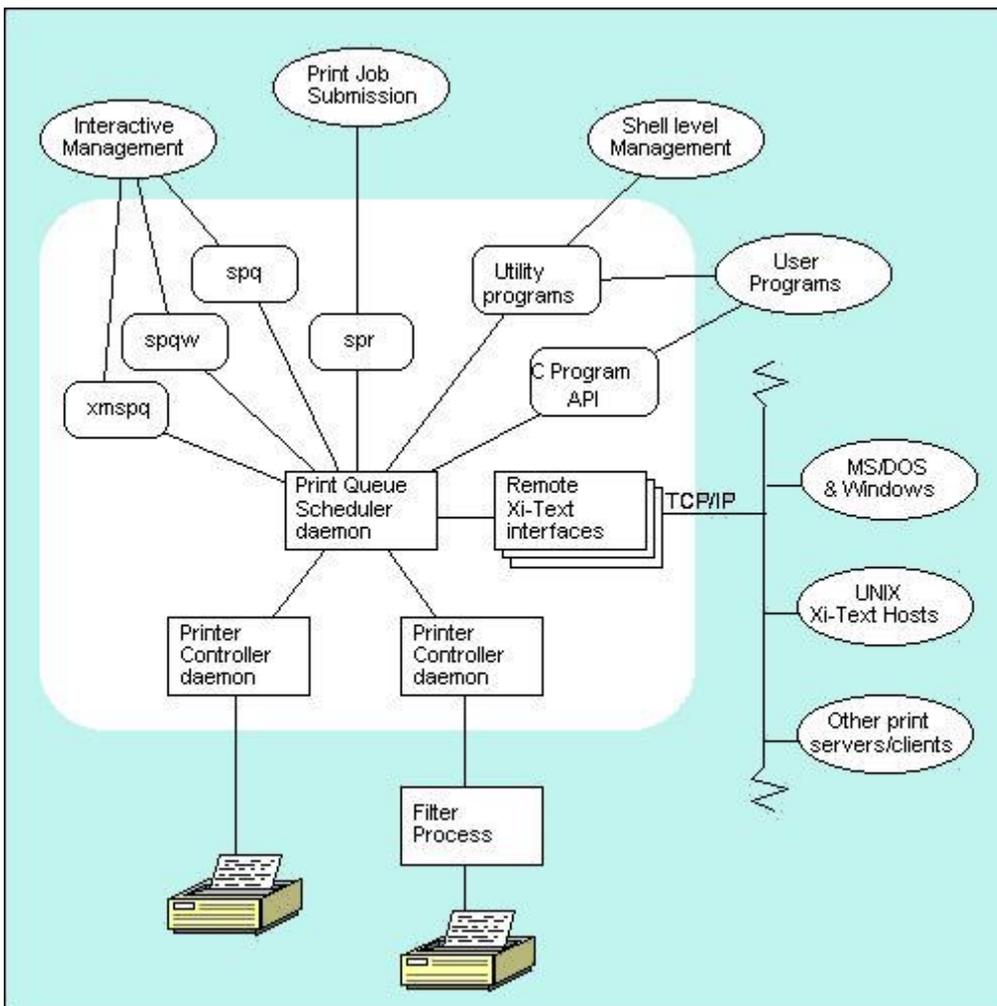
In addition, all the commands have an option `-?` or `+explain` (or `--explain`) whose function is to list all the other options and exit.

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

Chapter 2

Overview

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



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

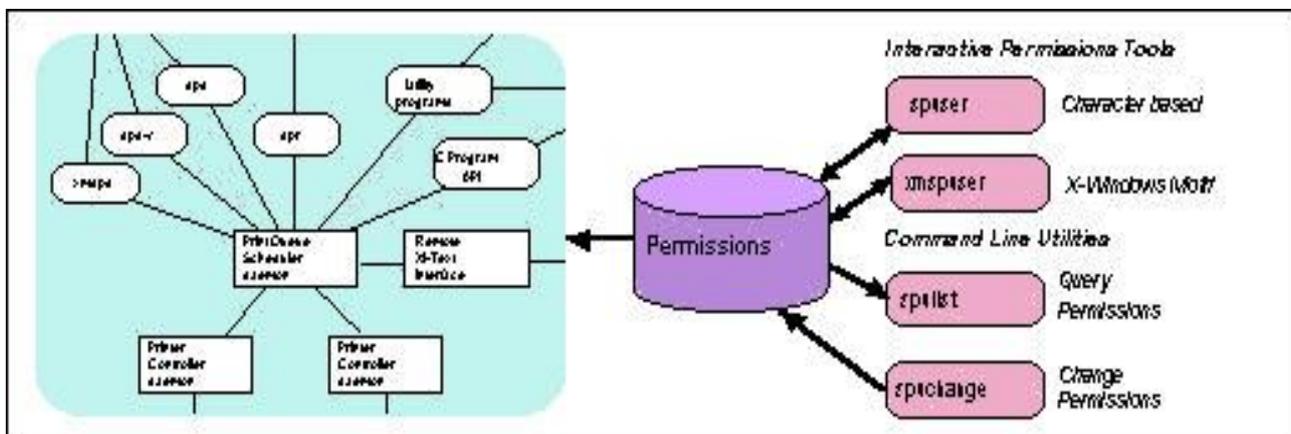
Each printer that is currently enabled to print is managed by an instance of the printer controller or “de-spooler” daemon `spd`. This daemon may communicate directly with a device driver or some intermediate program, as required. Such intermediate programs, or filter processes, may be used for special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

Jobs can be submitted to the print queue for printing using the programs `spr` and `lpcover` (our version of the standard `lp` command). Windows PCs and machines that send jobs to remote lp servers can also submit jobs to **Xi-Text**. These facilities are available via our Windows Client PC Option, described in a separate manual, and the LPD server interface, described in chapter 12.

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

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

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



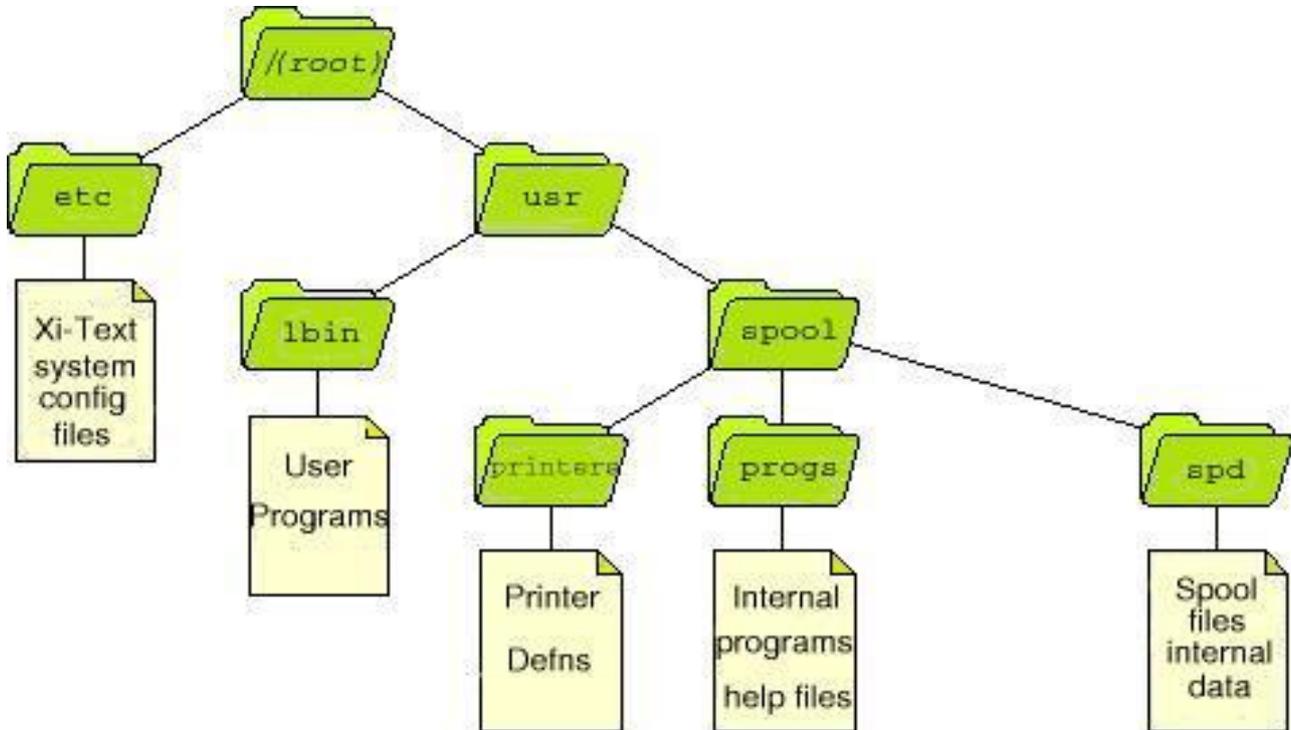
2.1 Directory and File Structure

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

- Global configuration files are always held in the `/etc` directory.
- User programs can be placed in any directory which is on the **Xi-Text** users' `PATH` variable.

- Internal programs and data are held in three or sometimes four separate directories.

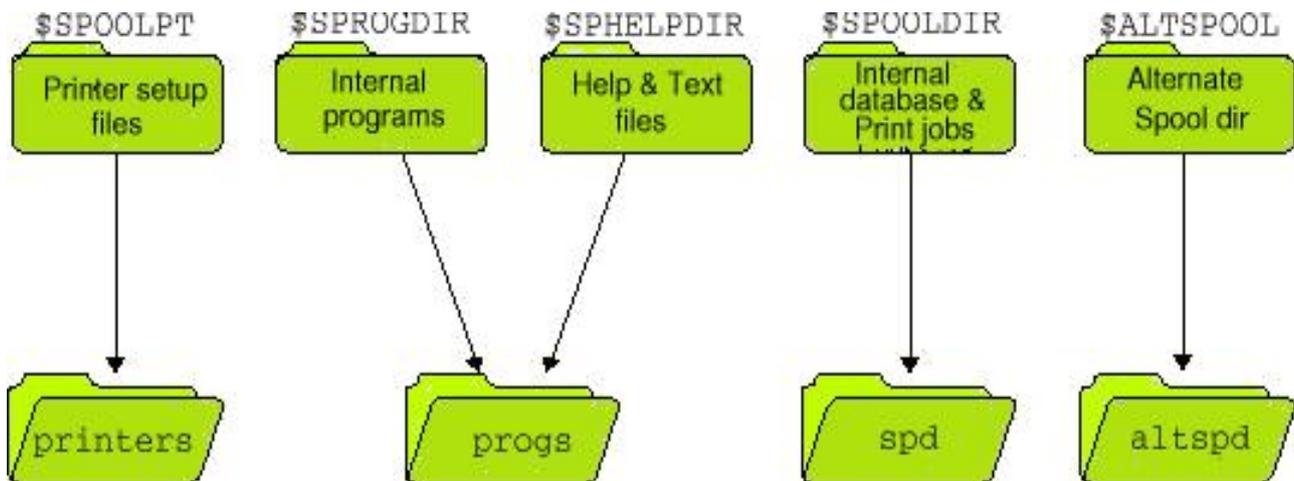
If your public program directory is `lbin` (some machines use `/usr/local/bin`) and the spool directory is under `/usr` (some are under `/var`), then the default installation will look like this:



It is normal (and advisable) to leave the directory containing the Installation Kit in a filesystem somewhere as well. This gives easy access to the original copies of configuration files and utilities and the validation programs for entry of new license codes. We recommend that this be placed elsewhere than from the live installation so that it is backed-up separately.

2.1.1 Internal Directories

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



These directories may be relocated by assignment to the five environment variables: `SPOOLPT`, `SPROGDIR`, `SPHELDIR` and `SPOOLDIR`. These environment variable assignments should be placed in the *master configuration file*, `/etc/Xitext-config`, to ensure consistency.

The 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>SPHELDIR</code>	Help and message files

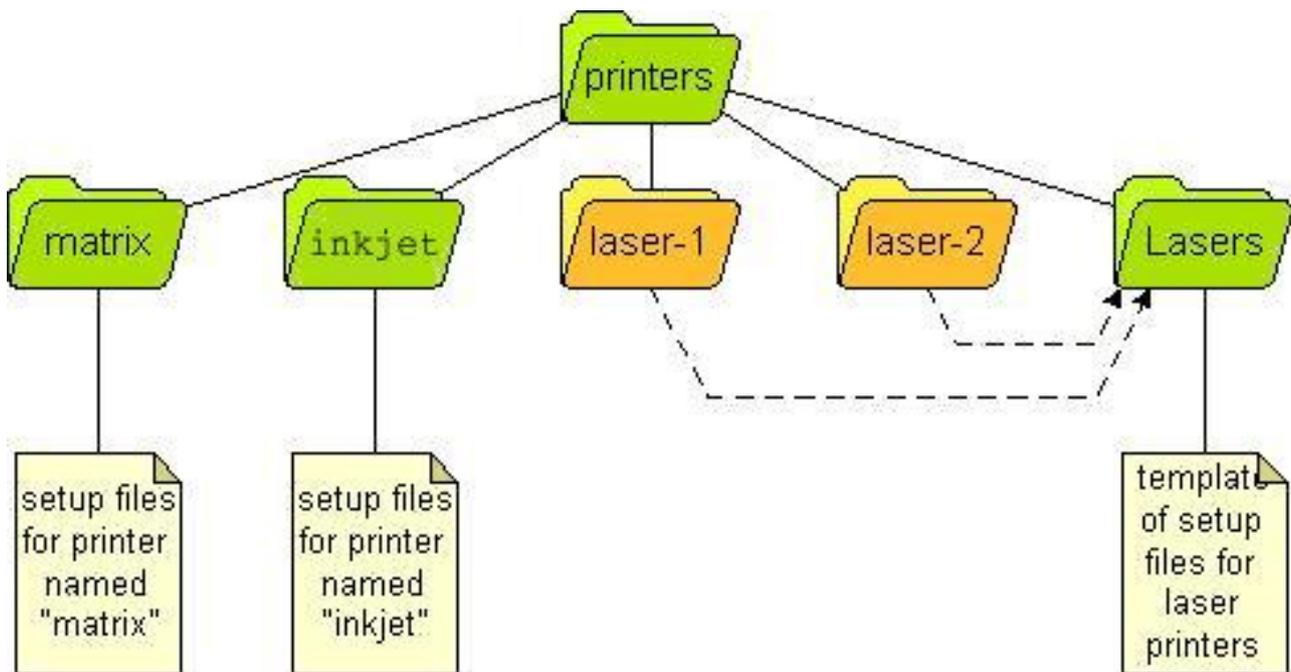
Take care not to assign values to these environment variables arbitrarily; very strange things will happen if one part of **Xi-Text** is using one set of directories and some other part is using another!

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

2.1.2 Printer Definitions and Setup Files

The printers directory contains the print control and formatting information for all of the printers that are logically attached to **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. (We refer to the symbolic links as “clones” in the printer setup script).



Both “hard” and symbolic links can also be used inside printer directories where just certain components are identical across two or more printers. This probably is not a good idea, however.

Setting up printers and their definitions is described in chapter 4 of this manual.

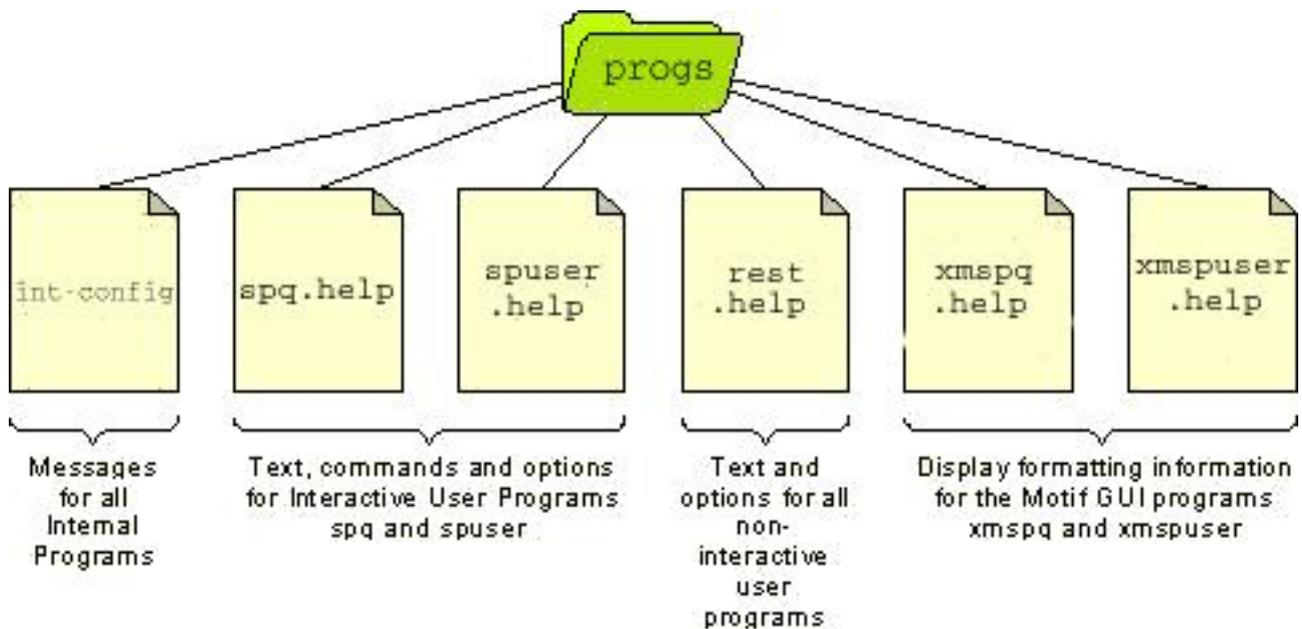
2.1.3 Help and message files

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

The files are:

File	Function
<code>spq.help</code>	Message file used by <code>spq</code> .
<code>xmispq.help</code>	Message file used by <code>xmispq</code> and <code>xspq</code> .
<code>spuser.help</code>	Message file used by <code>spuser</code> .
<code>xmispuser.help</code>	Message file used by <code>xmispuser</code> and <code>xspuser</code> .
<code>splpd.help</code>	Message file used by LPD interface programs
<code>rest.help</code>	Message file used by all other user programs including the web browser interface.
<code>int-config</code>	Message file used by internal programs.

By default the files are held in the directory `/usr/spool/progs`, sharing space with the internal programs, however a different location may be chosen if desired.



2.1.4 Spool directory files

The following files are held in the spool directory, `/usr/spool/spdby` default.

File	Function
<code>pwdump23</code>	Optional saved password map file (now deprecated)
<code>spufile23</code>	File of user permissions (with suffix of the major release of Xi-Text).
<code>spcharges23</code>	File of user charges (now deprecated).
<code>sputmp</code>	Temporary file used when reformatting <code>spufile23</code>
<code>spshed_reps</code>	Log file recording system startup, 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>SPDENnnnnnnn</code>	Used to hold the standard error output of filter commands.

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

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

2.1.5 Internal programs

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

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

All of the internal programs are described in Chapter 5 of this manual.

2.1.6 User Programs

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

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

```
-rwsr-xr-x 1 spooler bin 167619 Mar 23 2000 spq
```

The user programs are described in detail in Chapter 6 User Programs.

2.1.7 Configuration files held in /etc

2.1.7.1 Xi-Text Hosts 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.

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

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

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

An alias *must* be supplied if the host name is given as an internet address.

3. Flags, which are further described below.
4. A numeric time-out value in seconds. The default if this is omitted is 1000. This is most important for Windows clients, as it also denotes a time after which the connection becomes “stale” and must be refreshed, possibly by re-entering the password.

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

<code>probe</code>	Denotes that the scheduler should check that the specified host is active before attempting a connection.
<code>manual</code>	denotes that no connection is attempted until the operator invokes one with <code>spconn</code> .
<code>external</code>	denotes that the named host is some external system, possibly running LPD protocol, for the benefit, in particular, of incoming requests from <code>xtlpd</code> using the <code>-o</code> (originating host) option to <code>spr</code> .
<code>dos (user)</code>	for a Microsoft Windows client PC. Requests are allocated by default to the username given.
<code>client (user)</code>	is a synonym for <code>dos (user)</code> .
<code>clientuser</code>	Denotes that the first and second fields are <i>user</i> , not machine names, for DHCP clients. As a special case, if the first field is default and the second field is a user name on the Unix host, then a default user name is thereby supplied for all unknown Windows users.
<code>clientuser (machine)</code>	As <code>clientuser</code> , but denotes that the default client machine is given, otherwise a password is required.
<code>trusted</code>	Exchange information with this Unix host about Windows client users. (This is now deprecated.)
<code>pwchk</code>	Demand Unix password from Windows clients in all cases.

```

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

```

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

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

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

```
spconn green
```

or

```
spconn mach20
```

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

Next the Windows user name of `john` on any Windows PC is translated to a Unix user name of `jmc`, after checking the password.

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

The utility program `hostedit` (or the GTK+ version `xhostedit`) may be used to create or edit this file with appropriate checks.

Note that the mapping of UNIX names to Windows names in this file is deprecated – this is now done in the user mapping file.

2.1.7.1.1 Multiple IP addresses

It is sometimes unclear what the local address is, i.e. the IP address corresponding to the host on which it is running. It is important for the software to know this, as other hosts will use this to identify jobs and printers belonging to the host. It is possible to specify this in the hosts file thus:

```
localaddress 193.112.238.250
```

The `localaddress` statement must be the first item (other than comments or blank lines) in the host file. The address given can be either a host name or an IP address.

The address can also be obtained each time it is started by connecting to another host and running `getsockname()` on the result. To signify this, the following format is used.

```
Localaddress GSN(www.google.com,80)
```

The integer gives a port number to use.

The host name can be given as above, or an IP address can be used.

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

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

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

For example:

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

Please note that the text to the right of the = sign¹ is taken literally; there is no recursive expansion of `$name` constructs except for the message file names `SPQCONF`, `SPUSERCONF` and `SPRESTCONF`.

¹Also note that there is no colon (:) alternative as with **Xi-Batch**.

2.1.7.3 User Mapping 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.

2.2 Standard Exit Codes

It is often convenient to run `spr` or other programs such as `spstart` from other programs or shell scripts. In order to allow convenient error diagnosis, there are a standard set of exit statuses which are used by all the **Xi-Text** programs.

Exit	Description of Probable Cause
2	Bad class specification - would give zero
3	Bad command arguments e.g. unknown option
5	Cannot find spooler directory (usually <code>/usr/spool/spd</code>)
6	Scheduler not running or crashed
7	Illegal priority specified
8	Illegal form type specified
9	Illegal number of copies specified
10	Bad printer name specified
11	<code>spstart</code> – printer is being shut down
12	<code>spstart</code> - printer is in <code>a/w oper</code> state
13	Unknown job (<code>sqdel</code> or <code>sqchange</code>)
16	Insufficient privilege for requested operation
40	(Internal to <code>spdinit</code>) bad setup file
100	Bad or non-existent configuration file
101	I/O error on terminal input
150	(Internal to <code>spjobdump</code>) file not found.
151	(Internal to <code>spjobdump</code>) cannot change directory.
152	(Internal to <code>spjobdump</code>) cannot create output file.
153	(Internal to <code>spjobdump</code>) job not found.
200	Unexpected signal received
201	Cannot create pipe
202	Cannot fork
203	Job queue non-existent (internal error)
204	Print queue non-existent (internal error)
230	<code>spr</code> - cannot create spool file - is disk full?
240	Scheduler internal error
250	Xi-Text badly installed; setup error
254	Run out of memory
255	Missing message file

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

Error numbers less than 100 are for “normal” errors, such as invalid options or privilege violations. Error numbers >100 are “internal” or system generated errors, i.e. they are outside the control of **Xi-Text**.

In the cases of `spstart` (and its links `sphalt`, `spok`, `spnok`, `spinter`, `spadd` and `spdel`), `sqdel` and `sqchange` exit codes less than 100 will not cause the program to halt; the program will do as much as it can, reporting

appropriate error messages and terminating with an exit code corresponding to the last error message reported.

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

Chapter 3

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 who has a Unix account will also have a **Xi-Text** account.

User permissions are now held as a default set together with differences for specific users, so no special action need be taken when users are added or deleted, unless they are distinguished in some way.

There are 3 (formerly 4) aspects to the **Xi-Text** user account:

Privileges control access to usage and administration functions of the system. For example, the privilege to delete jobs from the queue.

Classcodes effectively subdivide the print resources between separate groups of users.

Priorities position print jobs in the queue relative to the other jobs. The higher the priority the closer to the front of the queue a job is placed. Facilities exist to specify what priorities each **Xi-Text** user may put on their print jobs.

Xi-Text no longer maintains a table of charges against each user.

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

3.1 Privileges

The following aspects of **Xi-Text** are subject to control by privileges.

Privilege Name	Meaning
Edit admin file.	Assign other users' privileges
Masquerade	"Pretend to be" another user for certain operations
Override class.	Increase class code (set bits on)
Stop spooler.	Run sstop
Use other forms.	Select other than pre-defined form type
Use other printers	Select other than pre-defined printer(s). On by default.
Change priority on queue.	May change priority of job on queue
Edit other users' jobs.	May inspect or change other users' jobs
Select printer list.	May move to printer list in <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 queue.	May set any priority not just in range
Change own default prio/form.	May run the change own defaults option in <code>spuser</code>
Unqueue jobs.	May take jobs off the queue, copying to a file for editing
View other users' jobs	May inspect but not change other users' jobs
Edit remote jobs	May edit jobs on remote machines
Edit remote printers	May change printers on remote machines
Access attributes	May access attributes of jobs not displayed by default on <code>spq</code> . On by default.
Freeze attributes	May save changes to default display. On by default.

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

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

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

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

When installed, the users `spooler` and `root` (the super-user) have all the privileges and other users none. The privileges and defaults should be set up for each user and the defaults for new users, using `spuser`, `xspuser` or `xmspuser`. Current privileges may be displayed by typing:

```
spuser
```

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

```
Default form standard types standard
Maximum Copies 10
Priority: Min 10 Max 100 Def 50 Current charge is 0
Default printer '' types ''
Class Code A-Pa-p
Privileges:
You may Edit admin file.
You may Masquerade as other users.
You may Override class.
You may Stop spooler.
You may Use other forms.
You may Use other printers.
You may Change priority on Q.
You may Edit other users' jobs.
You may Select printer list.
You may Halt/restart printers.
You may Add/delete printers.
You may Set any priority on Q.
You may Change own default prio/form.
You may Unqueue jobs.
You may view other users' jobs.
You may Edit remote jobs.
You may Edit remote printers.
You may access queue options.
You may save new default options.
```

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

The charge is the total charge so far accumulated by this user; this is now deprecated and always reports zero.

When a new user is added, they will get an initial set of permissions given by the default values unless these are changed.

These initial values may themselves be altered by the administrator to apply to users not marked as exceptional. When **Xi-Text** is first set up, these initial values are set as follows:

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

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

The standard default set of privileges is set as follows:

- Set other forms from default.
- Set other printers from default.
- Access printer list.
- Stop and start printers.
- Access other attributes.
- Allow access to remote jobs.
- Allow access to remote printers.
- Freeze options.

3.2 Priorities

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

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

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

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

Combination	Meaning
$min \leq default \leq max$	Normal usage; User may set priority anywhere between <i>min</i> and <i>max</i> .
$min = default = max$	Special case; User can only use <i>default</i> priority.
$min > max$	"Impossible"; User may not use Xi-Text , <code>spr</code> etc will complain.
$default > max$ or $default < min$	Default priority is effectively undefined; user must explicitly give a priority to <code>spr</code> etc, or the job will be rejected with an error message.

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

3.3 Charging

Charging is deprecated and has been removed from **Xi-Text**.

3.4 Class codes

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

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

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

Code Printer “Group”

A	General Printers
B	Accounts dept printers
C	Laser printer 1
D	Laser printer 2
E	Plotter 1
F	Plotter 2

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

3.4.1 Treatment of class codes

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

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

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

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

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

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

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

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

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

3.4.2 User access to class codes.

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

Continuing from the earlier example, if you were only interested in using laser printer 1, which has a class code of `C`, then you could specify the argument `-C C` to various programs such as `spq` and `spr`. This would guarantee that jobs only went to printers having class `C`, and when using `spq`, only jobs for that class are visible. (You can save typing by creating configuration files or environment variables to hold these options; see chapter 10 for more details).

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

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

Chapter 4

Printer Definitions

This chapter explains in detail the format of the printer setup files used to define how **Xi-Text** accesses printers.

First we need to explain the concepts of *form types*.

4.1 Definitions

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

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

The structure of the printer definitions is described below.

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

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

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

- Baud rate and other line parameters.
- Network commands for terminal server type devices.
- Flags to turn off banners and specify manual feeding (single-job mode).
- Strings to be output or commands to be executed in various circumstances.
- Alignment files, for test output onto pre-printed stationery before printing.

- “Filter” programs to post-process output before printing or to handle non-standard printer protocols.
- Optional per-printer accounting, with a separate named accounting file for a particular printer type.
- Control of banner (header) pages.
- Control of all these according to the form suffix.

4.2 Form Types

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

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

1. A paper type and
2. How to print on it

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

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

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

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

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

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

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

In addition various post-processing “filters” may be selected to perform translations and conversions as the job is printed, for example converting PCL to PostScript as the document is printed.

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

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

It is entirely up to each installation how these are defined. However we recommend that the same form type, including sets of suffixes, are interpreted on each printer to give similar effects, so for example if the form type is `a4.ps` or `letter.ps` where `ps` stands for PostScript, the handling should be as similar as possible.

Just to recap:

- The *paper type*, the `a4` in `a4.ps`, or the `letterhead` in `letterhead.p12` cannot change automatically, the printer has to be stopped, the form changed and then the printer restarted.
- The *suffix*, `ps` or `p12` in the above examples, can be changed automatically without intervention and appropriate commands sent to the printer.

We call the files which control how the printer operates, “Setup Files”.

4.3 Location and naming of setup files

The setup files for a printer are located in a directory with the same name as the printer type under the printers directory, which by default is `/usr/spool/printers`.

So for example if you had decided to call your printer `ptr1`, then you would have the setup file or files for the printer in the directory

```
/usr/spool/printers/ptr1
```

Within this directory, **Xi-Text** will look first for a file of the same name as the *paper type*, i.e. the form type without the suffix, and if it cannot find that, the file `default`.

It will use the file it finds as a setup file to denote how to print with the given form type on the printer.

So if the directory `/usr/spool/printers/ptr1` contains the files

```
a4
default
letter
```

then jobs for `ptr1` will cause the following files to be read in the following circumstances

Form	File read	Suffix
<code>a4.ps</code>	<code>/usr/spool/printers/ptr1/a4</code>	<code>ps</code>
<code>letter.p10</code>	<code>/usr/spool/printers/ptr1/letter</code>	<code>p10</code>
<code>legal</code>	<code>/usr/spool/printers/ptr1/default</code>	(none)

In nearly all cases only a `default` file is ever needed and created.

The `Ptrinstall()` script only looks at and creates this.

If neither a file with the same name as the paper type, or a `default` file can be found, this is an error and the printer will not start.

4.3.1 Changing the name of the default file

Do not use this if you are not sure what you are doing!!!

As we have said, the name `default` is special. If **Xi-Text** cannot find a setup file with the same name as the paper type, it will use the file `default` instead and if it cannot find this, it is an error.

If you want the name of the file used in place of the setup file if it cannot find one as the same name as the printer to be something other than `default`, you should specify the alternate name by assigning the variable `DEFAULT_FORM` in the master configuration file `/etc/Xitext-config`, for example:

```
DEFAULT_FORM=noformspecified
```

This would make the printer (and for that matter all printers) look for a file called `./noformspecified` rather than `../default` if they could not find a file with the same name as the paper type.

This is almost certainly a bad idea, but you might instead want to change the handling so `default` is not searched for and you want to insist on a setup file name exactly matching the paper type in every case.

You do that by specifying a null string, by putting

```
DEFAULT_FORM=
```

in the maseter configuration file `/etc/Xitext-config`.

Please don't use this facility other than very sparingly if at all. We may not realise very fast that you've done this when you call us with a problem!

4.3.2 The formname "standard"

This section is included as some users have been confused!

When **Xi-Text** is first installed, the default form type for each user is set to `standard`.

There is nothing special about this name. It is intended only as a placeholder for sites to insert their own standard paper type name.

If you keep `standard` as the default form type for each user, install printers with this form type and create a setup file for each printer called `default`, everything will work fine. Jobs will be created with form type `standard` and will be printed as intended. We just think it might be a slight waste, that is all.

The printer installation script `Ptrinstall` offers to set the default form type for users to something else if it sees `standard` as the default form type.

4.4 Printer help files

You can put an extra file `-Help` in the subdirectory for printer files corresponding to the printer, i.e. the file

```
/usr/spool/printers/printername/-Help
```

This should be a short text file summarising the options of paper types and suffixes you provide.

This will be displayed as online help to anyone asking for help whilst displaying the form types to load onto a printer, otherwise a list of paper types will be displayed based upon the file names in the corresponding directory.

4.5 Naming of files in the printer subdirectory

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

4.6 The .device file

To specify common material to be included in all setup files, the file `.device` may be specified in the printers subdirectory thus.

```
/usr/spool/printers/printername/.device
```

This file will always be read prior to the setup file (with the same name as the paper type, or `default`).

Although this is not enforced, it is most strongly recommended that this file only be used to set parameters and options relating to the hardware interface and protocol for talking to the printer, which presumably does not change with the form type.

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

The printer install script `Ptinstall` places, and expects to find, all device parameters in this file and information about the actual data sent to the printer in the setup file (the one that corresponds to the paper type).

4.7 Reading of setup files

The setup file (and the `.device` file) is read whenever

- The printer is started
- The form type suffix changes from one job to the next.

However we strongly recommend that the printer be stopped before and restarted after the files are changed. The printer does not have to be de-installed and re-installed (although `Ptinstall` does this for other reasons).

Syntax and other errors in the setup file are reported by appending lines to the file `spshed_reps`, in the spooler directory, by default `/usr/spool/spd`.

This file should always be examined if a printer appears to be misbehaving (the state of the printer will in such cases appear on the `spq` display as `error`).

All the interactive queue view utilities have an option to display the error log file. In the case of `spq`, this is the `X` key.

4.8 What setup files provide

The following are what setup files can provide:

Hardware and network parameters belong in the `.device` file, and provide options and timeout parameters and similar for talking to the device.

Other parameters such as for turning on or off header pages and controlling how jobs are sent to the device once connected.

Strings which specify strings to be output, or commands to be run, at various stages of the print cycle.

Filter command to postprocess all data.

Conditional constructs which select different sets of the above options according to the form type suffix.

The following sections explain how this is achieved.

4.9 Format of setup files

Setup files consist of text statements and strings, looking like this:

```
# Comments like this and blank lines are ignored
# But do note that Ptrinstall puts comments about the device type and
# selected options in - change these to "Custom" before you change anything.

# Here is an example of something to set a device parameter
# This sets the timeout on "open" to 60 seconds

open 60

# Here is a command to send output data via a program to output to the device
network=/usr/spool/progs/xtelnet -p 9100 -h $SPOOLDEV

# Here is a keyword to set an option - turn off all header pages

nohdr

# Here is a command saying that when the printer is halted, a string
# Escape-E should be sent

halt='\eE'

# Here is where we give symbolic names to some character strings
# Note no quotes - the string runs to the end of the line including spaces

PORTRAIT=\e&100
```

```

LANDSCAPE=\e&l10

# Here is where we select different actions depending on the suffix.
# If the suffix starts with "l" we select landscape otherwise portrait

{
    (l*)
        sufstart LANDSCAPE
    (*)
        sufstart PORTRAIT
}

```

Notice that in the conditional construct introduced by `{` and ended by `}`, the first condition that matches suffix applies, so the second and subsequent conditions will be ignored if the suffix starts with `l`.

Also notice that there is no `=` between `sufstart` and `LANDSCAPE`. The string (given in this case as a symbolic name) is concatenated here to the end of whatever previous parts of the setup file have put into the string `sufstart`. If you had wanted to start from scratch and delete whatever else was in `sufstart` before, you would have put the `=` in, thus

```
sufstart=LANDSCAPE
```

4.10 Keywords in setup files

In the following subsections we list the keywords which control the action of the setup file.

Any other keywords are available for user definition as giving symbolic names to strings. Predefined keywords are all lower case, so it is normally conventional to make symbolic names all upper case as with `LANDSCAPE` and `PORTRAIT` above.

4.10.1 Keywords specific to devices

The following keywords specify actions or settings relevant to devices in general.

All of them probably should only be used in the `.device` file.

Some kinds of devices have commands relevant to them which are tabulated later

Keyword	Function	See Page
<code>canhang</code>	Possible hangup	50
<code>close</code>	Close timeout	50
<code>offline</code>	Offline timeout	49
<code>open</code>	Open timeout	49
<code>outbuffer</code>	Output buffer size	49
<code>portsetup</code>	Pre-open action	51
<code>postclose</code>	Sleep after close	50
<code>reopen</code>	Close/reopen device	51
<code>stty</code>	Post-open command	64

4.10.1.1 outbuffer keyword

The keyword command

```
outbuffer n
```

where *n* is an integer, sets the buffer size for blocks of output by **Xi-Text** to the given size in bytes, which can be anything from 1 upwards to 4GB.

The default if this is not specified is 1024.

4.10.1.2 open keyword

The keyword command

```
open n
```

where *n* is an integer, sets the timeout for attempts to open the device or network connection to the given number of seconds. This may be between 1 and 32767.

The default if this is not specified is 30 seconds.

subsubsectionoffline keyword

The keyword command

```
offline n
```

where *n* is an integer, sets the timeout for attempts to send data to the device or network connection to the given number of seconds. This may be between 1 and 32767.

The default if this is not specified is 30 seconds.

If the device does not respond within the timeout, it is halted and set to *offline* state.

4.10.1.3 close keyword

The keyword command

```
close n
```

where *n* is an integer, sets the timeout for attempts to close the device or network connection to the given number of seconds. This may be between 1 and 32767.

The default if this is not specified is 300 seconds (5 minutes).

The reason for having this parameter is that some devices refuse to allow the device to be closed until they have finished printing or some error condition has cleared. This sets a time limit after which an error is assumed to have occurred.

If you need to worry about this, you will almost certainly want to invoke the `canhang` keyword described on page 50.

4.10.1.4 postclose keyword

The keyword command

```
postclose n
```

where *n* is an integer, sets a “sleep time” after printing a job to the given number of seconds. This may be between 0 and 32767.

The default if this is not specified is 0 seconds (i.e. no sleep).

The reason for having this parameter is that **Xi-Text** runs too fast for some devices and if there is too little a delay before reopening the device again some may refuse to operate. This will cause the despooler process `spd` to pause for this time before reporting back that the job has been completed.

Most of the drivers supplied with **Xi-Text** handle this for themselves, so this should normally not be needed.

4.10.1.5 canhang keyword

The keyword command

```
canhang
```

Warns **Xi-Text** that the device or network command is “hard to kill”.

On some systems (often legacy UNIX systems) a device will cause the last process trying to close a connection to it “hang” immortally until whatever caused the hanging connection is resolved.

If this is one of the **Xi-Text** processes, in particular `spd`, the “immortal” but non-functioning process can cause the whole of **Xi-Text** to lock up.

By specifying this, `spd` will fork off a copy of itself to be the last process to actually close the connection after closing everything else, leaving the main path able to exit.

4.10.1.6 reopen keyword

The keyword command

```
reopen
```

Causes **Xi-Text** to close the device or network connection at the end of each job and restart it at the beginning of the next job.

This is almost always required for network connections, which frequently do not start printing a job until the connection has been closed.

It is often necessary for parallel or USB devices which may not print the last part of a job until the connection is closed.

4.10.1.7 portsetup keyword

The keyword command

```
portsetup=command
```

Specifies a shell command which should be executed immediately prior to any attempt is made to open the device or network connection.

This is sometimes necessary where the printer has to have firmware downloaded or similar at the start of the day, or some network connection has to be enabled.

Note that the environment variables `SPOOLPTR`, `SPOOLDEV` and `SPOOLFORM` will be set in the environment of the command invoked to that of the printer name, device name and form type with which the printer was started.

The usual way of specifying this is as an assignment with a string, which is taken literally including spaces to the end of the line. However the technique of assigning successive segments usually associated with the spooler actions such as `docstart` may be used, so instead of

```
portsetup=/shell/command arg1 arg2
```

You could put

```
portsetup '/shell/command'
portsetup ' arg1'
portsetup ' arg2'
```

if you wanted to. Note however that escape sequences such as `\e` for escape etc are not interpreted in the strings, they are taken literally.

There is a minor subtlety in the function of `portsetup` if the `exec` keyword is specified, for example

```
porsetup exec '/full/command/path arg1 arg2'
```

The command will be executed directly, not via the shell and with environment variables in the command line prefixed with `$` expanded first.

If any kind of error occurs with the command, or it returns a non-zero exit code, the printer will not start and will revert to *error* state.

4.10.1.8 `stty` keyword

The keyword command

```
stty=command
```

Specifies a shell command which should be executed immediately after the device or network connection has been opened with the standard input, standard output and standard error set to the device.

Note that the environment variables `SPOOLPTR`, `SPOOLDEV` and `SPOOLFORM` will be set in the environment of the command invoked to that of the printer name, device name and form type with which the printer was started. Others such as the job title of the next job to be printed will also be set in `SPOOLHDR` but this is unlikely to be useful or interesting.

This is called `stty` for historical reasons to cope with cases which the standard `ioctl` calls for serial devices did not cover all the facilities, but it can be used to run any command at all.

It should be contrasted with `portsetup`, which is run *before* the device or network connection is opened.

The usual way of specifying this is as an assignment with a string, which is taken literally including spaces to the end of the line. However the technique of assigning successive segments usually associated with the spooler actions such as `docstart` may be used, so instead of

```
stty=/shell/command arg1 arg2
```

You could put

```
stty '/shell/command'
stty ' arg1'
stty ' arg2'
```

if you wanted to. Note however that escape sequences such as `\e` for escape etc are not interpreted in the strings, they are taken literally.

There is a minor subtlety in the function of `stty` if the `exec` keyword is specified, for example

```
stty exec '/full/command/path arg1 arg2'
```

The command will be executed directly, not via the shell and with environment variables in the command line prefixed with `$` expanded first.

If any kind of error occurs with the command, or it returns a non-zero exit code, the device or network connection will be closed and the printer will revert to *error* state.

4.10.2 Keywords specific to serial devices

The following keywords are specific to serial devices. Not all of them are available on both System V-based and BSD-based serial devices.

Almost all are very much “legacy” operations as we have to support all kinds of ancient devices and are ignored with other kinds of device.

If you do use any of them, be sure to avoid the ones that transform the output, such as expanding tabs and inserting CR before LF characters as these will cease to work if the hardware interface changes. Also, they are a bad idea if you ever need to send binary graphics to the printer, as they will be corrupted.

Keyword	Function
<code>baud n</code>	Baud rate
<code>cbreak</code>	CBREAK output mode
<code>clocal</code>	Ignore modem control
<code>crmod</code>	Add/remove CR
<code>cs5</code>	Character size 5
<code>cs6</code>	Character size 6
<code>cs7</code>	Character size 7
<code>cs8</code>	Character size 8
<code>even</code>	Even parity
<code>extabs</code>	Expand tabs
<code>ibaud</code>	Input baud
<code>ixany</code>	Any char restarts
<code>ixon</code>	X-ON/X-OFF flow control
<code>lcase</code>	Map to upper case
<code>obaud</code>	Output baud
<code>ocrnl</code>	Map CR to LF
<code>odd</code>	Odd parity
<code>olcuc</code>	Lower to upper case
<code>onlret</code>	Inserts CR
<code>onlcr</code>	Add CR to LF
<code>onocr</code>	No CR
<code>parodd</code>	Odd parity
<code>parenb</code>	Parity generation
<code>raw</code>	Raw mode
<code>tabs</code>	Same as extabs
<code>tandem</code>	Tandem flow control
<code>twostop</code>	Two stop bits

4.10.3 Keywords specific to network devices

The following keywords relate only to network-style devices.

Network commands are requested by using the `network` keyword. This specifies the subcommand process which receives the spooler data on standard input and sends it on to the device.

Strictly speaking, it doesn't have to be a network device, it could be some other kind of device which needs a subprocess to be run to send it on.

Keyword	Function	See Page
<code>closekill</code>	Kill signal	56
<code>exit</code>	Exit handling	56
<code>fberror</code>	Feed back error	55
<code>logerror</code>	Log errors	55
<code>network</code>	Network filter	54
<code>signal</code>	Signal handling	56

4.10.3.1 Network keyword

The keyword command

```
network=command
```

Specifies a subprocess to be used to pass the data onto the device, usually a network connection.

This should take data to be printed on standard input, and send it on, using the appropriate protocol.

Any error messages should be display on standard error, which will be saved and passed on according to the settings given by `fberror` and `logerror`.

Note that the environment variables `SPOOLPTR`, `SPOOLDEV` and `SPOOLF` will be set in the environment of the command invoked to that of the printer name, device name and form type with which the printer was started, as will the other environment variables such as `SPOOLHDR`. Nothing, apart from the possible display of job titles, should be dependent on these variables.

The usual way of specifying this is as an assignment with a string, which is taken literally including spaces to the end of the line. However the technique of assigning successive segments usually associated with the spooler actions such as `docstart` may be used, so instead of

```
network=/shell/command arg1 arg2
```

You could put

```
network '/shell/command'
network ' arg1'
network ' arg2'
```

if you wanted to. Note however that escape sequences such as `\e` for escape etc are not interpreted in the strings, they are taken literally.

Note that most installation commands such as `Ptrinstall` only use the `network=` form.

There is a minor subtlety in the function of `network` if the `exec` keyword is specified, for example

```
network exec '/full/command/path arg1 arg2'
```

The command will be executed directly, not via the shell and with environment variables in the command line prefixed with `$` expanded first.

If any kind of error occurs with the command, or it returns a non-zero exit code, the device or network connection will be closed and the printer will revert to `error` state, although this can be fine-tuned using the `exit` keyword.

4.10.3.2 `fberror` keyword

The keyword command

```
fberror
```

Arranges for the last line of any output on standard error of the network subprocess to be displayed against the indication `error` on the `spq` printer list display or similar for `xspq`, `xmispq` and `splist`.

This could mean that with appropriate catching of error indications from the printer, the display of a printer halted with an error message with `splist` could be something like:

```
ptr1 <p1> a4.ps error:Paper Jam
ptr2 <p2> a4.ps offline:Out of paper
```

See page 305 to see how this could be extended further to send alarm messages regarding the unexpected halting of printers.

The `Ptrinstall` script sets this on by default, there is no particular reason not to use it unless you have a terminal server driver process which produces unhelpful error messages.

4.10.3.3 `logerror` keyword

The keyword command

```
logerror
```

Arranges for standard error from the network subprocess to be copied to the system log file `spshed_reps` in the spool directory, by default in `/usr/spool/spd`.

There is no particular reason not to do this unless you have a terminal server driver process which produces endless streams of unhelpful error messages.

The `Ptrinstall` script sets this on by default.

4.10.3.4 closekill keyword

The keyword command

```
closekill n
```

Sets the signal which will be sent to the network subprocess if the printer is halted in the course of printing.

This can be any valid signal number, from 1 up to the maximum.

The default if nothing is specified is 2 or `SIGINT`.

4.10.3.5 exit keyword

The keyword commands (for example)

```
exit setoffline 1,10-50
exit seterror 2-4,51-255
```

Indicates that if the network subprocess exits with an exit code in the ranges given, then the printer will be halted and set into state `offline` if the `setoffline` is given or `error` if the `seterror` is given.

The exit codes specified may be between 0 and 255.

If the exit code does is not specified in either variation, then the process is considered to have normally terminated.

If the exit code appears in both ranges, it is not really defined which it will be treated as, probably as being offline, but this should not be relied upon.

You might want to combine this with the `fberror` keyword to give a helpful display of what went wrong during printing.

The default if this is not specified is to take exit code 0 as “normal” and 1-255 as error.

4.10.3.6 signal keyword

The keyword commands (for example)

```
signal setoffline 1,15-17
signal seterror 2,3,9-11
```

Indicates that if the network subprocess exits with a signal in the ranges given, then the printer will be halted and set into state `offline` if the `setoffline` is given or `error` if the `seterror` is given.

The signal numbers specified may be between 1 and the maximum on the system.

If the signal number is not specified in either variation, then the process is considered to have terminated with an error.

If the signal number appears in both ranges, it is not really defined which it will be treated as, probably as being offline, but this should not be relied upon.

The default if this is not specified is to take signal number 1 (`SIGHUP`) as offline and anything else as an error.

4.10.4 Keywords selecting spooling options

The keywords below relate entirely to how the spooler operates once connection with the printer has been established.

They should not appear in the `.device` file, but in the file corresponding to the paper type or `default`, possibly conditional on various suffixes.

They are listed here with a reference to the type of action which they relate, which are explained separately.

Keyword	Function	Relates to	See page
<code>abort</code>	Abort action	Setup strings	59
<code>addcr</code>	Add CR before LF	Miscellaneous	69
<code>align</code>	Set alignment page	Alignment pages	67
<code>banner</code>	Banner options	Header pages	66
<code>bannprog</code>	Banner program	Header pages	65
<code>charge</code>	Charge rate	Miscellaneous	70
<code>delimiter</code>	Page Delimiter	Page breaks	68
<code>docend</code>	Document end action	Setup strings	59
<code>docstart</code>	Document start action	Setup strings	59
<code>exec</code>	Execute string	Setup strings	59
<code>execalign</code>	Alignment program	Alignment pages	67
<code>filter</code>	Output filter	Output filtering	63
<code>forcehdr</code>	Force header on	Header pages	64
<code>halt</code>	Device halt action	Setup strings	59
<code>hdrpercopy</code>	Header on each copy	Header pages	64
<code>inclpage1</code>	Print page 1 always	Page breaks	69
<code>logfile</code>	Log file	Miscellaneous	71
<code>nohdr</code>	Turn off headers	Header pages	64
<code>noranges</code>	Turn off pages	Page breaks	68
<code>onecopy</code>	One copy only	Miscellaneous	70
<code>pageend</code>	Page end action	Setup strings	59
<code>pagestart</code>	Page start action	Setup strings	59
<code>record</code>	Binary record	Miscellaneous	70
<code>restart</code>	Restart action	Setup strings	59
<code>retain</code>	Retain after printing	Miscellaneous	70
<code>setup</code>	Startup action	Setup strings	59
<code>single</code>	Single job mode	Miscellaneous	69
<code>stdhdr</code>	Standard header	Header pages	64
<code>sufend</code>	Suffix end action	Setup strings	59
<code>sufstart</code>	Suffix start action	Setup strings	59
<code>width</code>	Set page width	Header pages	66
<code>windback</code>	Wind back pages	Page breaks	68

4.11 Setup strings

The way in which **Xi-Text** provides a flexible and easy-to-control handling system for programmatically setting the printer in various modes is by the system of “setup strings”, or alternatively post-processing filters, described on page 63.

By conditionally setting them according to the value of the suffix, it is possible to insert commands to the printer to select programmable options such as double-sided printing, input or output trays, fonts and sizes or other facilities.

There are 10 setup strings, which can be “hung” onto various stages of the print cycle.

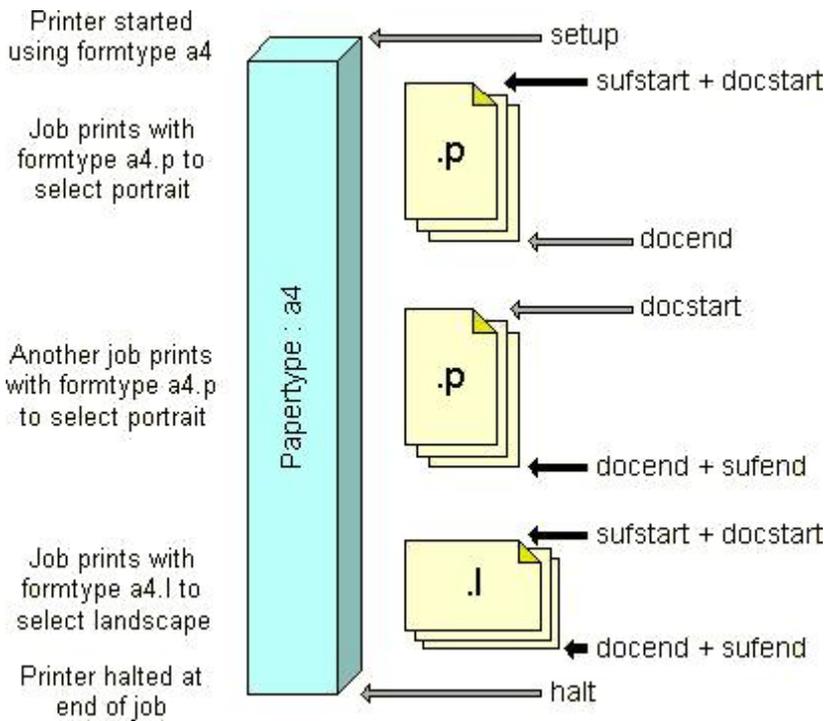
These are as follows:

String	Function
<code>setup</code>	Action when printer first started
<code>halt</code>	Action when printer is halted
<code>sufstart</code>	Action when suffix is selected
<code>sufend</code>	Action when suffix is deselected
<code>docstart</code>	Action at the start of each document
<code>docend</code>	Action at the end of each document
<code>pagestart</code>	Action at the start of each page
<code>pageend</code>	Action at the end of each page
<code>abort</code>	Action if printer is halted whilst printing
<code>restart</code>	Action if job is restarted at the beginning

Probably `setup` and `halt` will not be dependent on any suffix, but all the others often are.

Please note that the action is **either** to output to the printer **or** to run as a shell command with output to the printer **but not both**.

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

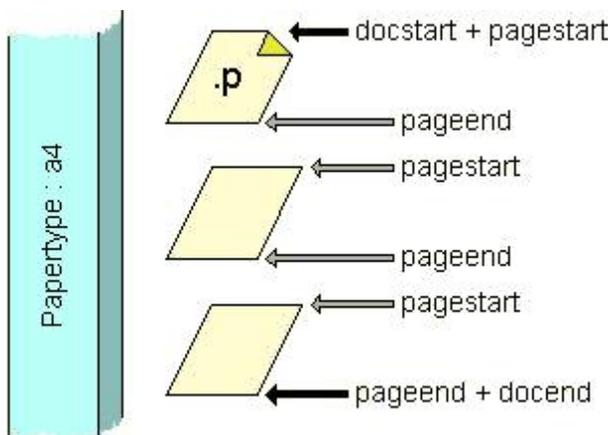


The diagram shows the sequence of commands, excluding `pagestart` and `pageend`, that are sent to the printer. The dark arrows show where more than one command is sent consecutively. The print jobs have the same papertype with different suffices, `a4.p` for portrait and `a4.l` for landscape.

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

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

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



There are two other setup strings which do not apply in the normal cycle of printing, the `abort` string and the `restart` string. These strings or commands are invoked if printing is halted prematurely or printing is restarted from the beginning.

4.11.1 How to define setup strings

There are two basic formats of definition of setup string (illustrated here with `setup`, but applying equally to the other versions).

```
setup='\eE'
setup '\eE'
```

The first format assigns the setup string, clearing what was there before.

You might want to do this if there are a variety of paths through the setup file according to the suffix and you want to be sure of “starting from scratch”.

The second format accumulates the new string onto the end of the selected string.

In fact one of the string selection keywords sets the “currently selected string” so that any other strings will be appended to that string, so you could put

```
setup 'part1' 'part2'
      'part3'
      'part4'
```

However you should not write the setup files so as to rely on this by assuming one string is selected for dozens of lines in the file!

4.11.2 Executing strings

Choosing whether strings are to be sent, or executed as a command, is selected by the `exec` keyword.

The `exec` keyword indicates that the whole of the string in question is to be used as a command with standard output sent to the device thus:

```
setup exec '/usr/local/bin/mysetup'
```

This command will be executed by the shell with environment variables as given on page 71.

If you need to reset this indication, perhaps because a different suffix calls for different treatment, you put a minus sign in front, thus:

```
docstart -exec
docstart=''
```

The first line turns off the `exec`, the second resets `docstart` to be the empty string.

If you set `exec`, the handling of escape characters, such as in `\e` for escape is disabled, the characters in the string are taken literally henceforth, **but not** the characters already in the string, so be sure to reset it by assigning the empty string if in doubt.

4.11.3 Format of strings

Strings are delimited by single quotes, as in

```
'single-quoted string'
```

double-quoted strings as in

```
"double-quoted string"
```

or matching angle brackets as in

```
<angle-bracket-quoted string>
```

If you need to include one of the quote styles in a string, use one of the other styles, remembering that strings are concatenated thus:

```
docstart 'String with " in' "'"
```

or escape with `\` as with

```
docstart 'String with \' in'
```

The following escape sequences are used to produce non-printing characters. As noted above, they are ignored when the `exec` keyword is in effect.

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

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

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

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

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

4.11.4 Symbolic names for strings

Rather than typing in the same escape sequence each time, it is often easier to give a symbolic name to the string once and use the sybolic name in place of a quoted string, for example

```
RESET=\eE
.....
setup=RESET
halt=RESET
```

Note that there are no quotes, the string is taken literally up to the end of the line including spaces, but escaped characters such as `\e` continue to be interpreted.

We recommend you represent spaces and tabs (especially trailing ones) with `\s` and `\t` respectively to leave no doubt as to the intention.

As strings are appended to the current string, this makes a handy way of setting up a composite string, thus

```
docstart PORTRAIT PITCH12 LPI8
```

The same symbolic name can be redefined any number of times in the setup file, perhaps conditionally according to the suffix. However it is a mistake to use a symbolic name before it has been defined.

You can give a symbolic name to strings which are to be executed using the `exec` keyword, but remember that the interpretation of escape sequences introduced by `\` and `^` takes place at the definition time, so if you need them in the shell commands to be executed by `exec`, they should be doubled up.

```
COMMAND=/usr/local/bin/mycommand
ARG=String with \\ in
SPACE=\s
QUOTE='
....
docstart exec COMMAND SPACE QUOTE ARG QUOTE
```

Will set the `docstart` string to execute

```
/usr/local/bin/mycommand 'String with \ in'
```

4.11.5 Default string setting

If a string apart from `abort` or `restart` is not specified, it will be empty, except that the string `docend` will contain a single formfeed character if it is not assigned to.

If you do not need this, any assignment, even of the null string appended to it, will turn it off, thus:

```
docend ''
```

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

4.12 Filter commands

An alternative to using setup strings is to have all the output passed through a “filter” to transform the text.

We often use [Ghostscript](#) to turn PostScript files to PCL files and let us set up all the options to it.

A filter is specified by assigning to the special string `filter`.

This can be done by an assignment (which replaces any previous contents)

```
filter=/usr/bin/myfiltercommand arg1 arg2
```

(The whole of the string given to the end of the line is taken literally)

Or in segments as with setup strings

```
filter '/usr/bin/myfiltercommand '
filter 'arg1 '
filter 'arg2 '
```

Strings with symbolic names can also be used as with

```
CMD=/usr/bin/myfiltercommand
SPACE=\s
ARG1=arg1
ARG2=arg2
filter CMD SPACE ARG1 SPACE ARG2
```

Note that escape characters are recognised at the definitions, but not in a string after the `filter` command, so any `\` and `^` characters should be doubled in the definition of symbolic names.

The filter command should accept data on standard input and send its results on standard output. Any output on standard error will be mailed back to the user. It is probably best to divert this to a log file if possible.

There is a minor subtlety in the function of `filter` if the `exec` keyword is specified, for example

```
filter exec 'usr/bin/myfiltercommand arg1 arg2'
```

The command will be executed directly, not via the shell and with environment variables in the command line prefixed with `$` expanded first.

The full set of environment variables as specified on page 71 is passed in either case.

Note that if setup strings like `docstart` and so forth are also supplied, their output will go through the filter. You probably want to avoid this, or only use the command version to do something else.

If need be, if you are using filters, reset all the setup strings to the null string.

Note that the exception of a default formfeed character if no `docend` string is assigned does not apply if a filter is specified.

4.13 Header or banner pages

Sites differ about what they like with header or banner pages in front of their output.

Some like them in front of their output, some not, some let users choose. Some have their own formats. We try to cater for everything.

4.13.1 Whether header pages are printed

The default, if nothing else is set, is to print header pages unless the user puts `-s` to `spr` when the job is submitted or changes this setting on the job before it is printed.

If a header page is printed and multiple copies are being printed, then the header appears before the first of a set of multiple copies but not the remainder. It is possible, however, that **Xi-Text** may decide to stop printing all of a set of multiple copies to intersperse higher priority jobs in between. In such a case, the header is printed again before the first of the resumed set of multiple copies.

If you want to force headers to be printed whatever the user sets the options to on the job, use the keyword command

```
forcehdr
```

If you want to force headers *not* to be printed whatever option is set on the job, use the keyword

```
nohdr
```

If you have set these in the setup file and want (perhaps after looking at the suffixes) to restore the default behaviour so users can choose, put

```
stdhdr
```

If you want to have a header before every copy of a set of multiple copies, use the keyword

```
hdrpercopy
```

If you want to change this back, use a minus sign in front, thus

```
-hdrpercopy
```

4.13.2 Format of banner pages

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

```
bannprog=shell command
```

As with other such paramets you con build them up using concatenated strings

```
bannprog 'shell command'
```

You can also include the `exec` keyword to signify the program is to be executed directly, with expansion of `$` constructs, rather than via the shell, but this is unlikely to be particularly useful.

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

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

6. Priority of job as decimal digits.
7. Organisation name licensed to.
8. Serial number.

Don't forget that the program will have other parameters such as the form type and job title available in the environment variables `SPOOLFORM`, `SPOOLHDR` etc. These are further documented on page 71.

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

Note that for the purpose of keeping a log of the number of pages sent, 1 page is assumed to be generated by the banner program.

4.13.3 Tuning the standard banner

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

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

```
banner
```

and

```
-banner
```

For example:

```
banner
docstart '.....'
docend '.....'
-banner
# "Main" docstart etc
docstart '.....'
```

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

```
banner
-banner
```

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

For historical reasons, there is a second set of serial line parameters similar to the main set available for banners, but Serial printers are so rare these days, they should be forgotten. If you really want to specify them, they can go inside the `banner ... -banner` construct as with the `docstart` and `docend` strings, perhaps in the `.device` file.

A final tweak affecting the standard banner, which makes big characters out of `X` characters, is that you can specify the page width in characters with

```
width n
```

However it is best to customise the banner using the `bannprog` construct.

We provide scripts `pcbanner` and `psbanner` in the internal programs directory `/usr/spool/progs` to generate custom banners for PCL and PostScript printers respectively.

4.14 Alignment pages

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

```
align=.setpage
```

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

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

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

```
execalign=myalign
```

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

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

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

4.15 Settings relating to page breaks

Xi-Text does not attempt to analyse the data, so it is only really possible to detect the end of pages with plain text and simple page delimiters such as formfeed or counting 66 lines.

If the pages are not easy to recognise in the data, it is inappropriate to start trying to identify pages, specify ranges or send `pagestart` or `pageend` strings.

This can be signified in the setup file using the keyword

```
noranges
```

This will turn off the identification of pages and interpretation of page ranges.

The following sections apply where it is possible to identify page breaks.

4.15.1 Delimiters

The default delimiter if nothing is specified is the formfeed character. This works well for many kinds of plain text.

If you want to specify some other delimiter, you can do so when the job is queued using the `-d` and `-D` options to `spr` described on page 99.

Alternatively you can specify a delimiter so they are calculated (or possibly recalculated) when the job is printed.

You can do this with the keyword command `delimiter`, for example

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

which means pages are delimited by four asterisks, or:

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

which means that the page is delimited by counting multiples of 66 linefeed characters. The string can also be specified as a symbolic name as with setup strings, see page 62, or explicitly as characters in double or single quotes, or angle brackets `<>`.

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

4.15.2 Windback pages

If the print is aborted because the printer is stopped, **Xi-Text** will remember the last page printed as a "halted at" page separate from the starting or ending pages on the job.

This can be manually reset from `spq` or similar. but if you always want to go back a given number of pages when the print is resumed, you can signify this using the command keyword `windback`, for example

```
windback 3
```

will move back automatically 3 pages if the printer is halted in the middle of a document, or to the beginning of the document if the current page was 3 or less.

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

4.15.3 Inclpage1 keyword

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

The `inclpage1` keyword indicates that the first page should always be printed whatever the pages ranges are set to.

4.16 Miscellaneous other facilities

The following are additional options that may be useful in some situations.

4.16.1 Add Carriage Return

The keyword `addcr` causes **Xi-Text** to prefix every linefeed character with a carriage return character.

This may be the only way of getting lines to start afresh at the start of the line each time. Be wary however, as if binary data ever gets sent it will corrupt it. If need be cancel it by means of

```
-addcr
```

4.16.2 Single job mode

The keyword `single` causes the single-job mode of operation to be selected. In this mode each job causes **Xi-Text** to halt the printer in `a/w oper` mode, for the operator to restart it using the `y` command against the printer line in `spq` and equivalents in the other interactive programs, or `spok` from the command line (actually `n` and `spnok` also work).

It is intended

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

4.16.3 Retention of jobs

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

Many sites like to keep certain prints on the queue until they are satisfied that they have been printed correctly. They can always be reprinted by appropriately resetting the number of copies.

4.16.4 Onecopy

The `onecopy` keyword restricts **Xi-Text** to send only one copy to a printer, regardless of how many copies were requested.

You might want to do this in order to limit the number of copies.

Alternatively you may have a filter process or even a network filter process which performs the multiple copies itself, you may want to turn it on.

The process can discover what the number of copies originally requested were by looking at the environment variable `SPOOLCPS`. (It can anyway, this will have the number of copies remaining to be printed).

4.16.5 Charge keyword

The sequence

```
charge n
```

where `n` is some number, is ignored.

It used to be for setting the charging rate for different printers and form types, but charging has been deprecated. However the syntax is maintained so as not to break existing setup files.

4.16.6 Record

It is sometimes useful to record the total number of characters output on a printer. This can be useful for estimating lifetimes of ribbons or toner cartridges.

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

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

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

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

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

This feature is mostly deprecated, if you want to do this kind of thing, use log files instead.

4.16.7 Log Files

The keyword `logfile` creates a log of each job printed on the printer, for example

```
logfile=.usage
```

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

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

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

It is assumed that utilities outside of **Xi-Text** will process this file. (E.g. `grep` or `sed`).

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

4.17 Environment variables passed to commands

In various places commands or subprocesses may be invoked, these are:

- Network commands to send data to a hardware device.
- Filter commands.
- Any of the setup strings `setup`, `sufstart`, `docstart` and similar if they are set to be executed as a shell command.
- Banner programs.
- Alignment programs.

Each of these are passed the following environment variables, although some of them may be the null string or a meaningless value, for example the `halt` string will probably have a fossil value of the last job title.

The environment variables are as follows:

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

Do not confuse these with the environment variables set up in `/etc/Xitext-config`.

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

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

4.18 Interaction of filter and setup strings.

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

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

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

```
docend '\f'
```

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

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

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

4.19 Avoiding a terminating formfeed

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

```
docend ''
```

4.20 Conditions.

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

The syntax is as follows:

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

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

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

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

sufstart
{
  (dr)      DRAFT
  (co)      CORRESP
```

```

        record 0:totals
(inv)   CORRESP
        align=.invoices
        record 4:totals
(lq)   LQ
        nohdr
        single
        record 8:totals
}

```

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

Note that:

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

A simple “wildcard” notation, just like the Bourne shell’s, may be used:

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

It would probably be sensible to replace this example by:

```

DRAFT=\em0
CORRESP=\em1
LQ=\em2
{
  (co)  sufstart CORRESP record 4:totals
  (inv) sufstart CORRESP
        align=.invoices
        record 4:totals
  (lq)  sufstart LQ
        nohdr
        single
        record 8:totals
  (*)  DRAFT record 0:totals
}

```

To treat all undefined suffix types as draft.

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

```

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

```

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

4.21 Real setup files.

The following is a setup file used on a laser at Xi Software which had a serial interface

```

ixon nohdr
DOCSEP=\\$E*
NOJUST=\\$L0
DISPLAY=\\$Z.MESSAGE \\"$SPOOLHDR\\"
NEWL=\\\n
ECHO=echo
FONTSHELL=/usr/l-lib/laser/bin/listfonts

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

```

Notes:

1. This provides for 3 types of suffixes:
 - (a) `scw10`, to denote landscape program listing, together with font type and point size,
 - (b) `lpcw12`, for portrait program listing in the same manner, and
 - (c) a default, that includes a filter to detect and download fonts as used by the document.
2. In the program listing mode, the shell script `/usr/l-lib/laser/bin/listfonts` either downloads (with argument `start`), or clears (with argument `end`) the required font(s). `listfonts` determines the appropriate font from the form type, which is passed in the `SPOOLFORM` environment variable.

3. The laser printer requires documents to begin and end with the string `$E*`. In the case of the `docstart` string, this is written `\\$E*`, because the shell requires a `\` to be inserted before the `$` if the `$` is not to be interpreted by the shell, and the syntax of the setup files requires two backslashes `\\` to be inserted to generate a single `\` as part of a string. Similar complications occur with the definition of `DISPLAY` used in the `docstart` string.
4. The elaborate `docstart` command puts up the job title on the front panel display of the laser printer.

The next example is that of a setup file for an Epson-compatible dot matrix printer on a serial interface (we now put parameters like this in the `.device` file).

```

NLQMODE=\ex1
DRAFTMODE=\ex0
ELITE=\eM
PICA=\eP
CONDENSED=\e\x0f
CANCCOND=\x12
PROP=\ep1
CANCPROP=\ep0
ixon extabs
width 136
{
    (*lq)
        sufstart NLQMODE
        sufend DRAFTMODE
        charge 2000
}
{
    (g*)
        nohdr
    (e*)
        sufstart ELITE
        sufend PICA
        width 163
        onlcr
        sufstart ELITE CONDENSED
        sufend PICA CANCCOND
        width 272
        onlcr
    (c*)
        sufstart CONDENSED
        sufend CANCCOND
        width 233
        onlcr
    (p*)
        sufstart PROP
        sufend CANCPROP
        width 200
        onlcr
    (nr*)
        sufstart PICA
    (*)
        onlcr
}

```

```
{
  (*uk*)
  sufstart '\eR\x03'
  sufend '\eR\x00'
}
docend '\f\r'
```

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

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

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

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

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

logfile=../Record

# Set the offline timeout to 3000 seconds

offline 3000

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

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

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

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

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

PORTRAIT=\e&l00
LANDSCAPE=\e&l10

# Give symbolic name to the reset command.
```

```
RESET=\eE

# Give symbolic names to the built-in fonts

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

# Give symbolic names to the pitches available

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

# Normally insert carriage returns before every line
# feed.

onlcr

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

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

    (lh)

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

BOTTOMFEED

# if your printer behaves differently

    banner
    docstart LINE_TERM
    docend OFF_LINE_TERM '\f'
    -banner

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

```

docstart TOPFEED
docend '\r'
sufend BOTTOMFEED

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

(l[!h]*)
sufstart LINE_TERM

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

banner
-banner
docstart LANDSCAPE
docend PORTRAIT
sufend OFF_LINE_TERM

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

(p*)
sufstart LINE_TERM
sufend OFF_LINE_TERM

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

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

# Next we provide for font selection.
# We do this by looking for strings somewhere in the
# suffix, thus ltml0 will match the second option
# below and select the "BRT" font - in landscape mode
{

```

```

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

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

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

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

```

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

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

Chapter 5

Internal Programs and files

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

The internal programs are all held in the same directory, `/usr/spool/progs` by default. A different directory may be selected by assigning to the environment variable `SPROGDIR`, usually in the *Master Configuration File* `/etc/Xitext-config`.

5.1 Core Internal Programs

5.1.1 Spshed

`Spshed` is the spool scheduler process for the **Xi-Text** spooler and document management system.

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

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

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

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

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

Appropriate log messages are written by `spshed` and other system processes to a log file, `spshed_reps` in the spool directory, by default `/usr/spool/spd`. Be sure to check this file for any error messages relevant to any problems you encounter.

5.1.1.1 Files used by spshed

<code>/etc/Xitext-hosts</code>	host names and descriptions
<code>/etc/Xitext-config</code>	master configuration file
<code>/usr/spool/progs/int-config</code>	message file
<code>/usr/spool/spd</code>	spool directory
<code>/usr/spool/spd/spshed_jfile</code>	job file
<code>/usr/spool/spd/spshed_pfile</code>	printer file
<code>/usr/spool/spd/spshed_reps</code>	error log file
<code>/usr/spool/spd/spufile23</code>	user data
<code>/usr/spool/spd/charges23</code>	user charges data (deprecated)
<code>/usr/spool/spd/spmm_jobi</code>	job memory-mapped hash file
<code>/usr/spool/spd/spmm_jobd</code>	job memory-mapped data file
<code>/usr/spool/spd/spmm_ptrs</code>	printers memory-mapped file
<code>/usr/spool/spd/spmm_xfer</code>	communication buffer memory-mapped file

5.1.1.2 Environment variables relevant to spshed

<code>SPOOLDIR</code>	alternative location for spool directory, by default <code>/usr/spool/spd</code> .
<code>SPHELDIR</code>	alternative location for help file directory, by default <code>/usr/spool/progs</code> .
<code>SPROGDIR</code>	alternative location for internal programs directory, by default <code>/usr/spool/progs</code> .

5.1.1.3 IPC facilities used by spshed

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

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

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

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

Some versions of `spshed` following version 23.51 of July 2002 may use memory-mapped files rather than shared memory. The files are held in the spool directory, by default `/usr/spool/spd`, and have the names `spmm_jobi`, `spmm_jobd`, `spmm_ptrs` and `spmm_xfer`.

A set of 5 semaphores, with the key `0x58691001` is created to interlock access to the shared memory segments.

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

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

5.1.1.4 Internet Ports used by `spshed`

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

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

5.1.1.5 Diagnostics from `spshed`

`spshed` is invoked from system startup procedures or other programs such as `spstart`. Thereafter it runs as a “daemon process” and diagnostics are not written to any terminal but to the file `/usr/spool/spd/spshed_reps`.

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

5.1.2 Xtnetserv

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

It is normally invoked by `spstart`.

Information, in respect of other machines to connect to, is read from the file `/etc/Xitext-hosts`.

5.1.2.1 Files used by `xtnetserv`

<code>/etc/Xitext-hosts</code>	host/client names and descriptions
<code>/etc/Xitext-config</code>	master configuration file
<code>/usr/spool/progs/int-config</code>	message file
<code>/usr/spool/spd</code>	spool directory
<code>/usr/spool/spd/spshed_reps</code>	error log file
<code>/usr/spool/spd/spufile</code>	user data

5.1.2.2 Environment variables relevant to xtnetsrv

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

5.1.2.3 Internet ports used by xtnetsrv

`xtnetsrv` uses 2 ports:

`xtnetsrv`

(N.B. no second “e”) to accept incoming jobs on TCP from `rspr`, and on UDP from the DOS/Windows interface, and 2 further ports

`xtapi`

to communicate with the API library.

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

5.1.2.4 Diagnostics from xtnetsrv

`xtnetsrv` runs as a “daemon process” and diagnostics, apart from those detected when it is first started, are not usually written to any terminal but to the file `/usr/spool/spd/spshed_reps`.

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

5.1.3 Spd

`Spd` is the spool despooler process for the **Xi-Text** spooler and document management system.

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

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

5.1.3.1 Files used by spd

<code>/etc/Xitext-config</code>	master configuration file
<code>/usr/spool/progs/int-config</code>	message file
<code>/usr/spool/spd</code>	spool directory
<code>/usr/spool/printers</code>	printer setup file directory
<code>/usr/spool/spd/spshed_reps</code>	error log file

5.1.3.2 Environment variables relevant to spd

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

5.1.3.3 Diagnostics from spd

`Spd` is usually invoked indirectly, via `spshed` from system startup procedures or other programs such as `spstart`.

Thereafter it runs as a “daemon process” and diagnostics are not written to any terminal but to the file `/usr/spool/spd/spshed_reps`.

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

5.1.4 Spdinit

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

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

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

5.1.4.1 Files used by spdinit

<code>/etc/Xitext-config</code>	master configuration file
<code>/usr/spool/progs/int-config</code>	message file
<code>/usr/spool/spd</code>	spool directory
<code>/usr/spool/printers</code>	printer setup file directory
<code>/usr/spool/spd/spshed_reps</code>	error log file

5.1.4.2 Environment variables relevant to spdinit

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

5.1.4.3 Diagnostics from spdinit

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

Diagnostics are not written to any terminal but to the file `/usr/spool/spd/spshed_reps`.

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

5.1.5 Spjobdump

`Spjobdump` is invoked by `spq`, `xspq`, `xmspq` and `sqdel` to unqueue jobs when required.

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

5.2 Message Handlers

5.2.1 Spmdisp

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

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

The messages are generated by `spmdisp` from the system message file, which by default is `/usr/spool/progs/int-config`.

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

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

The interface (options etc) are internal to **Xi-Text** and are not documented here.

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

5.2.1.1 Files used by spmdisp

<code>/usr/spool/progs/int-config</code>	message file
<code>~/.xitext</code>	user's home directory

5.2.1.2 Environment variables relevant to spmdisp

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

5.2.2 Spwrite

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

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

Note that `spwrite` is identical to **Xi-Batch's** `btwrite`, apart from using a different message file.

5.2.2.1 Files used by spwrite

`/usr/spool/progs/int-config` message file

5.2.2.2 Environment variables relevant to spwrite

`SPHELPPDIR` alternative location for help file directory, by default `/usr/spool/progs`.

5.2.3 Dossppwrite

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

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

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

Note that `dossppwrite` is identical to **Xi-Batch's** `dosbtwrite`, apart from using a different message file.

5.2.3.1 Files used by dossppwrite

`/usr/spool/progs/int-config` message file

5.2.3.2 Environment variables relevant to dosspwrite

`SPHELDIR` alternative location for help file directory, by default `/usr/spool/progs`.

5.2.4 Spexec

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

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

5.3 File Formats

5.3.1 /etc/Xitext-config

`/etc/Xitext-config` is an optional file for providing additional environment variables to component programs of the **Xi-Text** spooler and document management system.

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

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

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

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

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

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

5.3.1.1 Important notes for /etc/Xitext-config

1. Make sure that `/etc/Xitext-config` is world-readable!
2. Ensure that no part of **Xi-Text** is running before altering this file.

5.3.2 /etc/Xitext-hosts

`/etc/Xitext-hosts` is used to inform the Xi-text spooler and document management system, and in particular `spsched` and `xtnetserv`, which other host machines are to be attached.

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

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

5.3.2.1 Host name

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

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

5.3.2.2 Alias name

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

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

5.3.2.3 Flags

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

<code>probe</code>	Indicates that a datagram should be sent, and a reply awaited, from the host, before a full-blown connection is attempted. This is recommended wherever possible, or it is not sure in which order machines are booted.
<code>manual</code>	indicates that no connection at all is attempted. To connect to the machine in question, then <code>spconn</code> should be invoked.
<code>trusted</code>	indicates that the host is “trusted” by the current machine, which transmits information about Windows clients and their password validations, so the other host need not make such enquiries.
<code>client (username)</code>	indicates that no connection is attempted; the current machine is acting as a server for Windows clients. The specified username is to be considered as the owner of any jobs submitted, and the user to whom charges should be applied and to which privileges apply; see <code>spuser</code> . If <code>(username)</code> is omitted, then the Windows user is assumed, which should correspond to a user name on the host system.
<code>clientuser (name)</code>	Indicates that the whole entry identifies a “roaming user” who might be using one of several Windows clients, possibly with IP addresses assigned via DHCP. The host name in this case is replaced by the Windows user name, and the alias gives the Unix user name if different. If <code>(name)</code> is specified, then a password is demanded at the Windows client if the client's IP address does not match that of machine.
<code>dos (username)</code>	Is a synonym for <code>client (username)</code> kept for historical reasons.
<code>dosuser (name)</code>	Is a synonym for <code>clientuser (name)</code> kept for historical reasons.
<code>external</code>	Is a synonym for <code>client (no username)</code> kept for future extensions.
<code>pwchk</code>	Always demand the user's Unix password when first starting up.

5.3.2.4 Timeout

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

A default of 1000 seconds applies if none is specified.

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

5.3.2.5 Local address

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

```
localaddress 193.112.238.112
```

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

5.3.3 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.

The User Mapping file is used to convert Windows user names to UNIX user names for the Windows clients, or for the API functions.

Chapter 6

User Programs

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

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

More detailed descriptions of the interactive interfaces to `spq`, `spuser`, `xspq`, `xmispq`, `xspuser` and `xmispuser`. follow in the next two chapters, the descriptions here concentrating on the command line options to these programs.

6.1 Option Syntax

All of the options referred to in the descriptions of the shell-level programs for **Xi-Text** below may be supplied in a *configuration file* (q.v.), or in an environment variable whose name is the same as the calling program, except that it is in upper case¹.

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

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

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

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

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

```
-wmv
```

or they may be given separately as in

¹With the single exception of `spr`, which takes `SPOOLOPTS` instead, this being preserved for historical reasons.

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

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

```
-p150
```

and

```
-p 150
```

are acceptable and have the same effect.

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

```
+priority 150
```

6.1.1 Configuration files

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

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

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

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

The name assigned to is the same as that of the calling program but in upper case, for example that corresponding to `spq` is `SPQ` etc. This is the same as for the corresponding environment variable.

There is a single exception to the above statement, in that the environment variable corresponding to `spr` is `SPOOLOPTS`. This is retained for historical reasons.

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

Standard defaults	Each program has a set of standard defaults which are used to initialise the parameters when the program is invoked.	
User profile	In some cases, for example default priority, the user's profile as displayed by <code>spuser</code> is used to initialise the defaults.	
Home directory	The file <code>.xitext</code> in the home directory is read, and any options specified therein (i.e. with an assignment to the appropriate name) are interpreted.	
Environment	Any options specified in the appropriate environment variable (you will almost certainly have to use quotes when setting it via the shell in order to preserve the white space) are read and interpreted.	Most options have in-
Current Directory	The file <code>.xitext</code> is read, and any options specified therein (i.e. with an assignment to the appropriate name) are interpreted. Note that this may mean that the <code>.xitext</code> file is read twice if the command is run from the current directory.	
Command line	Any options specified on the command line are interpreted last. verses so that it is possible to reset anything which may have been set by previously-read options. Extra care should be taken with cumulative options, so that these are not doubled, especially in the case where the home directory is also the current directory.	

6.1.2 Option path

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

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

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

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

The default value of `XI_CONFIGPATH` is

```
~$LIBRARY:!:.-
```

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

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

The commands

```
+freeze-home
```

and

```
+freeze-current
```

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

Please note that `XI_CONFIGPATH`, with its default and interpretation is the same in other Xi Software products, such as **Xi-Batch**.

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

6.1.3 Message files

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

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

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

```
A300:?,explain
```

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

These sequences define

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

6.1.4 Location of message files

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

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

The search is in the following order:

- | | | |
|-------------------|---|--------------|
| Current Directory | If a configuration file in the current directory specifies a location for the message file, by means of an assignment to the relevant variable (for most shell-based commands this is <code>SPRESTCONF</code>), then this file is taken. Environment variables in the form <code>\$ABC</code> and users' home directories in the form <code>~user</code> are appropriately expanded. The sequence <code>\$0</code> is replaced by the name of the program being invoked. (This process may run recursively up to a level of 10) ² . | If a file is |
| Environment | If the relevant environment variable (for most shell-based commands this is <code>SPRESTCONF</code>) specifies a location, then this is taken. | |
| Home Directory | A configuration file in the home directory may specify a location for the message file. | |
| Default Location | If none of the above specify a replacement message file then the default location is taken. | |

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

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

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

For example if the normal message file for the command were

```
/usr/spool/progs/rest.help
```

and the user had specified in a configuration file

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

then if he or she were to run, say, `spr`, then the file

```
~/spr.help
```

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

```
~/rest.help
```

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

6.1.5 Path to locate message files

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

²I.e. If the environment variable expansion contains another `$name` construct, this will be expanded again, but only to 10 levels deep.

The default value of `XI_HELPPATH` is `.:!::~~$LIBRARY` giving the interpretation described above. Note that this is in the opposite order to `XI_CONFIGPATH`.

6.2 Submitting print jobs

6.2.1 spr

```
spr [ -options ] [ file ... ]
```

`spr` queues up one or more print jobs for the **Xi-Text** spooler.

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

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

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

6.2.1.1 Options

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

6.2.1.1.1 -? or +explain option

```
-?
--explain
+explain
```

This option causes a summary of the other options to be displayed without taking further action.

6.2.1.1.2 -a or +mail-attention option

```
-a
--mail-attention
+mail-attention
```

This option requests that a message be sent in the mail to the user if a job is awaiting attention, either because it reaches the top of the queue and the required form type is currently unavailable or because the form requires alignment.

6.2.1.1.3 -A or +write-attention option

```
-A
--write-attention
+write-attention
```

This option requests that a message be sent to every terminal at which the user is logged in, or by mail otherwise if the user is not logged in, if a job is awaiting attention, either because it reaches the top of the queue and the required form type is currently unavailable or because the form requires alignment.

6.2.1.1.4 -b or +noatt-message option

```
-b
--noatt-message
+noatt-message
```

This option turns off any job attention messages previously requested.

6.2.1.1.5 -c or +copies option

```
-c n
--copies n
+copies n
```

This option sets the number of copies to be printed to *n*, which may possibly be zero.

The maximum number of copies at a time may be restricted to a number for each user such as 10. See the output from [spuser](#) for information on the maximum number of copies.

6.2.1.1.6 -C or +classcode option

```
-C nnnn
--classcode nnnn
+classcode nnnn
```

This option takes an argument *nnnn* where *nnnn* is some collection of the letters **A** through to **P** inclusive (upper or lower case) with **-** to indicate ranges and requests that the job be given the specified class code.

The class code of a job may restrict which printers the job may be printed on, or which other users can display details of the job.

In normal circumstances (except where the user has *override class* privilege), the specified class code is reduced to the intersection of (i.e. is “anded” with) the specified class code and the user’s own class code. It is an error if the result is zero.

6.2.1.1.7 -d or +delimiter-number option

```
-d n
--delimiter-number n
+delimiter-number n
```

This option is used in conjunction with the `-Dstring` option to specify that a page is to be considered complete after reading *n* delimiter strings. The default value is 1.

6.2.1.1.8 -D or +delimiter option

```
-D string
--delimiter string
+delimiter string
```

This option specifies an alternative delimiter to the default of formfeed as being the end of a page. If a number is given using the `-dn` option, then the page is considered ended after that number of occurrences of the delimiter.

The string given may contain standard constructs such as `\n \r \t` etc to denote linefeed, carriage return etc. Remember that when the job is printed, the printer’s setup file may re-specify the delimiter, which will replace the delimiter specified by these options.

For jobs other than simple text, these options are probably best avoided.

6.2.1.1.9 -E or +external-system option

```
-E n
--external-system n
+external-system n
```

This option is used internally to signify originating system type. *n* may be a number, or the name of a system type given in `/etc/Xitext-extern`. This option may only be used by `root` or `spooler` regardless of other privileges set, as it is intended for use by `xtlpd` to set parameters on incoming jobs. Any errors in this parameter, including permissions, are silently ignored.

6.2.1.1.10 -f or +formtype option

```
-f string  
--formtype string  
+formtype string
```

This option specifies that the supplied form type (with optional suffix introduced by “.” or “-”) is used. Note that not all users may be allowed to select form types other than some pattern, causing the job to be rejected.

6.2.1.1.11 -F or +post-proc-flags option

```
-F string  
--post-proc-flags string  
+post-proc-flags string
```

This option causes the specified string to be passed as the value of the environment variable `SPOOLFLAGS` to any program invoked at the time the job is printed. This might be as a post-processing filter command, or alternatively some shell command invoked as (for example) a `docstart` string.

6.2.1.1.12 -h or +header option

```
-h string  
--header string  
+header string
```

This option supplies *string* as the title for each job, otherwise it is derived from the file name specified. The title appears on the banner page, if this is printed.

6.2.1.1.13 -i or +interpolate option

```
-i  
--interpolate  
+interpolate
```

This option causes any file name suffixes `.c`, `.ps` etc of the files from which jobs are created, to be appended to the form type, unless it already has a suffix, for example:

```
spr -i -f a4 x.c y.ps
```

would submit jobs using the files `x.c` and `y.ps` with form types `a4.c` and `a4.ps` respectively.

6.2.1.1.14 -I or +no-interpolate option

```
-I
--no-interpolate
+no-interpolate
```

This option cancels any previously-specified `-i` option.

6.2.1.1.15 -j or +job-wait-time option

```
-j n
--job-wait-time n
+job-wait-time n
```

This option is relevant only if input to `spr` or `rspr` is from a pipeline as in

```
program|spr ....
```

This causes a job to be submitted if no data is received over the pipe for `n` seconds.

`spr` will still wait, and may submit any number of subsequent jobs until the pipe is closed by the program generating the output.

Note that nothing can be done by `spr` to guess at what output might appear (possibly halfway through a line) if the program writing to the pipe does not flush its output buffer after each occasion that it writes something.

Specify a parameter of zero to cancel any previous `-j` option, if required.

6.2.1.1.16 -l or +local-only option

```
-l
--local-only
+local-only
```

This option, in a networked environment, requests that the job be printed only on the printers local to the machine at which the job is submitted.

If your copy of **Xi-Text** is set as “standalone” this will always be set (and you may have to unset it on existing jobs if you change it to “networked”).

6.2.1.1.17 -L or +network-wide option

```
-L  
--network-wide  
+network-wide
```

This option cancels the request that the job or jobs be printed only on printers local to the machine at which the job is submitted.

Note that this option will have no effect if your copy of **Xi-Text** is set as “standalone”.

6.2.1.1.18 -m or +mail-message option

```
-m  
--mail-message  
+mail-message
```

This option requests a message to be sent in the mail to the user on completion or termination of the job.

6.2.1.1.19 -n or +delay-for option

```
-n nn  
--delay-for nn  
+delay-for nn
```

This option specifies that the job should be held on the queue for at least *nn* minutes from the current time before being printed. The time may alternatively be specified as *hh:mm* or as *hh:mm:ss*, specifying a delay in hours and minutes, or hours, minutes and seconds.

To cancel this option, an argument of zero should be given.

6.2.1.1.20 -N or +delay-until option

```
-N time  
--delay-until time  
+delay-until time
```

This option is an alternative to the `-n` option to specify the earliest time at which the job is to be printed. The argument may be *hh:mm* or *hh:mm:ss* to give the time of day in 24-hour clock notation. If the time has passed, then tomorrow is assumed.

Alternatively a date and a comma may be prefixed to the time in the form *mm/dd* or *dd/mm* depending upon the local convention for date format used. Thus the resulting argument might be

```
-N 10/11,12:30
```

If both this and the `-n` option are given, the most recently specified will apply.

6.2.1.1.21 `-o` or `+originating-host` option

```
-o host
--originating-host host
+originating-host host
```

This option is Used internally to signify originating host name, in place of the local host. The host name given should appear in the host file `/etc/Xitext-hosts`, possibly with the `external` keyword. This option may only be specified by `spooler` or `root` users, but is silently ignored in other cases as are all other errors. It is intended for use by internal programs only.

6.2.1.1.22 `-O` or `+odd-even-flags` option

```
-O flag
--odd-even-flags flag
+odd-even-flags flag
```

This option arranges for only odd-numbered pages to be printed, only even-numbered pages to be printed, odd-numbered pages to be printed on odd-numbered copies and even-numbered pages on even-numbered copies, or vice-versa.

The argument can be

- cancel this option
- O print odd numbered pages
- E print even-numbered pages
- A print odd-numbered pages, then even-numbered ones
- B print even-numbered pages, then odd-numbered ones

For example

```
spr -c2 -OA
```

will print odd-numbered pages on the first copy and even ones on the second.

For this to work properly, page delimiters must be set appropriately.

6.2.1.1.23 -p or +priority option

```
-p n  
--priority n  
+priority n
```

This option specifies the priority of the job, between 1 (lowest) and 255 (highest) or some narrower range to which the user is limited.

Increasing the priority of a job increases its chances of being printed earlier than it otherwise would be.

6.2.1.1.24 -P or +printer option

```
-P name  
--printer name  
+printer name
```

This option specifies that the job is to be sent to the given printer, as opposed to printing it on the first available printer with the given form type.

The name given may be a pattern to select any printer whose name matches the pattern.

Note that a user may be limited to just one printers or one of a range of printers, see the output of [spuser](#).

To “turn off” a printer name previously specified by a preceding `-Pname` option, put a single `-` sign as the printer name.

6.2.1.1.25 -q or +retain option

```
-q  
--retain  
+retain
```

This option requests that the job or jobs be retained on the queue with copies set to zero after printing, for explicit deletion, or automatically at the expiry of the timeout (as set by the `-t` option).

Some printers may be set up to enforce this on all jobs anyway, see page [70](#).

6.2.1.1.26 -Q or +host option

```
-Q host  
--host host  
+host host
```

This option indicates that the job should be sent to the given server rather than queued on the local server.

Its effect on [spr](#) is to invoke [rspr](#) directly with the same command-line options.

6.2.1.1.27 -r or +banner option

```
-r  
--banner  
+banner
```

This option restores banner pages previously suppressed using `-s`. Note that some form types may be set up never to use banners, or always to use banners, regardless of this option.

6.2.1.1.28 -R or +page-range option

```
-R m-n  
--page-range m-n  
+page-range m-n
```

This option specifies that pages `m` through to `n` inclusive are to be printed. This does of course assume that the job has recognisable pages. If `m` or `n` are omitted, then “the beginning” or “the end” respectively is assumed, so `-R 3-5` would print pages 3 to 5, `-7` would print pages 1 to 7 inclusive, and `-R 4-` would print page 4 to the end inclusive. `-R 1-` would turn this option off by selecting 1 to the end.

6.2.1.1.29 -s or +no-banner option

```
-s  
--no-banner  
+no-banner
```

This option suppresses any banner page (large letter user name etc) which is printed before the job itself. Note that some form types may be set up always or never to print banners regardless of this option.

6.2.1.1.30 -t or +printed-timeout option

```
-t n  
--printed-timeout n  
+printed-timeout n
```

This option specifies that if a job is held on the queue after being printed it will be automatically deleted after the given number of hours, which may be 1 to 32767 (nearly 4 years).

The default if this is not specified is 24, i.e. 1 day.

6.2.1.1.31 -T or +not-printed-timeout option

```
-T n  
--not-printed-timeout n  
+not-printed-timeout n
```

This option specifies that if held on the queue *without* being printed, the job will be deleted automatically. The default value is 168 hours (1 week), and the maximum value is 32767 hours (nearly 4 years).

6.2.1.1.32 -u or +post-user option

```
-u name  
--post-user name  
+post-user name
```

This option requests that the specified user name be substituted for the submitting user on the banner page optionally printed at the start of the job. The job still remains the responsibility of the submitting user.

To “turn off” a user name specified in a previous `-u` option, put a single `-` sign as the user name.

The user name must exist on the machine to which the job is queued, but this will be ignored if not.

6.2.1.1.33 -U or +originating-user option

```
-U user  
--originating-user user  
+originating-user user
```

This option is Used internally to signify originating user name. This may only be specified by users with *masquerade as other users* privilege (by default only `spooler` and `root`) to have any effect, otherwise the option is silently ignored, as are all errors. It is intended for internal use only by other programs.

6.2.1.1.34 -V or +toggle-verbose option

```
-V  
--toggle-verbose  
+toggle-verbose
```

This option, alternately with successive uses turns on or off the verbose switch. This causes job number information to be output on standard error when job are submitted.

6.2.1.1.35 +verbose option

```
--verbose  
+verbose
```

There is no default letter option (however it is possible to create one by editing the message file).

6.2.1.1.36 +no-verbose option

```
--no-verbose  
+no-verbose
```

This option turns off the verbose switch. There is no default letter option (however it is possible to create one by editing the message file).

6.2.1.1.37 -w or +write-message option

```
-w  
--write-message  
+write-message
```

This option requests a message to be sent to every terminal at which the user is logged in, or by mail otherwise if the user is not logged in, when the job is completed or terminated.

6.2.1.1.38 -x or +no-message option

```
-x  
--no-message  
+no-message
```

This option turns off any job completion messages previously requested with `-m` or `-w`.

6.2.1.1.39 -z or +no-retain option

```
-z  
--no-retain  
+no-retain
```

This option cancels a request that the job or jobs be retained on the queue after printing.

Note that if the printer is set to always retain jobs after printing, this will have no effect.

6.2.1.1.40 -Z or +job-size-limit option

```
-Z limit  
--job-size-limit limit  
+job-size-limit limit
```

This option limits the size of jobs to the given limit, which may be a number, giving a size in bytes, or it may be suffixed with `P` to indicate the number of pages.

If a job exceeds the limit it is truncated with a warning message, but a job is still created. If the limit is prefixed with an `E`, then the warning becomes an error, and no job is created. Supply an argument of a single `-` to turn off this option.

6.2.1.1.41 +freeze-current option

```
--freeze-current  
+freeze-current
```

Save all the current options in a `.xitext` file in the current directory with keyword `SPOOLOPTS`.

No further action will be taken if this option is specified.

6.2.1.1.42 +freeze-home option

```
--freeze-home  
+freeze-home
```

Save all the current options in a `.xitext` file in the user's home directory with keyword `SPOOLOPTS`.

No further action will be taken if this option is specified.

6.2.1.2 Environment variables used by spr

`SPOOLOPTS` space-separated options to override defaults for `spr`.

`SPRESTCONF` location of alternative help file.

6.2.1.3 Important Note

N.B. Please note that the scheduler `spshed` is no longer automatically started by `spr` if **Xi-Text** is not running, this should be done by `spstart` instead.

6.2.1.4 Messages via terminal and e-mail

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

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

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

(Do not forget you can override the `MAILER` program to change this behaviour if required as described on page 303).

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

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

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

6.2.1.5 Queue Timeouts

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

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

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

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

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

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

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

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

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

6.2.1.6 Delay times

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

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

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

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

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

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

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

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

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

6.2.1.7 Interaction of spr and rspr

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

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

6.2.2 rspr

```
rspr [ -options ] [ file ... ]
```

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

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

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

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

6.2.2.1 Options

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

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

6.2.2.1.1 **-?** or **+explain** option

```
-?  
--explain  
+explain
```

This option causes a summary of the other options to be displayed without taking further action.

6.2.2.1.2 **-a** or **+mail-attention** option

```
-a  
--mail-attention  
+mail-attention
```

This option requests that a message be sent in the mail to the user if a job is awaiting attention, either because it reaches the top of the queue and the required form type is currently unavailable or because the form requires alignment.

6.2.2.1.3 **-A** or **+write-attention** option

```
-A  
--write-attention  
+write-attention
```

This option requests that a message be sent to every terminal at which the user is logged in, or by mail otherwise if the user is not logged in, if a job is awaiting attention, either because it reaches the top of the queue and the required form type is currently unavailable or because the form requires alignment.

6.2.2.1.4 **-b** or **+noatt-message** option

```
-b  
--noatt-message  
+noatt-message
```

This option turns off any job attention messages previously requested.

6.2.2.1.5 **-c** or **+copies** option

```
-c n  
--copies n  
+copies n
```

This option sets the number of copies to be printed to *n*, which may possibly be zero.

The maximum number of copies at a time may be restricted to a number for each user such as 10. See the output from `spuser` for information on the maximum number of copies.

6.2.2.1.6 -C or +classcode option

```
-C nnnn
--classcode nnnn
+classcode nnnn
```

This option takes an argument *nnnn* where *nnnn* is some collection of the letters **A** through to **P** inclusive (upper or lower case) with `-` to indicate ranges and requests that the job be given the specified class code.

The class code of a job may restrict which printers the job may be printed on, or which other users can display details of the job.

In normal circumstances (except where the user has *override class* privilege), the specified class code is reduced to the intersection of (i.e. is “anded” with) the specified class code and the user’s own class code. It is an error if the result is zero.

6.2.2.1.7 -d or +delimiter-number option

```
-d n
--delimiter-number n
+delimiter-number n
```

This option is used in conjunction with the `-Dstring` option to specify that a page is to be considered complete after reading *n* delimiter strings. The default value is 1.

6.2.2.1.8 -D or +delimiter option

```
-D string
--delimiter string
+delimiter string
```

This option specifies an alternative delimiter to the default of formfeed as being the end of a page. If a number is given using the `-dn` option, then the page is considered ended after that number of occurrences of the delimiter.

The string given may contain standard constructs such as `\n \r \t` etc to denote linefeed, carriage return etc. Remember that when the job is printed, the printer’s setup file may re-specify the delimiter, which will replace the delimiter specified by these options.

For jobs other than simple text, these options are probably best avoided.

6.2.2.1.9 -f or +formtype option

```
-f string  
--formtype string  
+formtype string
```

This option specifies that the supplied form type (with optional suffix introduced by “.” or “-”) is used. Note that not all users may be allowed to select form types other than some pattern, causing the job to be rejected.

6.2.2.1.10 -F or +post-proc-flags option

```
-F string  
--post-proc-flags string  
+post-proc-flags string
```

This option causes the specified string to be passed as the value of the environment variable `SPOOLFLAGS` to any program invoked at the time the job is printed. This might be as a post-processing filter command, or alternatively some shell command invoked as (for example) a `docstart` string.

6.2.2.1.11 -h or +header option

```
-h string  
--header string  
+header string
```

This option supplies *string* as the title for each job, otherwise it is derived from the file name specified. The title appears on the banner page, if this is printed.

6.2.2.1.12 -i or +interpolate option

```
-i  
--interpolate  
+interpolate
```

This option causes any file name suffixes `.c`, `.ps` etc of the files from which jobs are created, to be appended to the form type, unless it already has a suffix, for example:

```
rspr -i -f a4 x.c y.ps
```

would submit jobs using the files `x.c` and `y.ps` with form types `a4.c` and `a4.ps` respectively.

6.2.2.1.13 -I or +no-interpolate option

```
-I
--no-interpolate
+no-interpolate
```

This option cancels any previously-specified `-i` option.

6.2.2.1.14 -j or +job-wait-time option

```
-j n
--job-wait-time n
+job-wait-time n
```

This option is relevant only if input to `spr` or `rspr` is from a pipeline as in

```
program|rspr ....
```

This causes a job to be submitted if no data is received over the pipe for `n` seconds.

`rspr` will still wait, and may submit any number of subsequent jobs until the pipe is closed by the program generating the output.

Note that nothing can be done by `rspr` to guess at what output might appear (possibly halfway through a line) if the program writing to the pipe does not flush its output buffer after each occasion that it writes something.

Specify a parameter of zero to cancel any previous `-j` option, if required.

6.2.2.1.15 -l or +local-only option

```
-l
--local-only
+local-only
```

This option, in a networked environment, requests that the job be printed only on the printers local to the machine at which the job is submitted.

6.2.2.1.16 -L or +network-wide option

```
-L
--network-wide
+network-wide
```

This option cancels the request that the job or jobs be printed only on printers local to the machine at which the job is submitted.

6.2.2.1.17 **-m or +mail-message option**

```
-m  
--mail-message  
+mail-message
```

This option requests a message to be sent in the mail to the user on completion or termination of the job.

6.2.2.1.18 **-n or +delay-for option**

```
-n nn  
--delay-for nn  
+delay-for nn
```

This option specifies that the job should be held on the queue for at least *nn* minutes from the current time before being printed. The time may alternatively be specified as *hh:mm* or as *hh:mm:ss*, specifying a delay in hours and minutes, or hours, minutes and seconds.

To cancel this option, an argument of zero should be given.

6.2.2.1.19 **-N or +delay-until option**

```
-N time  
--delay-until time  
+delay-until time
```

This option is an alternative to the `-n` option to specify the earliest time at which the job is to be printed. The argument may be *hh:mm* or *hh:mm:ss* to give the time of day in 24-hour clock notation. If the time has passed, then tomorrow is assumed.

Alternatively a date and a comma may be prefixed to the time in the form *mm/dd* or *dd/mm* depending upon the local convention for date format used. Thus the resulting argument might be

```
-N 10/11,12:30
```

If both this and the `-n` option are given, the most recently specified will apply.

6.2.2.1.20 -O or +odd-even-flags option

```
-O flag  
--odd-even-flags flag  
+odd-even-flags flag
```

This option arranges for only odd-numbered pages to be printed, only even-numbered pages to be printed, odd-numbered pages to be printed on odd-numbered copies and even-numbered pages on even-numbered copies, or vice-versa.

The argument can be

- cancel this option
- O print odd numbered pages
- E print even-numbered pages
- A print odd-numbered pages, then even-numbered ones
- B print even-numbered pages, then odd-numbered ones

For example

```
rspr -c2 -OA
```

will print odd-numbered pages on the first copy and even ones on the second.

For this to work properly, page delimiters must be set appropriately.

6.2.2.1.21 -p or +priority option

```
-p n  
--priority n  
+priority n
```

This option specifies the priority of the job, between 1 (lowest) and 255 (highest) or some narrower range to which the user is limited.

Increasing the priority of a job increases its chances of being printed earlier than it otherwise would be.

6.2.2.1.22 -P or +printer option

```
-P name  
--printer name  
+printer name
```

This option specifies that the job is to be sent to the given printer, as opposed to printing it on the first available printer with the given form type.

The name given may be a pattern to select any printer whose name matches the pattern.

Note that a user may be limited to just one printers or one of a range of printers, see the output of `spuser`.

To “turn off” a printer name previously specified by a preceding `-Pname` option, put a single `-` sign as the printer name.

6.2.2.1.23 `-q` or `+retain` option

```
-q
--retain
+retain
```

This option requests that the job or jobs be retained on the queue with copies set to zero after printing, for explicit deletion, or automatically at the expiry of the timeout (as set by the `-t` option).

Some printers may be set up to enforce this on all jobs anyway, see page 70.

6.2.2.1.24 `-Q` or `+host` option

```
-Q host
--host host
+host host
```

This option directs `rspr` to send the job or jobs to the given hostname. Note that **Xi-Text** must be running in “networked” mode on the receiving host.

6.2.2.1.25 `-r` or `+banner` option

```
-r
--banner
+banner
```

This option restores banner pages previously suppressed using `-s`. Note that some form types may be set up never to use banners, or always to use banners, regardless of this option.

6.2.2.1.26 `-R` or `+page-range` option

```
-R m-n
--page-range m-n
+page-range m-n
```

This option specifies that pages *m* through to *n* inclusive are to be printed. This does of course assume that the job has recognisable pages. If *m* or *n* are omitted, then “the beginning” or “the end” respectively is assumed, so `-R 3-5` would print pages 3 to 5, `-7` would print pages 1 to 7 inclusive, and `-R 4-` would print page 4 to the end inclusive. `-R 1-` would turn this option off by selecting 1 to the end.

6.2.2.1.27 `-s` or `+no-banner` option

```
-s  
--no-banner  
+no-banner
```

This option suppresses any banner page (large letter user name etc) which is printed before the job itself. Note that some form types may be set up always or never to print banners regardless of this option.

6.2.2.1.28 `-t` or `+printed-timeout` option

```
-t n  
--printed-timeout n  
+printed-timeout n
```

This option specifies that if a job is held on the queue after being printed it will be automatically deleted after the given number of hours, which may be 1 to 32767 (nearly 4 years).

The default if this is not specified is 24, i.e. 1 day.

6.2.2.1.29 `-T` or `+not-printed-timeout` option

```
-T n  
--not-printed-timeout n  
+not-printed-timeout n
```

This option specifies that if held on the queue *without* being printed, the job will be deleted automatically. The default value is 168 hours (1 week), and the maximum value is 32767 hours (nearly 4 years).

6.2.2.1.30 `-u` or `+post-user` option

```
-u name  
--post-user name  
+post-user name
```

This option requests that the specified user name be substituted for the submitting user on the banner page optionally printed at the start of the job. The job still remains the responsibility of the submitting user.

To “turn off” a user name specified in a previous `-u` option, put a single `-` sign as the user name. The user name must exist on the machine to which the job is queued, but this will be ignored if not.

6.2.2.1.31 `-V` or `+toggle-verbose` option

```
-V
--toggle-verbose
+toggle-verbose
```

This option, alternately with successive uses turns on or off the verbose switch. This causes job number information to be output on standard error when job are submitted.

6.2.2.1.32 `+verbose` option

```
--verbose
+verbose
```

There is no default letter option (however it is possible to create one by editing the message file).

6.2.2.1.33 `+no-verbose` option

```
--no-verbose
+no-verbose
```

This option turns off the verbose switch. There is no default letter option (however it is possible to create one by editing the message file).

6.2.2.1.34 `-w` or `+write-message` option

```
-w
--write-message
+write-message
```

This option requests a message to be sent to every terminal at which the user is logged in, or by mail otherwise if the user is not logged in, when the job is completed or terminated.

6.2.2.1.35 -x or +no-message option

```
-x  
--no-message  
+no-message
```

This option turns off any job completion messages previously requested with `-m` or `-w`.

6.2.2.1.36 -z or +no-retain option

```
-z  
--no-retain  
+no-retain
```

This option cancels a request that the job or jobs be retained on the queue after printing.

Note that if the printer is set to always retain jobs after printing, this will have no effect.

6.2.2.1.37 -Z or +job-size-limit option

```
-Z limit  
--job-size-limit limit  
+job-size-limit limit
```

This option limits the size of jobs to the given limit, which may be a number, giving a size in bytes, or it may be suffixed with `P` to indicate the number of pages.

If a job exceeds the limit it is truncated with a warning message, but a job is still created. If the limit is prefixed with an `E`, then the warning becomes an error, and no job is created. Supply an argument of a single `-` to turn off this option.

6.2.2.1.38 +freeze-current option

```
--freeze-current  
+freeze-current
```

Save all the current options in a `.xitext` file in the current directory with keyword `RSPR`.

No further action will be taken if this option is specified.

6.2.2.1.39 +freeze-home option

```
--freeze-home
+freeze-home
```

Save all the current options in a `.xitext` file in the user's home directory with keyword `RSPR`.

No further action will be taken if this option is specified.

6.2.2.2 Environment variables used by rspr

`RSPR` space-separated options to override defaults for `rspr`.
`SPRESTCONF` location of alternative help file.

6.2.3 lpcover

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

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

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

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

6.2.3.1 Options

6.2.3.1.1 -c option

```
-c
```

This option has no effect, but generates a warning message on standard error, as with **Xi-Text** spool files are always copied.

6.2.3.1.2 -d option

```
-d name
```

This option specifies *name* as the printer destination. The destination may also be specified in the environment variable `LPDEST`, which this option overrides.

This is translated directly into a `-P` option to `spr`.

6.2.3.1.3 -f option

-f form

This option specifies the supplied form type. This is translated directly into an *-f* option to *spr*.

6.2.3.1.4 -H option

-H arg

There is no direct mapping for the *-H* argument.

6.2.3.1.5 -m option

-m

This option arranges for mail about completion to be mailed to the user.

This is translated directly into an *-m* option to *spr*.

6.2.3.1.6 -n option

-n n

This option specifies the number of copies and is translated directly into a *-c* option to *spr*.

6.2.3.1.7 -o option

-o arg

This option passes other options to *lp*. The only argument supported is *-o nobanner*, to suppress the header page, which is interpreted and translated to the *-s* option to *spr*.

6.2.3.1.8 -P option

-P range

This is translated into *-R* page range arguments for *spr*.

6.2.3.1.9 -q option

`-q n`

This option is translated into a `-p` argument for `spr` scaled appropriately, taking into account the user's maximum, minimum and default priorities.

6.2.3.1.10 -S option

`-S`

This option is ignored.

6.2.3.1.11 -s option

`-s`

This turns off the `-v` argument for `spr` which is normally included.

6.2.3.1.12 -T option

`-T`

This option is ignored.

6.2.3.1.13 -t option

`-t arg`

This option is translated to a `-h` (title) argument to `spr`.

6.2.3.1.14 -w option

`-w`

This option is translated directly to the `-w` (write message) option to `spr`.

6.2.3.1.15 -y option

`-y`

This option is ignored.

6.2.3.2 Environment variables used

`LPDEST` Specify destination for `lpcover`.

6.2.3.3 Diagnostics

Various warnings about unimplemented features are printed by `lpcover`.

Others may be generated by software invoked.

6.3 Managing the spooler

6.3.1 spstart – for system

```
spstart jobs ptrs
```

`spstart` may be used to start the **Xi-Text** scheduler, allowing space for the given number of jobs and printers in shared memory.

`spstart` may also be used to start printers, or to halt, change the form type and to restart in one operation – see page 154.

```
spstart 2000 20
```

Will start the **Xi-Text** scheduler, reserving space for 2000 jobs and 20 printers.

We recommend you always do this, allowing space for the maximum number of jobs and printers you will ever have, as it is hard to “grow” shared memory later.

6.3.2 sstop

```
sstop [ -y ]
```

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

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

6.3.2.1 Options for sstop

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

6.3.2.1.1 `-y` option

`-y`

This option suppresses the request for confirmation.

6.3.3 `spconn`

```
spconn host
```

`spconn` instructs the **Xi-Text** scheduler to attempt to raise a connection to the given host.

It returns immediately; however the connection may take some time and the problem, if any, will only be recorded in the system log file, `spshed_reps` in the spool directory, by default `/usr/spool/spd`. This can be quickly viewed from the various interactive view utilities, for example `spq`.

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

6.3.4 `spdisconn`

```
spdisconn host
```

`spdisconn` instructs the **Xi-Text** scheduler to close a connection to the given host, which should be currently active.

The command returns immediately; however the disconnection may take some time and the error, if any, will only be recorded in the system log file, `spshed_reps` in the spool directory, by default `/usr/spool/spd`. This can be quickly viewed from the various interactive view utilities, for example `spq`.

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

6.3.5 `ssuspend`

```
ssuspend [ -y ] [ n ]
```

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

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

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

6.3.5.1 Options

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

6.3.5.1.1 `-y` option

`-y`

Do not ask for confirmation before proceeding.

6.3.6 `srelease`

```
srelease
```

`srelease` releases scheduling again where it has previously been halted with `ssuspend`. It has no effect if scheduling is not currently suspended. There are no options.

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

6.3.7 `hostedit`

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

`hostedit` is a simple curses-based program to edit host tables for `/etc/Xitext-hosts`, the host table for **Xi-Text**.

`xhostedit` is a GTK+ alternative which may be available to you.

It knows about local addresses, using web servers to get the local address, Windows clients, DHCP, trusted hosts (although this is now deprecated), manual connections, probes and timeouts.

That said, you may not need to use it for a straightforward network connection to another host as it is no longer compulsory to have details in the hosts file.

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

Alternatively use the `-I` option to edit a file in place.

Normally this would be run as follows:

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

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

6.3.7.1 Options

The options for `xhostedit` are the same as for `hostedit`, except that the former may take options interpreted by GTK+ as well.

6.3.7.1.1 -o option

```
-o file
```

Output to the named file rather than Standard Output

6.3.7.1.2 -s option

```
-s char
```

Where `char` is `h` or `i`.

Sort display by host name or by IP address.

6.3.7.1.3 -I option

```
-I
```

Edit the named file, which must always be given but need not exist, in place.

6.3.7.2 Commands

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

<code>k</code> or cursor up	Move cursor up.
<code>j</code> or cursor down	Move cursor down.
<code>N</code> or next page	Scroll down a screenful.
<code>P</code> or previous page	Scroll up a screenful.
<code>q</code>	Quit and write hosts file.
<code>a</code>	Create a new hosts entry.
<code>c</code>	Edit the selected hosts entry.
<code>d</code>	Delete the selected hosts entry.
<code>l</code>	Edit the local address.
<code>L</code>	Set the local address from the selected host.
<code>w</code>	Set the local address from a web server.
<code>u</code>	Set the default user name for DHCP clients.

6.4 Managing print jobs (command line)

6.4.1 `sqlist`

```
sqlist [ -options ] [ job number ... ]
```

`sqlist` outputs a table on standard output showing jobs on the **Xi-Text** spool queue, including those queued on other hosts on the network.

Each line of the output corresponds to a single job, and by default the output is generally similar to the default format of the jobs screen of the `spq` program, something like:

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

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

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

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

6.4.1.1 Options

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

6.4.1.1.1 -? or +explain option

```
-?  
--explain  
+explain
```

This option causes a summary of the other options to be displayed without taking further action.

6.4.1.1.2 -C or +classcode option

```
-C nnnn  
--classcode nnnn  
+classcode nnnn
```

This option restricts attention to jobs matching (i.e. having at least one bit in common with) the classcode given.

Unless he/she has *override class* privilege, the class code given will be masked “anded” with the user’s class code and the result (which must not be zero, otherwise an error message will be given) used.

6.4.1.1.3 -D or +default-format option

```
-D  
--default-format  
+default-format
```

This option cancels a `-F` option and restores the default format.

6.4.1.1.4 -e or +no-page-counts option

```
-e  
--no-page-counts  
+no-page-counts
```

This option stops `sqlist` trying to represent page numbers and counts on the output (this only affects the default format).

6.4.1.1.5 -E or +page-counts option

```
-E  
--page-counts  
+page-counts
```

This option cancels the `-e` option if previously given.

This only affects the default format.

6.4.1.1.6 -F or +format option

```
-F string  
--format string  
+format string
```

This option specifies a user-defined output format. The string argument contains characters to be printed and symbols introduced by `%` indicating the required attributes of the jobs to be displayed.

The format symbols are listed on page [135](#).

6.4.1.1.7 -H or +header option

```
-H  
--header  
+header
```

This option causes a header to be generated for each column of the output according the field displayed there.

The headers are taken from the message file, by default `/usr/spool/progs/rest.help`, which may be adjusted as required.

6.4.1.1.8 -l or +local-only option

```
-l  
--local-only  
+local-only
```

This option indicates to display only jobs local to the current host.

6.4.1.1.9 -N or +no-header option

```
-N
--no-header
+no-header
```

This option turns off the header requested by a previously-requested `-H` option.

6.4.1.1.10 -n or +no-view-job option

```
-n
--no-view-job
+no-view-job
```

The option cancels a previously-requested `-V` option.

6.4.1.1.11 -P or +view-pages option

```
-P range
--view-pages range
+view-pages range
```

In conjunction with the `-V` option, specify a range of pages to display.

A range may consist of single page numbers or subranges separated by commas, thus:

```
-V -P 1
-V -P 1,3,4-9
-V -P 2-7,12-
```

such arguments would respectively select page 1, pages 1, 3 and 4 to 9 and pages 2 to 7 and 12 to the end. Omitted numbers either side of the `-` denote “the beginning” and “the end”.

6.4.1.1.12 -p or +printed-jobs option

```
-p
--printed-jobs
+printed-jobs
```

This option restricts the display to jobs which have been printed.

6.4.1.1.13 -q or +list-only option

```
-q name  
--list-only name  
+list-only name
```

This option restricts the display to jobs for, or which could be printed on the specified subset of printers.

The set of printers may contain just one name, a list of names or a list of patterns for matching printer names. Shell-like pattern matching applies to the pattern. You may well need to put quotes around the pattern to avoid misinterpretation by the shell.

6.4.1.1.14 -r or +remotes option

```
-r  
--remotes  
+remotes
```

This option display jobs local to the current host and connected hosts, cancelling a preceding `-l` option.

6.4.1.1.15 -t or +just-title option

```
-t string  
--just-title string  
+just-title string
```

This option displays only jobs whose title matches the pattern given.

6.4.1.1.16 -u or +just-user option

```
-u user  
--just-user user  
+just-user user
```

This option restricts the display to jobs owned by the given user or list of users given by a pattern.

6.4.1.1.17 -V or +view-job option

```
-V  
--view-job  
+view-job
```

Instead of listing attributes of jobs, send the text of the job or jobs to the standard output.

6.4.1.1.18 -Y or +all-jobs option

```
-Y  
--all-jobs  
+all-jobs
```

This option causes all jobs to be displayed which might otherwise be suppressed by previously-specified `-y` or `-p` options.

6.4.1.1.19 -y or +unprinted-jobs option

```
-y  
--unprinted-jobs  
+unprinted-jobs
```

This option causes only jobs which have not been printed to be displayed.

6.4.1.1.20 -z or +include-null option

```
-z  
--include-null  
+include-null
```

This option, in conjunction with the `-q` option, includes jobs with no destination printer specified.

6.4.1.1.21 -Z or +no-include-null option

```
-Z  
--no-include-null  
+no-include-null
```

This option, in conjunction with the `-q` option, excludes jobs with no destination printer specified.

6.4.1.1.22 +freeze-current option

```
--freeze-current  
+freeze-current
```

Save all the current options in a `.xitext` file in the current directory with keyword `SQLIST`.

No further action will be taken if this option is specified.

6.4.1.1.23 +freeze-home option

```
--freeze-home
+freeze-home
```

Save all the current options in a `.xitext` file in the user's home directory with keyword `SQLIST`.

No further action will be taken if this option is specified.

6.4.1.2 Patterns

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

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

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

For example

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

6.4.1.3 Format codes

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

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

%% insert a single % sign
 %A *wattn* if the write attention message flag is set.
 %a *mattn* if the mail attention message flag is set.
 %C class code
 %c copies (right-justified)
 %D page delimiter
 %d delimiter number
 %F post-processing flags
 %f form type
 %G "halted at" page number.
 %H hold time
 %h job title
 %K size of job in bytes
 %L position reached in bytes.
 %l *local* if local job only
 %m *mail* if mail completion flag set.
 %N job number, prefixed by host name if remote
 %O odd/even/swap flags.
 %o originating host (might be different if via *rspr*)
 %P printer destination
 %p priority.
 %Q page reached.
 %q *retain* if retained on queue
 %R page range
 %S size in pages
 %s *nohdr* if header suppressed
 %T delete time if not printed
 %t delete time if printed
 %U user name to be posted to
 %u user name owned by
 %W submission or start time
 %w *write* if write completion message flag set

Note that the various strings such as *nohdr* etc are read from the message file also, so it is possible to modify them as required by the user.

The default format is

```
%N %u %h %f %Q %S %c %p %P
```

with the (default) `-E` option and

```
%N %u %h %f %L %K %c %p %P
```

with the `-e` option.

6.4.2 sqdel

```
sqdel [ -options ] job number ...
```

`sqdel` deletes one or more jobs from the **Xi-Text** job spool queue, which may be on the local host or other hosts on the network. Jobs are specified by the job number, which is reported by `splist` in the first column of the default output, or by `spr` or `rspr` when the job is queued with the verbose option.

Remote jobs are specified by prefixing the job number by the host name and a colon, thus `host:1234`. Leading zeroes in the job number may be omitted.

`sqdel` will not delete jobs not marked as “printed” unless the `-y` option is given.

6.4.2.1 Options

The environment variable on which options may be supplied is `SQDEL` and the environment variable to specify the help file is `SPRESTCONF`.

6.4.2.1.1 `-?` or `+explain` option

```
-?
--explain
+explain
```

This option causes a summary of the other options to be displayed without taking further action.

6.4.2.1.2 `-C` or `+classcode` option

```
-C nnnn
--classcode nnnn
+classcode nnnn
```

This option may be used to set a class code to narrow or expand the range of jobs from which jobs are selected. It is probably only relevant where the user has *override class* privilege and it is required to expand the default class code to select additional jobs.

6.4.2.1.3 -d or +delete option

```
-d  
--delete  
+delete
```

This option cancels a previously-specified `-k` option. The job or jobs are deleted.

6.4.2.1.4 -D or +directory option

```
-D name  
--directory name  
+directory name
```

Save unqueued jobs to the selected directory rather than the current directory.

6.4.2.1.5 -e or +do-not-unqueue option

```
-e  
--do-not-unqueue  
+do-not-unqueue
```

This option cancels a previously-specified `-u` option. The jobs will just be deleted, not unqueued.

6.4.2.1.6 -J or +job-prefix option

```
-J name  
--job-prefix name  
+job-prefix name
```

This option specifies the given name as the prefix for the job file, followed by the job number, to be used by the `-u` option rather than the default of `J` (which in turn may be changed by editing the message file).

6.4.2.1.7 -k or +do-not-delete option

```
-k  
--do-not-delete  
+do-not-delete
```

This option suppresses the delete otherwise performed by the `-u` option.

6.4.2.1.8 -n or +keep-unprinted option

```
-n  
--keep-unprinted  
+keep-unprinted
```

This is the default case and specifies that jobs which have not been printed are not deleted.

6.4.2.1.9 -S or +command-prefix option

```
-S name  
--command-prefix name  
+command-prefix name
```

This option specifies the given name as the prefix for the command file, followed by the job number, to be used by the `-u` option rather than the default of `C` (which in turn may be changed by editing the message file).

6.4.2.1.10 -u or +unqueue option

```
-u  
--unqueue  
+unqueue
```

This option selects the unqueue mode, in which the jobs are copied to the current directory or the directory selected with the `-D` option. Jobs are copied as two files, a job file containing the text of the job and a command file containing an `spr` command. The command file may be run as a shell script to re-queue the job with the same options.

6.4.2.1.11 -y or +delete-unprinted option

```
-y  
--delete-unprinted  
+delete-unprinted
```

This option applies the selected other operations on all jobs, whether printed or not.

6.4.2.1.12 +freeze-current option

```
--freeze-current  
+freeze-current
```

Save all the current options in a `.xitext` file in the current directory with keyword `SQDEL`.

No further action will be taken if this option is specified.

6.4.2.1.13 `+freeze-home` option

```
--freeze-home  
+freeze-home
```

Save all the current options in a `.xitext` file in the user's home directory with keyword `SQDEL`.

No further action will be taken if this option is specified.

6.4.2.2 NOTES

The `-C` option selects the class code for finding the jobs; if you have *override class* privilege, but your default class code does not allow the jobs "to be seen" in the output of `sqlist` then you may need to specify this.

Any errors are reported with appropriate messages on standard error. If any errors are detected operating on several jobs, the exit code will correspond to the last such error displayed.

6.4.3 `sqchange`

```
sqchange [ -options ] job number ...
```

`sqchange` makes changes to one or more **Xi-Text** spooled jobs, which may be on the local host or other hosts on the network. Jobs are specified by the job number, which is reported by `sqlist` in the first column of the default output, or by `spr` or `rspr` when the job is queued with the verbose option. Remote jobs are specified by prefixing the job number by the host name and a colon, thus `host:1234`. Leading zeroes in the job number may be omitted.

Each of the jobs must be owned by the user, or the user must have permission to edit other user's jobs. The user must also have permission to edit remote jobs to operate on remote jobs (this is normally set for all users).

The changes are specified by options, which are as similar as possible to those for `spr`. It is regarded as a mistake not to specify at least one option.

6.4.3.1 Options

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

6.4.3.1.1 **-?** or **+explain** option

```
-?  
--explain  
+explain
```

This option causes a summary of the other options to be displayed without taking further action.

6.4.3.1.2 **-a** or **+mail-attention** option

```
-a  
--mail-attention  
+mail-attention
```

This option turns on the setting requesting that a message be sent in the mail to the user if a job is awaiting attention, either because it reaches the top of the queue and the required form type is currently unavailable or because the form requires alignment.

6.4.3.1.3 **-A** or **+write-attention** option

```
-A  
--write-attention  
+write-attention
```

This option turns on the setting requesting that a message be sent to every terminal at which the user is logged in, or by mail otherwise if the user is not logged in, if a job is awaiting attention, either because it reaches the top of the queue and the required form type is currently unavailable or because the form requires alignment.

6.4.3.1.4 **-b** or **+noatt-message** option

```
-b  
--noatt-message  
+noatt-message
```

This option turns off any job attention messages previously requested.

6.4.3.1.5 -c or +copies option

```
-c n  
--copies n  
+copies n
```

This option sets the number of copies to be printed to *n*, which may possibly be zero.

The maximum number of copies at a time may be restricted to a number for each user such as 10. See the output from [spuser](#) for information on the maximum number of copies.

6.4.3.1.6 -C or +classcode option

```
-C nnnn  
--classcode nnnn  
+classcode nnnn
```

This option may be used to set a class code narrow or expand the range of jobs from which jobs are selected. It is probably only relevant where the user has *override class* privilege and it is required to expand the default class code to select additional jobs which would otherwise not be “found”.

N.B. This does not set the classcode. Use `-S` actually to set the classcode.

6.4.3.1.7 -f or +formtype option

```
-f string  
--formtype string  
+formtype string
```

This option sets the form type of the job or jobs to that given.

Note that not all users may be allowed to select form types other than some default pattern, causing the request to be rejected with an error without changing any jobs.

6.4.3.1.8 -F or +post-proc-flags option

```
-F string  
--post-proc-flags string  
+post-proc-flags string
```

This option changes to the specified string, the data to be passed as the value of the environment variable `SPOOLFLAGS` to any program invoked at the time the job is printed. This might be as a post-processing filter command, or alternatively some shell command invoked as (for example) a `docstart` string.

6.4.3.1.9 -h or +header option

```
-h string  
--header string  
+header string
```

This option changes the title for each job to the specified *string*. The title appears on the banner page, if this is printed.

6.4.3.1.10 -l or +local-only option

```
-l  
--local-only  
+local-only
```

This option, in a networked environment, requests that the job be printed only on the printers local to the machine at which the job is submitted.

If your copy of **Xi-Text** is set as “standalone” this will always be set (and you may have to unset it on existing jobs if you change it to “networked”).

6.4.3.1.11 -L or +network-wide option

```
-L  
--network-wide  
+network-wide
```

This option cancels the request that the job or jobs be printed only on printers local to the machine at which the job is submitted.

Note that this option will have no effect if your copy of **Xi-Text** is set as “standalone”.

6.4.3.1.12 -m or +mail-message option

```
-m  
--mail-message  
+mail-message
```

This option requests a message to be sent in the mail to the user on completion or termination of the job.

6.4.3.1.13 -n or +delay-for option

```
-n nn  
--delay-for nn  
+delay-for nn
```

This option specifies that the job should be held on the queue for at least *nn* minutes from the current time before being printed. The time may alternatively be specified as *hh:mm* or as *hh:mm:ss*, specifying a delay in hours and minutes, or hours, minutes and seconds.

To cancel this option, an argument of zero should be given.

6.4.3.1.14 -N or +delay-until option

```
-N time  
--delay-until time  
+delay-until time
```

This option is an alternative to the *-n* option to specify the earliest time at which the job is to be printed. The argument may be *hh:mm* or *hh:mm:ss* to give the time of day in 24-hour clock notation. If the time has passed, then tomorrow is assumed.

Alternatively a date and a comma may be prefixed to the time in the form *mm/dd* or *dd/mm* depending upon the local convention for date format used. Thus the resulting argument might be

```
-N 10/11,12:30
```

If both this and the *-n* option are given, the most recently specified will apply.

6.4.3.1.15 -O or +odd-even-flags option

```
-O flag  
--odd-even-flags flag  
+odd-even-flags flag
```

This option resets the settings for only odd-numbered pages to be printed, only even-numbered pages to be printed, odd-numbered pages to be printed on odd-numbered copies and even-numbered pages on even-numbered copies, or vice-versa.

The argument can be

- cancel this option
- O print odd numbered pages
- E print even-numbered pages
- A print odd-numbered pages, then even-numbered ones
- B print even-numbered pages, then odd-numbered ones

For example

```
sqchange -c2 -OA
```

will print odd-numbered pages on the first copy and even ones on the second.

For this to work properly, page delimiters must be set appropriately.

6.4.3.1.16 -p or +priority option

```
-p n
--priority n
+priority n
```

This option specifies the priority of the job, between 1 (lowest) and 255 (highest) or some narrower range to which the user is limited unless he/she has *set any priority on queue* privilege.

A user must have the *change priority on queue* privilege to invoke this option.

6.4.3.1.17 -P or +printer option

```
-P name
--printer name
+printer name
```

This option specifies that the job is to be sent to the given printer, as opposed to printing it on the first available printer with the given form type.

The name given may be a pattern to select any printer whose name matches the pattern.

Note that a user may be limited to just one printers or one of a range of printers, see the output of [spuser](#).

To “turn off” a printer name previously specified by a preceding `-Pname` option, put a single `-` sign as the printer name.

6.4.3.1.18 -q or +retain option

```
-q  
--retain  
+retain
```

This option requests that the job or jobs be retained on the queue with copies set to zero after printing, for explicit deletion, or automatically at the expiry of the timeout (as set by the `-t` option).

Some printers may be set up to enforce this on all jobs anyway, see page [70](#).

6.4.3.1.19 -r or +banner option

```
-r  
--banner  
+banner
```

This option resets the option to suppress banner pages using `-s` or when the job or jobs were queued. Note that some form types may be set up never to use banners, or always to use banners, regardless of this option.

6.4.3.1.20 -R or +page-range option

```
-R m-n  
--page-range m-n  
+page-range m-n
```

This option specifies that pages `m` through to `n` inclusive are to be printed. This does of course assume that the job has recognisable pages. If `m` or `n` are omitted, then “the beginning” or “the end” respectively is assumed, so `-R 3-5` would print pages 3 to 5, `-7` would print pages 1 to 7 inclusive, and `-R 4-` would print page 4 to the end inclusive. `-R 1-` would turn this option off by selecting 1 to the end.

6.4.3.1.21 -s or +no-banner option

```
-s  
--no-banner  
+no-banner
```

This option suppresses any banner page (large letter user name etc) which is printed before the job or jobs. Note that some form types may be set up always or never to print banners regardless of this option.

6.4.3.1.22 -S or +set-classcode option

```
-S nnnn  
--set-classcode nnnn  
+set-classcode nnnn
```

This option resets the classcode of the job of jobs to that given (as opposed to viewing the jobs which is done with the `-C` option).

The argument supplied should be some collection of the letters `A` through to `P` inclusive (upper or lower case) with `-` to indicate ranges.

Users who have *override class* privilege as displayed by `spuser` will set the job or jobs to exactly that classcode given.

Other users will have the supplied classcode masked down by their default classcode before this is applied. If the masking results in no bits being set in the classcode, the command is aborted with an error.

6.4.3.1.23 -t or +printed-timeout option

```
-t n  
--printed-timeout n  
+printed-timeout n
```

This option resets the timeout whereby if a job is held on the queue after being printed it will be automatically deleted after the given number of hours, which may be 1 to 32767 (nearly 4 years).

6.4.3.1.24 -T or +not-printed-timeout option

```
-T n  
--not-printed-timeout n  
+not-printed-timeout n
```

This option resets the timeout whereby if held on the queue *without* being printed, the job or jobs will be deleted automatically, to between 1 hour and 32767 hours (nearly 4 years).

6.4.3.1.25 -u or +post-user option

```
-u name  
--post-user name  
+post-user name
```

This option requests that the specified user name be substituted for the submitting user on the banner page optionally printed at the start of the job. The job still remains the responsibility of the submitting user.

To “turn off” a user name specified in a previous `-u` option, put a single `-` sign as the user name. The user name must exist on the machine to which the job is queued, but this will be ignored if not.

6.4.3.1.26 `-w` or `+write-message` option

```
-w  
--write-message  
+write-message
```

This option requests a message to be sent to every terminal at which the user is logged in, or by mail otherwise if the user is not logged in, when the job is completed or terminated.

6.4.3.1.27 `-x` or `+no-message` option

```
-x  
--no-message  
+no-message
```

This option turns off any job completion messages previously requested with `-m` or `-w`.

6.4.3.1.28 `-z` or `+no-retain` option

```
-z  
--no-retain  
+no-retain
```

This option cancels a request that the job or jobs be retained on the queue after printing.

Note that if the printer is set to always retain jobs after printing, this will have no effect.

6.4.3.1.29 `+freeze-current` option

```
--freeze-current  
+freeze-current
```

Save all the current options in a `.xitext` file in the current directory with keyword `SQCHANGE`.

No further action will be taken if this option is specified.

6.4.3.1.30 +freeze-home option

```
--freeze-home
+freeze-home
```

Save all the current options in a `.xitext` file in the user's home directory with keyword `SQCHANGE`.

No further action will be taken if this option is specified.

6.4.3.2 Notes

The `-C` option selects the class code for finding the jobs; if you have override class privilege, but your default class code does not allow the jobs “to be seen” in the output of `sqlist`, then you may need to specify this. The `-S` option sets a new classcode, which may be “anded” with your default class if you do not have *override class* privilege.

Any errors are reported with appropriate messages on standard error. If any errors are detected operating on several jobs, the exit code will correspond to the last such error displayed.

6.4.3.3 Examples

To change the number of copies of job `24139` to 3

```
sqchange -c3 14139
```

To changed the formtype to `a4.ps` for 3 jobs including one on a remote machine, `avon`

```
sqchange -f a4.ps 24139 26921 avon:21941
```

Changing all the jobs associated with printer `laser` to come out on printer `ljet` may be done with a shell script like this:

```
sqlist -F "%N %P" | grep laser | while read JOB REST
do
    sqchange -P ljet $JOB
done
```

Or you can do it in one operation with this ‘one-liner’:

```
sqchange -Pljet `sqlist -F"%N %P"|grep laser|sed 's/\(.*\) .*/\1/'`
```

6.4.4 Lpstatc

```
lpstatc [ -options ]
cancel jobnumber [ ... ]
```

`Lpstat` is a program installed in place of `lpstat` to emulate it as far as is practicable, generating a report of outstanding print jobs from the print queue in a format similar to the original `lpstat`. This is installed when `lpcover` is installed.

`cancel` is a shell script which invokes `sqdel` on its arguments in a similar fashion as `cancel` does with `lp`. This is installed when `lpcover` is installed.

If options are supplied to `lpstat`, a warning is given that these are not implemented. `Cancel` does not take any options.

6.5 Managing printers (command line)

6.5.1 splist

```
splist [ -options ] [ printer ... ]
```

`splist` generates a list of the printers, with their current status, on the **Xi-Text** printer lists, including those on remote machines, on standard output. Each line of the output corresponds to a single printer, and by default the output is generally similar to the default format of the printers screen of the `spq` command.

Various options allow the user to control the output in various ways as described below.

The output can be restricted to one or more named printers by listing them on the command line.

The default output of `splist` looks like the following

```
ljet      lp      a4      idle
epson     tty15  invoice halted
blue:epson lp      invoice idle
```

Printer names for printers on remote machines are prefixed as shown by the machine name and a colon. The output is displayed in the minimum width to let each column line up as shown, with fields separated by spaces.

The output format and types of information displayed can be redefined using the `-F` option.

6.5.1.1 Options

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

6.5.1.1.1 -? or +explain option

```
-?
--explain
+explain
```

This option causes a summary of the other options to be displayed without taking further action.

6.5.1.1.2 -C or +classcode option

```
-C nnnn  
--classcode nnnn  
+classcode nnnn
```

This option restricts attention to jobs matching (i.e. having at least one bit in common with) the classcode given.

Unless he/she has *override class* privilege, the class code given will be masked “anded” with the user’s class code and the result (which must not be zero, otherwise an error message will be given) used.

6.5.1.1.3 -D or +default-format option

```
-D  
--default-format  
+default-format
```

This option cancels a `-F` option and restores the default format.

6.5.1.1.4 -F or +format option

```
-F string  
--format string  
+format string
```

This option specifies a user-defined output format. The string argument contains characters to be printed and symbols introduced by % indicating the required attributes of the jobs to be displayed.

The format symbols are listed on page [153](#).

6.5.1.1.5 -H or +header option

```
-H  
--header  
+header
```

This option causes a header to be generated for each column of the output according the field displayed there.

The headers are taken from the message file, by default `/usr/spool/progs/rest.help`, which may be adjusted as required.

6.5.1.1.6 -l or +local-only option

```
-l  
--local-only  
+local-only
```

This option indicates to display only printers local to the current host.

6.5.1.1.7 -N or +no-header option

```
-N  
--no-header  
+no-header
```

This option turns off the header requested by a previously-requested `-H` option.

6.5.1.1.8 -q or +list-only option

```
-q name  
--list-only name  
+list-only name
```

List only printers whose name matches the name given.

Name may be a pattern with wild cards matched in a similar fashion to the shell, or a list of comma-separated alternatives.

```
-q 'a*'
```

displays printers whose names start with `a`.

```
-q '[p-t]*,*[!h-m]'
```

displays printers whose names start with `p` to `t` or end with anything other than `h` to `m`. To cancel this argument, give a single `-` sign as name.

6.5.1.1.9 -r or +remotes option

```
-r  
--remotes  
+remotes
```

This option sets to display printers local to the current host and connected hosts, cancelling a preceding `-l` option.

6.5.1.1.10 -S or +sort-ptrs option

```
-S  
--sort-ptrs  
+sort-ptrs
```

This option requests that the printer list be sorted into order by host identifier and name.

6.5.1.1.11 -U or +no-sort-ptrs option

```
-U  
--no-sort-ptrs  
+no-sort-ptrs
```

This option restores the display of printers to be in the order encountered, which is the default.

6.5.1.1.12 +freeze-current option

```
--freeze-current  
+freeze-current
```

Save all the current options in a `.xitext` file in the current directory with keyword `SPLIST`.

No further action will be taken if this option is specified.

6.5.1.1.13 +freeze-home option

```
--freeze-home  
+freeze-home
```

Save all the current options in a `.xitext` file in the user's home directory with keyword `SPLIST`.

No further action will be taken if this option is specified.

6.5.1.2 Format codes

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

Each column is padded on the right to the length of the longest entry.

If a header is requested, the appropriate abbreviation is obtained from the message file and inserted.

`%%` insert a single `%` sign
`%a` display `AB!!` if printer has received an interrupt (abort print or interrupt print) message.
`%c` display class code.
`%d` display device name, with angle brackets for network devices.
`%e` display comment field.
`%f` form type
`%h` display `heoj` if printer marked to halt at end of current job.
`%i` process id of `spd` process.
`%j` job number being printed (preceded by host name if not a local job).
`%l` display `localonly` if printer is marked for local jobs only.
`%m` feedback message on standard error from network filter process.
`%n` N/A if printer is marked in need of running alignment routine.
`%p` printer name.
`%s` printer state, followed by a colon and the feedback message if applicable.
`%t` printer state only, no feedback message.
`%u` user name of owner of printing job.
`%y` minimum job size if applicable.
`%z` maximum job size if applicable.

Note that the various strings such as `localonly` etc are read from the message file also, so it is possible to modify them as required by the user.

The default format is

```
%p %d %f %s %j %u
```

6.5.2 spstart

```
spstart [ -options ] printer [ formtype ]
```

`spstart` may also be used to start printers, or to halt, change the form type and to restart in one operation.

`spstart` may be used to start the **Xi-Text** scheduler, allowing space for the given number of jobs and printers in shared memory – see page [125](#).

```
spstart lj4
```

Will start the printer `lj4` if it is not running with whatever formtype is currently set.

If it is idle, then it will have no effect.

If it is printing, but marked as to halt at the end of the current job, this indication will be turned off.

```
spstart lj4 a4.ps
```

This will have the same effect as `spstart` without the form type argument if the paper type, i.e. the form type without the suffix, in this case `a4`, is the same.

If the paper type is other than `a4` and is running, then it will halt it at the end of the current job if applicable, wait for it to halt, then change the formtype to `a4.ps` and restart it.

Only “paper types”, ignoring the suffix, are compared in deciding whether the form type is loaded.

Shell-style wild cards (which should probably be protected from incorrect expansion by shell using quotes) may be used, in conjunction with the `-f` option, to start multiple printers, thus:

```
spstart -f 'l*'
spstart -f '*[!a-m]'
spstart -f 'l*,*[!a-m]'
```

will respectively start all printers whose names begin with `l`, all printers whose names do not end with `a` through to `m` and both those classes of printers. Many installations include

```
spstart -f '*'
```

in the “system startup script” to start up all local printers.

Most of the options (e.g. `-l` and `-C`) serve only to clarify which printer or printers are being referred to where there is ambiguity or the user wishes to restrict the field.

Names of printers starting with digits will not cause any confusion if `spsched` is running.

Printers on remote machines may be started by prefixing the machine name, thus

```
spstart -f host1:l*
```

Printers are not started on more than one machine at once. You should start the printers as separate commands:

```
spstart -f '*'
spstart -f 'host1:*'
spstart -f 'host2:*
```

To run `spstart`, the user must have *halt and restart printers* permission.

6.5.2.1 Options

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

6.5.2.1.1 `-?` or `+explain` option

```
-?
--explain
+explain
```

This option causes a summary of the other options to be displayed without taking further action.

6.5.2.1.2 -l or +device-name option

```
-l device  
--device-name device  
+device-name device
```

This option specifies by giving the device name or network address which of several similarly-named printers is intended.

6.5.2.1.3 -C or +classcode option

```
-C nnnn  
--classcode nnnn  
+classcode nnnn
```

This option specifies via the class code which of several similarly-named printers is intended.

6.5.2.1.4 -f or +force-all option

```
-f  
--force-all  
+force-all
```

This option indicates that if the printer name argument selects more than one printer, then all such printers are intended and the command applied to them.

6.5.2.1.5 -n or +no-force option

```
-n  
--no-force  
+no-force
```

This option cancels the `-f` option.

6.5.2.1.6 -W or +wait-complete option

```
-W  
--wait-complete  
+wait-complete
```

This option causes `spstart` to wait for operations to complete before exiting.

6.5.2.1.7 -E or +no-wait option

```
-E
--no-wait
+no-wait
```

Do not wait for operations to complete before exiting.

6.5.2.1.8 +freeze-current option

```
--freeze-current
+freeze-current
```

Save all the current options in a `.xitext` file in the current directory with keyword `SPSTART`.

No further action will be taken if this option is specified.

6.5.2.1.9 +freeze-home option

```
--freeze-home
+freeze-home
```

Save all the current options in a `.xitext` file in the user's home directory with keyword `SPSTART`.

No further action will be taken if this option is specified.

6.5.3 sphalt and spstop

```
sphalt [-options] printer
spstop [-options] printer
```

`sphalt` may be used for halting one or more printers at the end of the current job, or immediately if they are not printing. `spstop` may be used for halting one or more printers immediately.

```
sphalt lj4
```

will halt printer `lj4` at the end of the current job if it is running and will otherwise have no effect.

```
sphalt -f 'l*'
```

will likewise halt all printers whose names start with `l` etc.

Most of the options (e.g. `-l` and `-C`) serve only to clarify which printer or printers are being referred to where there is ambiguity or the user wishes to restrict the field.

Printers on remote machines may be halted by prefixing the machine name, thus

```
spstop -f host1:l*
```

To run these commands, the user must have *halt and restart printers* permission.

6.5.3.1 Options

The environment variable on which options are supplied is `SPHALT` for `sphalt` and `SPSTOP` for `spstop`. The environment variable to specify the help file is `SPRESTCONF`.

6.5.3.1.1 -? or +explain option

```
-?  
--explain  
+explain
```

This option causes a summary of the other options to be displayed without taking further action.

6.5.3.1.2 -l or +device-name option

```
-l device  
--device-name device  
+device-name device
```

Specify by giving the device name or network address which of several similarly-named printers is intended.

6.5.3.1.3 -C or +classcode option

```
-C nnnn  
--classcode nnnn  
+classcode nnnn
```

Specify via the class code which of several similarly-named printers is intended.

6.5.3.1.4 -f or +force-all option

```
-f  
--force-all  
+force-all
```

Indicate that if the printer name argument selects more than one printer, then all such printers are intended.

6.5.3.1.5 -n or +no-force option

```
-n  
--no-force  
+no-force
```

Cancel the `-f` option.

6.5.3.1.6 -W or +wait-complete option

```
-W  
--wait-complete  
+wait-complete
```

Wait for operations to complete before exiting.

6.5.3.1.7 -E or +no-wait option

```
-E  
--no-wait  
+no-wait
```

Do not wait for operations to complete before exiting.

6.5.3.1.8 +freeze-current option

```
--freeze-current  
+freeze-current
```

Save all the current options in a `.xitext` file in the current directory with keyword `SPHALT` or `SPSTOP`.

No further action will be taken if this option is specified.

6.5.3.1.9 +freeze-home option

```
--freeze-home  
+freeze-home
```

Save all the current options in a `.xitext` file in the user's home directory with keyword `SPHALT` or `SPSTOP`.

No further action will be taken if this option is specified.

6.5.4 spchange

```
spchange [-options] printer [ formtype ]
```

`spchange` with appropriate options and a printer name (with optional form type) changes the specified parameters of a printer, which should be halted (and is left halted).

To run `spchange`, the user must have *add and delete printers* permission.

6.5.4.1 Options

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

6.5.4.1.1 -? or +explain option

```
-?  
--explain  
+explain
```

This option causes a summary of the other options to be displayed without taking further action.

6.5.4.1.2 -N or +network-device option

```
-N  
--network-device  
+network-device
```

Indicate that the argument to `-l` refers to a network device.

6.5.4.1.3 -L or +line-device option

```
-L  
--line-device  
+line-device
```

Indicate that the argument to `-l` refers to a physical device (under `/dev`).

6.5.4.1.4 -s or +local-only option

```
-s  
--local-only  
+local-only
```

Mark printer as being local only to the host and not visible from other connected servers.

6.5.4.1.5 -w or +network-wide option

```
-w  
--network-wide  
+network-wide
```

Mark printer as being available to other machines.

6.5.4.1.6 -l or +device-name option

```
-l device  
--device-name device  
+device-name device
```

Specify by giving the device name or network address which of several similarly-named printers is intended.

6.5.4.1.7 -C or +classcode option

```
-C nnnn  
--classcode nnnn  
+classcode nnnn
```

Specify via the (current) class code which of several similarly-named printers is intended.

6.5.4.1.8 -D or +description option

```
-D string  
--description string  
+description string
```

Set the description or comment field associated with the printer as specified.

6.5.4.1.9 -v or +new-device option

```
-v device  
--new-device device  
+new-device device
```

Reset the device name as specified.

6.5.4.1.10 -S or +set-classcode option

```
-S nnnn  
--set-classcode nnnn  
+set-classcode nnnn
```

Reset the class code as specified.

Note this *changes* the classcode and the `-C` option helps select the printer.

The specified class code will be restricted to a subset of the user's class code unless the user has *override class* privilege. It is an error if this process yields an all zero class code.

6.5.4.1.11 -f or +force-all option

```
-f  
--force-all  
+force-all
```

Indicate that if the printer name argument selects more than one printer, then all such printers are intended.

6.5.4.1.12 -n or +no-force option

```
-n  
--no-force  
+no-force
```

Cancel the `-f` option.

6.5.4.1.13 -W or +wait-complete option

```
-W  
--wait-complete  
+wait-complete
```

Wait for operations to complete before exiting.

6.5.4.1.14 -E or +no-wait option

```
-E
--no-wait
+no-wait
```

Do not wait for operations to complete before exiting.

6.5.4.1.15 +freeze-current option

```
--freeze-current
+freeze-current
```

Save all the current options in a `.xitext` file in the current directory with keyword `SPCHANGE`.

No further action will be taken if this option is specified.

6.5.4.1.16 +freeze-home option

```
--freeze-home
+freeze-home
```

Save all the current options in a `.xitext` file in the user's home directory with keyword `SPCHANGE`.

No further action will be taken if this option is specified.

6.5.5 spadd and spdel

```
spadd [-options] printer formtype
spdel [-options] printer
```

`spadd` may be used to add a printer and `spdel` may be used to delete a printer. Only one printer at a time may be added or deleted.

```
spadd -l tty15 lj4b a4
```

adds (or “installs”) a new printer with the specified parameters. The `-l` option is mandatory to denote the device name (or network address) as is the initial form type, in the above example `a4`.

```
spadd -l 193.112.238.95 -N ptr9 a4
```

adds a network (terminal server) printer with the IP address given.

Please note that this does nothing to create an initial setup file, which should be created before it is set running.

The `Ptrinstall` script takes care of all these options for you.

```
spdel lj4b
```

Deletes (or “de-installs”) the specified printer.

Printers on remote machines may not be added or deleted. This should be done from the relevant machine. Unless the printer is or was marked as local only, the printer will imappear or disappear from the display

To run `spadd` or `spdel` the user must have *add and delete printers* permission.

6.5.5.1 Options

The environment variable on which options are supplied is `SPADD` for `spadd` and `SPDEL` for `spdel`. The environment variable to specify the help file is `SPRESTCONF`.

6.5.5.1.1 -? or +explain option

```
-?
--explain
+explain
```

This option causes a summary of the other options to be displayed without taking further action.

6.5.5.1.2 -N or +network-device option

```
-N
--network-device
+network-device
```

Indicate that the argument to `-l` refers to a network device.

6.5.5.1.3 -L or +line-device option

```
-L
--line-device
+line-device
```

Indicate that the argument to `-l` refers to a physical device (under `/dev`).

6.5.5.1.4 -s or +local-only option

```
-s  
--local-only  
+local-only
```

Mark printer as being local only to the host and not visible from other connected servers.

6.5.5.1.5 -w or +network-wide option

```
-w  
--network-wide  
+network-wide
```

Mark printer as being available to other machines.

6.5.5.1.6 -l or +device-name option

```
-l device  
--device-name device  
+device-name device
```

in the case of `spadd`, this defines the device name or network address of the printer. For `spdel`, it may be used to specify which of several similarly-named printers is intended.

6.5.5.1.7 -C or +classcode option

```
-C nnnn  
--classcode nnnn  
+classcode nnnn
```

in the case of `spadd`, it defines the class code (subject to modification by the user's class code) which will be given to the printer.

For `spdel` it may be used to specify which of several similarly-named printers is intended.

6.5.5.1.8 -D or +description option

```
-D string  
--description string  
+description string
```

Set the description or comment field associated with the printer as specified.

6.5.5.1.9 -W or +wait-complete option

```
-W  
--wait-complete  
+wait-complete
```

Wait for operations to complete before exiting.

6.5.5.1.10 -E or +no-wait option

```
-E  
--no-wait  
+no-wait
```

Do not wait for operations to complete before exiting.

6.5.5.1.11 +freeze-current option

```
--freeze-current  
+freeze-current
```

Save all the current options in a `.xitext` file in the current directory with keyword `SPADD` or `SPDEL`.

No further action will be taken if this option is specified.

6.5.5.1.12 +freeze-home option

```
--freeze-home  
+freeze-home
```

Save all the current options in a `.xitext` file in the user's home directory with keyword `SPADD` or `SPDEL`.

No further action will be taken if this option is specified.

6.5.6 spinter

```
spinter [ -options ] printer
```

`spinter` interrupts the job on the specified printer. This may be to halt printing of that job, allow a higher-priority job to print and then resume printing of the interrupted job at the interrupted page. Note that this will only work as intended if page delimiters are correctly defined.

```
spinter lj4
```

interrupts the given printer (or printers).

Printers on remote machines may be interrupted by prefixing the machine name, thus

```
spinter host1:lj4
```

To run this, the user must have *halt and restart printers* permission.

6.5.6.1 Options

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

6.5.6.1.1 -? or +explain option

```
-?
--explain
+explain
```

This option causes a summary of the other options to be displayed without taking further action.

6.5.6.1.2 -l or +device-name option

```
-l device
--device-name device
+device-name device
```

Specify by giving the device name or network address which of several similarly-named printers is intended.

6.5.6.1.3 -C or +classcode option

```
-C nnnn
--classcode nnnn
+classcode nnnn
```

Specify via the (current) class code which of several similarly-named printers is intended.

6.5.6.1.4 -f or +force-all option

```
-f
--force-all
+force-all
```

Indicate that if the printer name argument selects more than one printer, then all such printers are intended.

6.5.6.1.5 -n or +no-force option

```
-n
--no-force
+no-force
```

Cancel the `-f` option.

6.5.6.1.6 -W or +wait-complete option

```
-W
--wait-complete
+wait-complete
```

Wait for operations to complete before exiting.

6.5.6.1.7 -E or +no-wait option

```
-E
--no-wait
+no-wait
```

Do not wait for operations to complete before exiting.

6.5.6.1.8 +freeze-current option

```
--freeze-current
+freeze-current
```

Save all the current options in a `.xitext` file in the current directory with keyword `SPINTER`.

No further action will be taken if this option is specified.

6.5.6.1.9 +freeze-home option

```
--freeze-home  
+freeze-home
```

Save all the current options in a `.xitext` file in the user's home directory with keyword `SPINTER`.

No further action will be taken if this option is specified.

6.5.7 spok and spnok

```
spok [ -options ] printer  
spnok [ -options ] printer
```

If a printer is in, or potentially in, “awaiting operator” state, then `spok` and `spnok` may be used to communicate with it.

A printer may be set up to require an alignment routine (see page 67).

If it does, then the first print job which it attempts to print will cause it to enter the state “awaiting operator”, which is usually signified by `a/w oper`. It will print out the alignment page and stop.

The printer can then either be adjusted and the alignment routine rerun, by running `spnok`, or the user can indicate that the printer is satisfactorily aligned by running `spok`, whereupon printing continues normally, until the printer is halted.

If the printer has just been started and is indicating that it is idle, but you are aware that the alignment is correct, you can bypass the alignment routine by running `spok`.

If the printer is idle, but you want to re-run the alignment routine before the next job, you can do this by running `spnok`.

Both commands are harmless if they are inapplicable to the printer or the printer is already in the state being set.

A second use for these commands is if the printer is set up to be in single-job mode, (see page 69). In such a case the printer will revert to “awaiting operator” state at the start of each job and it can be set printing by running either `spok` or `spnok`, only the former should be used if it has alignment pages as well.

To run `spok` and `spnok`, the user must have `select printer list` permission.

6.5.7.1 Options

The environment variable on which options are supplied is `SPOK` for `spok` and `SPNOK` for `spnok`. The environment variable to specify the help file is `SPRESTCONF`.

6.5.7.1.1 **-?** or **+explain** option

```
-?  
--explain  
+explain
```

This option causes a summary of the other options to be displayed without taking further action.

6.5.7.1.2 **-l** or **+device-name** option

```
-l device  
--device-name device  
+device-name device
```

Specify by giving the device name or network address which of several similarly-named printers is intended.

6.5.7.1.3 **-C** or **+classcode** option

```
-C nnnn  
--classcode nnnn  
+classcode nnnn
```

Specify via the class code which of several similarly-named printers is intended.

6.5.7.1.4 **-f** or **+force-all** option

```
-f  
--force-all  
+force-all
```

Indicate that if the printer name argument selects more than one printer, then all such printers are intended.

6.5.7.1.5 **-n** or **+no-force** option

```
-n  
--no-force  
+no-force
```

Cancel the `-f` option.

6.5.7.1.6 -W or +wait-complete option

```
-W  
--wait-complete  
+wait-complete
```

Wait for operations to complete before exiting.

6.5.7.1.7 -E or +no-wait option

```
-E  
--no-wait  
+no-wait
```

Do not wait for operations to complete before exiting.

6.5.7.1.8 +freeze-current option

```
--freeze-current  
+freeze-current
```

Save all the current options in a `.xitext` file in the current directory with keyword `SPOK` or `SPNOK`.

No further action will be taken if this option is specified.

6.5.7.1.9 +freeze-home option

```
--freeze-home  
+freeze-home
```

Save all the current options in a `.xitext` file in the user's home directory with keyword `SPOK` or `SPNOK`.

No further action will be taken if this option is specified.

6.5.8 spstat

```
spstat [ -options ] printer state
```

`spstat` may be used in shell scripts to test the state of a printer.

```
spstat lj4
```

will return an exit code of 0 (TRUE to shells) if the printer is running, (i.e. printing or idle) and otherwise 1 (FALSE to shells). Some other exit code will be returned, together with a message, if there is an error, e.g. non-existent printer.

To test for specific states, use the state names as listed by `splist`, (in fact it takes the state names from the same place in the same file `/usr/spool/progs/rest.help` by default) for example.

```
if spstat lj4 error || spstat lj4 offline
then    echo trouble with lj4
fi
```

6.5.8.1 Standard state names for spstat

The following state names are used by `spstat`, which are usually shared with `splist` in the message file `/usr/spool/progs/rest.help` by default. As with other **Xi-Text** programs, the message file may be edited as required:

```
offline  printer has gone off-line or timed out
error    an error has been raised by the printer, driver or Xi-Text
halted   printer is shut down and printing suspended by Xi-Text
startup  being initialised to a ready/printing state from the halted state
idle     ready to print next job but nothing to print
shutdown in process of being taken down to the halted state
printing a job is currently being printed
a/w oper awaiting operator attention
```

6.5.8.2 Options

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

Note that only one printer name may be operated upon at any one time.

6.5.8.2.1 -? or +explain option

```
-?
--explain
+explain
```

This option causes a summary of the other options to be displayed without taking further action.

6.5.8.2.2 -l or +device-name option

```
-l device  
--device-name device  
+device-name device
```

Specify by giving the device name or network address which of several similarly-named printers is intended.

6.5.8.2.3 -C or +classcode option

```
-C nnnn  
--classcode nnnn  
+classcode nnnn
```

Specify via the class code which of several similarly-named printers is intended.

6.5.8.2.4 +freeze-current option

```
--freeze-current  
+freeze-current
```

Save all the current options in a `.xitext` file in the current directory with keyword `SPSTAT`.

No further action will be taken if this option is specified.

6.5.8.2.5 +freeze-home option

```
--freeze-home  
+freeze-home
```

Save all the current options in a `.xitext` file in the user's home directory with keyword `SPSTAT`.

No further action will be taken if this option is specified.

6.5.9 Ptrinstall

```
Ptrinstall
```

`Ptrinstall` is an interactive Perl script which can create some printer definitions, including `.device` and `default` files, install printers with appropriate options, edit existing files and make adjustments.

This is intended as a “starter” printer setup file creator. It does not know about more esoteric options such as `align` keywords or similar.

It should be sufficient to get **Xi-Text** running with many common types of printer and interface.

For more sophisticated operations and facilities you may need to hand-edit the `.device` or `default` files yourself, or create additional form type files.

If you do this, please remember to change the description at the start of the `.device` and `default` to `filenameCustom`, so that `Ptinstall` will not attempt to make changes in ignorance of the functions of these files.

There are no options to `Ptinstall`. The user should answer the questions posed appropriately. In most cases pressing `?` or `F1` will give help.

6.5.10 `netwcmd`

```
netwcmd[-n] printer [form]
```

`netwcmd` can be used to extract the network (terminal server) printing command for a given printer and output it on standard output. This is often useful for testing terminal server commands from the command prompt in the event that error conditions arise and enquiring what a given command “would have been”.

The current form type is used unless the form argument is given, but this generally should not make any difference.

A shell script is output to standard output containing the command. By default, all the `$SPOOLDEV` etc printer environment variables are expanded in the output where appropriate as the printer “would see it”, however this may be suppressed by means of the `-n` option.

There is a very subtle difference if the network keyword is given with

```
network exec ....
```

rather than `network` without `exec` or via `network=`. In the first case, just the command is given, whereas in the other two cases the output is preceded with

```
#!/bin/sh
```

and a blank line.

Comments are added if no `reopen` statement appears. An error will be generated if the printer is directly connected.

6.5.11 `getsnmp`

```
getsnmp [options] variable
```

`getsnmp` can be used to obtain the value of an SNMP variable from a printer supporting the SNMP protocol.

In many cases the command

```
getsnmp -h hostname -N 1.3
```

will display the printer type on the *hostname* given.

This is used by the [Ptrinstall](#) script but is left for general use.

6.5.11.1 Options

The following options are used by `getsnmp`. There is no option to “remember” options as with other **Xi-Text** user-level programs.

6.5.11.1.1 -h option

```
-h host
```

Set the host name (or IP address) to which requests are sent. This must always be specified.

6.5.11.1.2 -H option

```
-H myhost
```

Set the outgoing host name (or IP address) from which requests go in the case where this may be wrong, as with multiple outgoing IP addresses.

6.5.11.1.3 -p option

```
-p port
```

Set the port number to be used, if not `snmp` (161).

6.5.11.1.4 -C option

```
-C community
```

Set the SNMP community to be used, if not `public`.

6.5.11.1.5 -d option

```
-d n
```

Set the debugging level. Increasing values of the argument will debug ever more closely with more voluminous output.

6.5.11.1.6 -T option

`-T n`

Specify the timeout for UDP requests, which may be fractional.

6.5.11.1.7 -f option

`-f file`

Specify the configuration file to be used (for SNMP macros, giving symbolic names to SNMP object IDs).

The default is `xtsnmpdef` in the programs directory for **Xi-Text**, by default `/usr/spool/progs`, as used by `xtlhp` (see page 277).

6.5.11.1.8 -N option

`-N`

If the *variable* requested cannot be found, get the next available one.

6.5.11.1.9 -I option

`-I`

Display the object ID of the variable extracted.

6.6 Interactive job and printer administration

6.6.1 spq

`spq [-options]`

`spq` is a screen-oriented program which displays in real time the current state of the **Xi-Text** spool queue and printer list, refreshing the screen automatically as the queue changes or printer state changes, and allowing the status of jobs on the queue to be altered according to each user's permissions and privileges.

Please see Chapter 7 for more details of the interactive commands. This section focuses on the command-line options which may be used to control the initial display.

6.6.1.1 Options

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

Certain commands available on-screen enable many of these options to be changed and saved in configuration files. (There is no “freeze” option because of this).

6.6.1.1.1 `-?` or `+explain` option

```
-?  
--explain  
+explain
```

This option causes a summary of the other options to be displayed without taking further action.

6.6.1.1.2 `-A` or `+no-confirm-abort` option

```
-A  
--no-confirm-abort  
+no-confirm-abort
```

This option indicates that the user does not require confirmation of any jobs being aborted or deleted.

6.6.1.1.3 `-a` or `+confirm-abort` option

```
-a  
--confirm-abort  
+confirm-abort
```

This option indicates that the user should confirm that a job should be aborted or deleted in all cases.

There are actually 3 possibilities.

1. No confirmation at all, this is selected with the `-A` option.
2. Confirmation of jobs which have not been printed. This is the default state, or if a `-A` option is followed by a single `-a` option.
3. Confirmation in all cases, if `-a` option is applied in the default state or after a previous `-a` option.

6.6.1.1.4 -B or +no-help-box option

```
-B  
--no-help-box  
+no-help-box
```

Put help messages in inverse video rather than in a box (this is the default).

6.6.1.1.5 -b or +help-box option

```
-b  
--help-box  
+help-box
```

Put help messages in a box rather than displaying inverse video.

6.6.1.1.6 -C or +classcode option

```
-C nnnn  
--classcode nnnn  
+classcode nnnn
```

List only jobs and printers which match (have at least one bit in common with) the specified classcode (which may be reduced further by the user's classcode).

6.6.1.1.7 -D or +printed-jobs option

```
-D  
--printed-jobs  
+printed-jobs
```

Restrict the jobs display to jobs which have been printed.

6.6.1.1.8 -d or +dont-care option

```
-d  
--dont-care  
+dont-care
```

Cancel `-j` or `-p` options; the cursor will be in the jobs section of the screen if there are any jobs to display and otherwise in the printers section, unless the user does not have access to the printer list.

6.6.1.1.9 -E or +page-counts option

```
-E  
--page-counts  
+page-counts
```

Try to display job page counts where applicable.

6.6.1.1.10 -e or +no-page-counts option

```
-e  
--no-page-counts  
+no-page-counts
```

Do not try to display page counts in jobs, just the size in bytes.

6.6.1.1.11 -H or +keep-char-help option

```
-H  
--keep-char-help  
+keep-char-help
```

When displaying a help screen, interpret the next key press as a command as well as clearing the help screen. This is the default.

6.6.1.1.12 -h or +lose-char-help option

```
-h  
--lose-char-help  
+lose-char-help
```

Discard whatever key press is made to clear a help screen.

6.6.1.1.13 -I or +include-all option

```
-I  
--include-all  
+include-all
```

Display all jobs regardless of printer (but still, if appropriate, restrict the display of printers).

6.6.1.1.14 -j or +jobs-screen option

```
-j  
--jobs-screen  
+jobs-screen
```

Commence with cursor in jobs section of the screen.

6.6.1.1.15 -l or +local-only option

```
-l  
--local-only  
+local-only
```

Display only jobs or printers local to the host on which `spq` is invoked.

6.6.1.1.16 -M or +no-error-box option

```
-M  
--no-error-box  
+no-error-box
```

Put error messages in inverse video rather than in a box.

6.6.1.1.17 -m or +error-box option

```
-m  
--error-box  
+error-box
```

Put error messages in a box rather than displaying inverse video.

6.6.1.1.18 -n or +follow-job option

```
-n  
--follow-job  
+follow-job
```

If a job or printer moves on the screen, try to follow it.

6.6.1.1.19 -P or +number-printers option

```
-P n
--number-printers n
+number-printers n
```

Set the number of lines on the printer section of the screen.

6.6.1.1.20 -p or +printer-screen option

```
-p
--printer-screen
+printer-screen
```

Commence with cursor in printers section of the screen (if the user is allowed to select it).

6.6.1.1.21 -q or +list-only option

```
-q name
--list-only name
+list-only name
```

List only jobs whose destination includes, or printers whose name matches the name given, which may be a pattern with wild cards matched in a similar fashion to the shell.

For example

```
-q 'a*'
```

displays jobs with printer names and printers whose names start with *a*.

```
-q '[p-t]*,*[!h-m]'
```

displays jobs and printer whose names start with *p* to *t* or end with anything other than *h* to *m*. To cancel this argument, give a single *-* sign as the pattern.

6.6.1.1.22 -R or +refresh-time option

```
-R n
--refresh-time n
+refresh-time n
```

sets the screen refresh time. This only affects minor changes such as the current page count. Addition or deletion of jobs etc count as major changes and will cause an immediate screen refresh.

6.6.1.1.23 -r or +remotes option

```
-r  
--remotes  
+remotes
```

display jobs or printers on all attached hosts (subject to other restrictions).

6.6.1.1.24 -S or +sort-ptrs option

```
-S  
--sort-ptrs  
+sort-ptrs
```

sort printers into order by host identifier and name.

6.6.1.1.25 -s or +keep-cursor option

```
-s  
--keep-cursor  
+keep-cursor
```

If a job or printer moves on the screen, try to preserve the relative position of the cursor on the screen rather than following the job or printer.

6.6.1.1.26 -t or +just-title option

```
-t string  
--just-title string  
+just-title string
```

Restrict the display to jobs whose title matches the pattern specified.

The pattern may be a pattern with shell-like wild cards as in the description of the `-q` parameter above.

6.6.1.1.27 -u or +no-sort-ptrs option

```
-u  
--no-sort-ptrs  
+no-sort-ptrs
```

Display printers in the order encountered (default).

6.6.1.1.28 -u or +just-user option

```
-u user  
--just-user user  
+just-user user
```

Restrict the display to jobs owned by the user specified. The user name may be a pattern with shell-like wild cards as in the description of the `-q` parameter.

6.6.1.1.29 -V or +all-jobs option

```
-V  
--all-jobs  
+all-jobs
```

Cancel the display restrictions of `-y` or `-D`.

6.6.1.1.30 -y or +unprinted-jobs option

```
-y  
--unprinted-jobs  
+unprinted-jobs
```

Restrict the jobs display to jobs which have not been printed.

6.6.1.1.31 -Z or +no-include-null option

```
-Z  
--no-include-null  
+no-include-null
```

When listing jobs and printers restricted by a `-q` argument, do not include jobs without a destination printer.

6.6.1.1.32 -z or +include-null option

```
-z  
--include-null  
+include-null
```

When listing jobs and printers restricted by a `-q` argument, include jobs without a destination printer in addition.

6.6.2 xspq and xmspq

`xspq` and `xmspq` are fully interactive GTK+ and Motif alternatives to the standard print queue manager `spq`.

As with `spq`, the format of the screen display, the help messages and the command keystrokes can be easily altered to suit your requirements.

Unlike `spq`, there are no specific command line options to `xspq` and `xmspq`.

The facility to change or specify resources settings for an X11 (and hence Motif) program on the command line can be used on `xmspq`. The menu settings on `xspq` are set up in a text file and can be edited as required.

6.7 User administration and charging

6.7.1 spulist

```
spulist [-options] [user ... ]
```

`spulist` lists the permissions of users known to the **Xi-Text** print spooler system.

The invoking user must have *edit admin file* permission to use `spulist`.

If a list of users is given, then the permissions for those users is displayed. Otherwise all the users on the system are displayed.

6.7.1.1 Options

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

6.7.1.1.1 -? or +explain option

```
-?  
--explain  
+explain
```

This option causes a summary of the other options to be displayed without taking further action.

6.7.1.1.2 -D or +default-format option

```
-D  
--default-format  
+default-format
```

Cancel the `-F` option and revert to the default format.

6.7.1.1.3 `-d` or `+default-line` option

```
-d
--default-line
+default-line
```

Display an initial line giving the default options. This is normally displayed, but this option is included in case the user had previously turned it off in the environment.

6.7.1.1.4 `-F` or `+format` option

```
-F string
--format string
+format string
```

Format the output according to the format string given. Sequences introduced by `%` characters select different user attributes. See page [187](#) for a list of format codes.

6.7.1.1.5 `-H` or `+header` option

```
-H
--header
+header
```

Generate a header for each column of the output appropriate for the format.

6.7.1.1.6 `-N` or `+no-header` option

```
-N
--no-header
+no-header
```

Cancel the `-H` option.

6.7.1.1.7 -n or +numeric-sort option

```
-n  
--numeric-sort  
+numeric-sort
```

Sort the list of users by the numeric user id. This is the default.

6.7.1.1.8 -S or +no-user-lines option

```
-S  
--no-user-lines  
+no-user-lines
```

Suppress the user lines. It is an error to invoke this and the `-s` option as well.

6.7.1.1.9 -s or +no-default-line option

```
-s  
--no-default-line  
+no-default-line
```

Suppress the initial line giving the default options.

6.7.1.1.10 -U or +user-lines option

```
-U  
--user-lines  
+user-lines
```

Display the user lines. These are normally displayed but this option is included in case the user had turned it off in the environment.

6.7.1.1.11 -u or +user-sort option

```
-u  
--user-sort  
+user-sort
```

Sort the list of users by the user name.

6.7.1.1.12 +freeze-current option

```
--freeze-current  
+freeze-current
```

Save all the current options in a `.xitext` file in the current directory with keyword `SPULIST`.

No further action will be taken if this option is specified.

6.7.1.1.13 +freeze-home option

```
--freeze-home  
+freeze-home
```

Save all the current options in a `.xitext` file in the user's home directory with keyword `SPULIST`.

No further action will be taken if this option is specified.

6.7.1.2 Format argument

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

Each column is padded on the right to the length of the longest entry.

If a header is requested, the appropriate abbreviation is obtained from the message file and inserted.

```
%% insert a single % character  
%a allowed form types  
%b allowed printer types  
%c allowed class code  
%d default priority  
%f default form type  
%l minimum priority  
%m maximum priority  
%n maximum copies  
%o default printer  
%p privileges  
%u user name
```

The string `DEFAULT` replaces the user name in the default values line.

Note that the various strings are read from the message file, so it is possible to modify them as required by the user.

The default format is

```
%u %d %l %m %n %c %p
```

6.7.1.3 Privileges format

The following are output via the `%p` format. Note that the actual strings are read from the message file, and are the same ones as are used by `spuchange`(see page 194). `ALL` is displayed if all privileges are set.

<code>Adm</code>	edit admin file
<code>Masq</code>	user can masquerade as others
<code>Stp</code>	stop scheduler
<code>Form</code>	select forms other than restriction pattern
<code>Otherp</code>	select printers other than restriction pattern
<code>Cpri</code>	change priority once queued
<code>Otherj</code>	edit other users' jobs.
<code>Prinq</code>	select printer list
<code>Hgo</code>	stop and start printers
<code>Anyp</code>	select any priority once queued
<code>Cdef</code>	change own default priority within range
<code>Addp</code>	add and delete printers
<code>Cover</code>	override class
<code>Unq</code>	unqueue jobs
<code>Votj</code>	view (but not change) other users' jobs
<code>Remj</code>	access remote jobs
<code>Remp</code>	access remote printers
<code>Accessj</code>	access non-displayed job attributes
<code>Freeze</code>	freeze parameters from display

6.7.2 spuchange

```
spuchange [-options] [ users ]
```

`spuchange` may be used to update the user permissions file giving the user profiles of various users and the operations which they may be permitted to perform within the **Xi-Text** system, together with the default permissions which are assigned by default to new **Xi-Text** users.

The "password dump" file is no longer supported, but the options which used to apply to them are still recognised so as not to break existing scripts.

The invoking user must have *edit admin file* permission.

A list of users to adjust is supplied unless the only changes are to the defaults.

6.7.2.1 Options

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

6.7.2.1.1 `-?` or `+explain` option

```
-?  
--explain  
+explain
```

This option causes a summary of the other options to be displayed without taking further action.

6.7.2.1.2 `-A` or `+copy-defaults` option

```
-A  
--copy-defaults  
+copy-defaults
```

Copy the default profile to all users.

This is done *after* any changes are made to the defaults but *before* any changes are made to the named users.

The privileges of the invoking user, `root` and `spooler` are not changed by this operation.

6.7.2.1.3 `-c` or `+class` option

```
-c nnnn  
--class nnnn  
+class nnnn
```

Set the class code of the user(s) as specified by the argument.

6.7.2.1.4 -D or +set-defaults option

```
-D  
--set-defaults  
+set-defaults
```

Indicate that the other options are to apply to the default profile.

6.7.2.1.5 -d or +default-priority option

```
-d n  
--default-priority n  
+default-priority n
```

Set the default priority to the number given, which must be between 1 and 255 and normally in between the minimum and maximum priority.

6.7.2.1.6 -F or +form-allowed option

```
-F pattern  
--form-allowed pattern  
+form-allowed pattern
```

This sets the pattern of allowed form types to which users restricted to particular forms only are restricted.

6.7.2.1.7 -f or +default-form option

```
-f form  
--default-form form  
+default-form form
```

This sets the default form type.

6.7.2.1.8 -l or +min-priority option

```
-l n  
--min-priority n  
+min-priority n
```

This sets the minimum priority, which must be between 1 and 255 and less than or equal to the maximum priority.

6.7.2.1.9 -m or +max-priority option

```
-m n  
--max-priority n  
+max-priority n
```

This sets the maximum priority, which must be between 1 and 255 and greater than or equal to the minimum priority.

6.7.2.1.10 -N or +no-rebuild option

```
-N  
--no-rebuild  
+no-rebuild
```

This option is used to cancel rebuilding of the user file set by the `-R` option, but now has no effect.

6.7.2.1.11 -n or +max-copies option

```
-n  
--max-copies  
+max-copies
```

This sets the maximum number of copies the user may set on a job, from 0 to 255.

6.7.2.1.12 -O or +ptr-allowed option

```
-O pattern  
--ptr-allowed pattern  
+ptr-allowed pattern
```

This sets the pattern of allowed printer names which the user may select to which users who are restricted to particular printers only are restricted.

6.7.2.1.13 -o or +default-ptr option

```
-o printer  
--default-ptr printer  
+default-ptr printer
```

This option sets the default printer, possibly to an empty string.

6.7.2.1.14 -p or +privileges option

```
-p privs  
--privileges privs  
+privileges privs
```

set the privileges of the user(s) as specified by the argument.

6.7.2.1.15 -R or +rebuild-file option

```
-R  
--rebuild-file  
+rebuild-file
```

This used to rebuild the user permissions file, but now does nothing.

6.7.2.1.16 -s or +no-copy-defaults option

```
-s  
--no-copy-defaults  
+no-copy-defaults
```

Cancel the effect of the `-A` option

6.7.2.1.17 -u or +set-users option

```
-u  
--set-users  
+set-users
```

Indicate that the other options are to apply to the users specified on the rest of the command line.

6.7.2.1.18 -X or +dump-passwd option

```
-X  
--dump-passwd  
+dump-passwd
```

This used to dump out the hash table of the password file but now does nothing.

6.7.2.1.19 -Y or +default-passwd option

```
-Y  
--default-passwd  
+default-passwd
```

This used to reset the default handling of password hash file dump but now does nothing.

6.7.2.1.20 -Z or +kill-dump-passwd option

```
-Z  
--kill-dump-passwd  
+kill-dump-passwd
```

This used to delete any existing dumped password hash file but now does nothing.

6.7.2.1.21 +freeze-current option

```
--freeze-current  
+freeze-current
```

Save all the current options in a `.xitext` file in the current directory with keyword {filenameSPUCHANGE}.

No further action will be taken if this option is specified.

6.7.2.1.22 +freeze-home option

```
--freeze-home  
+freeze-home
```

Save all the current options in a `.xitext` file in the user's home directory with keyword {filename-SPUCHANGE}.

No further action will be taken if this option is specified.

6.7.2.2 Users or default

In one operation `spuchange` either adjusts the default permissions, to be applied to new users, if `-D` is specified, or specified users, if nothing or `-u` is specified. So first set the required defaults:

```
spuchange -D -n 20 -p Form,Prinq,Hgo,Cdef -A
```

Then set named users

```
spuchange -p ALL jmc sec
```

6.7.2.3 Rebuilding the user control file

This is no longer necessary. The options are still provided to avoid breaking existing scripts but do nothing.

6.7.2.4 Dumping the password file

This is no longer necessary. The options are still provided to avoid breaking existing scripts but do nothing.

6.7.2.5 Privileges

The following may be specified as the argument to `-p`, as one or more (comma-separated) of the following codes, optionally preceded by a minus to turn off the corresponding privilege. These codes are the same as those displayed by `spulist`.

`spuchange` disregards the case of the codes entered.

<code>Adm</code>	edit admin file
<code>Masq</code>	masquerade as other users
<code>Stp</code>	stop scheduler
<code>Form</code>	select forms other than restriction pattern
<code>Otherp</code>	select printers other than restriction pattern
<code>Cpri</code>	change priority once queued
<code>Otherj</code>	edit other users' jobs.
<code>Prinq</code>	select printer list
<code>Hgo</code>	stop and start printers
<code>Anyp</code>	select any priority once queued
<code>Cdef</code>	change own default priority within range
<code>Addp</code>	add and delete printers
<code>Cover</code>	override class
<code>Unq</code>	unqueue jobs
<code>Votj</code>	view (but not change) other users' jobs
<code>Remj</code>	access remote jobs
<code>Remp</code>	access remote printers
<code>Accessj</code>	access non-displayed job attributes
<code>Freeze</code>	freeze parameters from display

ALL may be used to denote all of the permissions.

For example:

```
-p Otherj,Otherp
-p ALL,-Adm
```

Notice how ALL is set first and then Adm taken away in the second example.

A hexadecimal value is also accepted, but this is intended only for the benefit of the installation routines.

The privileges of root and spooler can never be changed from being all set.

6.7.3 spuser

```
spuser [-options ]
```

spuser provides 3 functions:

With no arguments it lists the permissions for the invoking user and exits.

With the -c option it enables the invoking user to edit his own default priority, form type and printer. The user must have *change default priority* permission to do this.

With the -i option it enables the invoking user to interactively review and update the list of permissions for all users. The user must have *edit admin file* permission to do this.

Please see chapter 7 for more details of the interactive commands. This section focuses on the command-line options which may be used to control the initial display.

6.7.3.1 Options

The environment variable on which options are supplied is SPUSER and the environment variable to specify the help file is SPUSERCONF.

Certain commands available on-screen enable many of these options to be changed and saved in configuration files. (There is no “freeze” option because of this).

6.7.3.1.1 -? or +explain option

```
-?
--explain
+explain
```

This option causes a summary of the other options to be displayed without taking further action.

6.7.3.1.2 -B or +no-help-box option

```
-B  
--no-help-box  
+no-help-box
```

Put help messages in inverse video rather than in a box (this is the default).

6.7.3.1.3 -b or +help-box option

```
-b  
--help-box  
+help-box
```

Put help messages in a box rather than displaying inverse video.

6.7.3.1.4 -c or +form-and-prio option

```
-c  
--form-and-prio  
+form-and-prio
```

Interactively set default priority, form type and printer for the invoking user.

6.7.3.1.5 -d or +display option

```
-d  
--display  
+display
```

This is the default if no options are given. A list of permissions is output to the standard output.

6.7.3.1.6 -H or +keep-char-help option

```
-H  
--keep-char-help  
+keep-char-help
```

When displaying a help screen, interpret the next key press as a command as well as clearing the help screen. This is the default.

6.7.3.1.7 -h or +lose-char-help option

```
-h  
--lose-char-help  
+lose-char-help
```

Discard whatever key press is made to clear a help screen.

6.7.3.1.8 -i or +administer option

```
-i  
--administer  
+administer
```

View and update the list of users.

6.7.3.1.9 -M or +no-error-box option

```
-M  
--no-error-box  
+no-error-box
```

Put error messages in inverse video rather than in a box.

6.7.3.1.10 -m or +error-box option

```
-m  
--error-box  
+error-box
```

Put error messages in a box rather than displaying inverse video.

6.7.3.1.11 -n or +numeric-sort option

```
-n  
--numeric-sort  
+numeric-sort
```

Sort users by numeric userid (default).

6.7.3.1.12 -u or +user-sort option

```
-u  
--user-sort  
+user-sort
```

Sort users by user name.

6.7.4 xspuser and xmspuser

[xspuser](#) and [xmspuser](#) are fully interactive GTK+ and Motif alternatives to the standard user control program [spuser](#) but only in the fully interactive mode, so they cannot be started by a user without *Edit admin file* privilege.

Unlike [spuser](#) there are no specific command line options to [xspuser](#) and [xmspuser](#). The facility to change or specify resources settings for an X11 (and hence Motif) programs on the command line can be used.

The path name for the alternative message file is `XSPUSERCONF` and `XMSPUSERCONF` respectively. By default they share the file `xmspuser.help` in the directory `/usr/spool/progs`.

6.7.5 spcharge

```
spcharge [-options] [user] ...
```

[spcharge](#) is now deprecated as charging is no longer supported.

If invoked, it prints a warning message and exits.

6.8 Web browser interface support

The following programs are parts of the web browser interfaces but are not documented here.

<code>rspccgi</code>	Operations on printers CGI program
<code>rspcgi</code>	List printers CGI program
<code>rsqccgi</code>	Operations on jobs CGI program
<code>rsqcgi</code>	List jobs CGI program
<code>rsqrcgi</code>	Create jobs CGI program
<code>rsqdcgi</code>	Delete jobs CGI program
<code>rsqvcgi</code>	View jobs CGI program
<code>spccgi</code>	Operations on printers CGI program
<code>spcgi</code>	List printers CGI program
<code>sqccgi</code>	Operations on jobs CGI program
<code>sqcgi</code>	List jobs CGI program
<code>sqrcgi</code>	Create jobs CGI program
<code>sqdcgi</code>	Delete jobs CGI program
<code>sqvcgi</code>	View jobs CGI program

6.9 Spool directory management

6.9.1 `xt-cjlist`

```
xt-cjlist [-D dir] [-v n] [-e n] [-u] [-s] [-f] jobfile outfile workdir
```

`xt-cjlist` converts **Xi-Text** job files held in the spool directory to an executable shell script which may be used to recreate them. This may be useful for backup purposes or for one stage in upgrading from one release of **Xi-Text** to another.

The jobs are copied into the backup directory `workdir`, and the generated shell script file `outfile` refers to files in that directory.

`xt-cjlist` understands the format of the saved job file for versions of **Xi-Text** going back to release 5, and when presented with a saved file, will attempt to work out from the format which release it relates to. If required, it will skip apparent errors in the job file.

In addition to options, three arguments are always supplied to `xt-cjlist`.

Job list file	This is the file containing the attributes of the jobs, <code>spsched_jfile</code> in the spool directory, by default <code>/usr/spool/spd</code> , or as relocated by re-specifying <code>SPOOLDIR</code> .
Output file	This file is created by <code>xt-cjlist</code> to contain the executable shell script which may be used to create the jobs.
Backup directory	This directory is used to hold the job data.

6.9.1.1 Options

Note that this program does not provide for saving options in `.xitext` files.

6.9.1.1.1 -D option

`-D directory`

This option specifies the source directory for the jobs and job file rather than the current directory. It can be specified as `$SPOOLDIR` or `${SPOOLDIR-/usr/spool/spd}` etc and the environment and/or `/etc/Xitext-config` will be interrogated to interpolate the value of the environment variable given.

If you use this, don't forget to put single quotes around it thus:

```
xt-cjlist -D '${SPOOLDIR-/usr/spool/spd}' ...
```

otherwise the shell will try to interpret the `$` construct and not `xt-cjlist`.

6.9.1.1.2 -e option

`-e n`

Tolerate `n` errors of the kinds denoted by the other options before giving up trying to convert the file.

6.9.1.1.3 -f option

`-f`

Ignore format errors in the saved jobs file, up to the limit of errors given by the `-e` option.

6.9.1.1.4 -s option

`-s`

Ignore file size errors in the saved jobs file (up to the number of total errors given by the `-e` option).

6.9.1.1.5 -u option

`-u`

Do not check user names in the saved job file.

6.9.1.1.6 -v option

`-v number`

Tell `xt-cjlist` that the jobs file is for release number of **Xi-Text**, where number is 20 to 23. This may be necessary where the user file is corrupted and `xt-cjlist` cannot guess what is meant.

6.9.2 xt-cplist

```
xt-cplist [ -D dir ] [ -v n ] [ -e n ] [ -s ] [ -f ] ptrfile outfile
```

`xt-cplist` converts **Xi-Text** printers held in the spool directory to an executable shell script which may be used to re-install them. This may be useful for backup purposes or for one stage in upgrade from one release of **Xi-Text** to another.

`xt-cplist` understands the format of the saved printers file for versions of **Xi-Text** going back to release 20, and when presented with a saved file, will attempt to work out from the format which release it relates to.

In addition to options, two arguments are always supplied to `xt-cplist`.

Note that you will also need to backup the printer definitions directory and subdirectories under it, by default in `/usr/spool/printers`.

Printer list file This is the file containing the attributes of the installed printers, `spsched_pfile` in the spool directory, by default `/usr/spool/spd`, or as relocated by re-specifying `SPOOLDIR`.

Output file This file is created by `xt-cplist` to contain the executable shell script, containing `spadd` commands, which may be used to recreate the printers.

6.9.2.1 Options

Note that this program does not provide for saving options in `.xitext` files.

6.9.2.1.1 -D option

`-D directory`

This option specifies the source directory for the printers rather than the current directory. It can be specified as `$SPOOLDIR` or `${SPOOLDIR-/usr/spool/spd}` etc and the environment and/or `/etc/Xitext-config` will be interrogated to interpolate the value of the environment variable given.

If you use this, don't forget to put single quotes around it thus:

```
xt-cplist -D '${SPOOLDIR-/usr/spool/spd}' ...
```

otherwise the shell will try to interpret the `$` construct and not `xt-cplist`.

6.9.2.1.2 -e option

`-e n`

Tolerate *n* errors of the kinds denoted by the other options before giving up trying to convert the file.

6.9.2.1.3 -f option

`-f`

Ignore format errors in the saved printers file, up to the limit of errors given by the `-e` option.

6.9.2.1.4 -s option

`-s`

Ignore file size errors in the saved printers file (up to the number of total errors given by the `-e` option).

6.9.2.1.5 -v option

`-v number`

Tell `xt-cplist` that the printers file is for release number of **Xi-Text**, where number is 20 to 23. This may be necessary where the user file is corrupted and `xt-cplist` cannot guess what is meant.

6.9.3 xt-spuconv

```
xt-spuconv [-D dir] [-v n] [-e n] [-s] [-f] userfile outfile
```

`xt-spuconv` converts the **Xi-Text** user file, which is usually `spufile` or `spufile23` held in the spool directory to an executable shell script file `outfile`, which if executed, would recreate the **Xi-Text** users' permissions with the same options and privileges.

`xt-spuconv` understands the format of the saved user file for versions of **Xi-Text** going back to release 20, and when presented with a saved file, will attempt to work out from the format which release it relates to.

In addition to options, two arguments are always supplied to `xt-spuconv`.

- User file This is the file containing the attributes of the user permissions, `spufile23` in the spool directory, by default `/usr/spool/spd`, or as relocated by re-specifying `SPOOLDIR`.
- Output file This file is created by `xt-spuconv` to contain the executable shell script, containing `spuchange` commands, which may be used to recreate the user file. This file should be run before restarting the scheduler.

6.9.3.1 Options

Note that this program does not provide for saving options in `.xitext` files.

6.9.3.1.1 -D option

`-D directory`

This option specifies the source directory for the user file. It can be specified as `$SPOOLDIR` or `${SPOOLDIR-/usr/spool/spd}` etc and the environment and/or `/etc/Xitext-config` will be interrogated to interpolate the value of the environment variable given.

If you use this, don't forget to put single quotes around it thus:

```
xt-spuconv -D '${SPOOLDIR-/usr/spool/spd}' ...
```

otherwise the shell will try to interpret the `$` construct and not `xt-spuconv`.

6.9.3.1.2 -e option

`-e n`

Tolerate `n` errors of the kinds denoted by the other options before giving up trying to convert the file.

6.9.3.1.3 -f option

`-f`

Ignore format errors in the saved user file, up to the limit of errors given by the `-e` option.

6.9.3.1.4 -s option

`-s`

Ignore file size errors in the saved user file (up to the number of total errors given by the `-e` option).

6.9.3.1.5 -v option

`-v number`

Tell `xt-spuconv` that the user file is for release number of **Xi-Text**, where number is 20 to 23. This may be necessary where the user file is corrupted and `xt-spuconv` cannot guess what is meant.

6.9.4 xt-ripc

```
xt-ripc [-d] [-r] [-F] [-A] [-D secs] [-P psarg] [-G] [-n] [-o file]
        [-S dir] [-x] [-B n] [-N char]
```

`xt-ripc` traces, and/or optionally monitors or deletes IPC facilities for **Xi-Text**. Many of the facilities are used for debugging, but it also serves as a quick method of deleting the IPC facilities, being easier to use than `ipcs` and `ipcrm`, in the event that the scheduler has crashed or been killed without deleting the IPC facilities.

To use this facility, just run `xt-ripc` thus:

```
xt-ripc -d >/dev/null
```

The diagnostic output may be useful as it reports any inconsistencies.

The monitoring option can be used to diagnose processes, possibly not **Xi-Text** ones, which are interfering with **Xi-Text** shared memory segments, in cases where a third-party application is suspected of damaging the shared memory.

`xt-ripc` also checks for errors in memory-mapped files where the version of **Xi-Text** is using those rather than shared memory.

6.9.4.1 Options

Note that this program does not provide for saving options in `.xitext` files.

6.9.4.1.1 -A option

`-A`

Display details of jobs and variables. This often generates a lot of output and is not really necessary.

6.9.4.1.2 -D option

`-D secs`

Monitor which process has last attached to the job shared memory segment and report apparent corruption, polling every `secs` seconds.

6.9.4.1.3 -d option

`-d`

Delete the IPC facilities after printing out contents. This saves messing with arguments to [ipcrm\(1\)](#).

6.9.4.1.4 -f option

`-f`

Display the free chains for jobs and variables. This generates a lot of output and isn't usually necessary.

6.9.4.1.5 -n option

`-n`

Suppress display from `-D` option if everything is OK.

6.9.4.1.6 -o option

`-o outfile`

Output to *outfile* rather than standard output. Set it to `/dev/null` if you don't want to see any output.

6.9.4.1.7 -P option

`-P psarg`

Specify argument to [ps\(1\)](#) to invoke if corruption detected when monitoring with `-D` option. The output is passed through [fgrep\(1\)](#) to find the line (if any) with the process id of the process which last attached to the shared memory.

6.9.4.1.8 -G option

`-G`

Used in conjunction with the `-P` option, the output from [ps\(1\)](#) is displayed in full, without passing it through [fgrep\(1\)](#).

6.9.4.1.9 -r option

`-r`

Read and display the entries on the message queue. This is normally suppressed because they can't be "peeked at" or "unread".

6.9.4.1.10 -S option

`-S dir`

This is only relevant for versions of **Xi-Text** which use memory-mapped files rather than shared memory. It specifies the location of the spool directory. If this is not specified, then the master configuration file `/etc/Xitext-config` is consulted to find the spool directory location, or failing that, the directory `/usr/spool/spd` is used.

6.9.4.1.11 -x option

`-x`

Dump the contents of shared memory or memory-mapped files in hexadecimal and ASCII characters.

6.9.4.1.12 -B option

`-B n`

Where *n* may be 1 to 8, specify the width of the hexadecimal dump output as a number of 32-bit words.

6.9.4.1.13 -N option

`-N char`

Replace the character in the ASCII part of the hexadecimal dump to represent non-ASCII characters with the specified character (the first character of the argument). The default is `.`. To specify a space, you may need to use quotes thus: `-N ' '`

6.9.4.2 Example

To delete all IPC facilities after **Xi-Text** has crashed.

```
xt-ripc -d -o /dev/null
```

To monitor the job shared memory segment for errors, printing out the `ps(1)` output (where the full listing is obtained with `-ef`) search for the process id of the last process to attach to the segment. Print out the contents of the segment including in hexadecimal after corruption is detected.

```
xt-ripc -D 30 -P -ef -o joblog -A -x
```

6.9.5 setspdir

```
setspdir[-s n] [-r] [-d] [-f form] [-y]
```

`setspdir` tidies up the **Xi-Text** spool directory, removing old spool files, or alternatively re-queuing them with a given default form type, and eliminating jobs where the spool file is missing.

Optionally, it may be used to split the main spool directory into a series of subdirectories `000`, `001` etc, reducing the size of the main spool directory to manageable proportions and also speeding up file searches. Alternatively, the subdirectories may be linked to different file systems by the user.

For reasonably obvious reasons, `setspdir` should be run with the spooler shut down.

6.9.5.1 Options

The following options are available to `setspdir`. There is no option for saving settings to files.

6.9.5.1.1 -d option

```
-d
```

Clear (delete) any orphaned job files found in the directory or directories.

6.9.5.1.2 -f option

```
-f form
```

Requeue any orphaned job files with the form type `form` rather than deleting them.

6.9.5.1.3 -r option

```
-r
```

Instead of moving jobs between subdirectories, renumber the jobs instead. This should only apply when there were existing subdirectories, and the number is being changed.

6.9.5.1.4 -s option

`-s n`

Split the spool directory into *n* subdirectories, or change the number of subdirectories from what it is at the moment to *n*, where *n* may be 0 (to convert back to a single spool directory), or a number up to 999.

We recommend that *n* be a prime number, as this minimises the chance of a cycle of operations which is a factor of the number of subdirectories causing just a small number of subdirectories to be filled more or less exclusively. Confirmation will be requested if *n* is not prime.

The subdirectories are given the names 000 to 999, and each job will be moved to a subdirectory given by the remainder when its job number (the number after the `SP`) is divided by the number of subdirectories.

6.9.5.1.5 -y option

`-y`

Answer 'yes' to any questions which `setspdir` asks.

6.9.6 xipasswd

```
xipasswd [-u user] [-p password] [-f] [-d] [-F file]
```

`xipasswd` sets a password for the current user or a specified user if `-u` is given. This is separate and distinct from the user's login password. This password is used by the web interfaces, the Windows interfaces and the APIs for both **Xi-Text** and **Xi-Batch**.

The reason for doing this is because it is considered insecure to possibly repeatedly try login passwords from user programs.

If any users have a password set in this way, then all users will have to have a password in the file to use any of the interfaces requiring a password.

Unlike the Unix `passwd(1)` routine, the old password is not prompted for and there is no confirmation.

6.9.6.1 Options

Note that this program does not provide for saving options in `.xitext` files.

6.9.6.1.1 -u option

`-u user`

Set password for given user. This may only be for other than the current user if `xipasswd` is invoked by `root`.

6.9.6.1.2 -p option

`-p passwd`

Specify the password to be set other than prompting for it.

6.9.6.1.3 -f option

`-f`

It is normally considered an error to include a password for `root` for the same reasons that the password file is separate. However this option may be set to insist upon it.

6.9.6.1.4 -d option

`-d`

Delete the user's password from the file.

6.9.6.1.5 -F option

`-F file`

Use file to hold the password. The default if no file is given is `/usr/spool/progs/xipasswd`. Any number of `-F` options may be given to set up several password files at once.

6.10 Licence, installation, version control

6.10.1 xt-vwrite

`xt-vwrite`

`xt-vwrite` installs a licence for **Xi-Text** on the invoking machine.

The user will be asked to confirm the date and time on his machine, then asked for the organisation name, and possibly a serial number. The serial number is often read from the `README` file issued with the distribution.

The codes are typed in as 4 blocks of 8 hexadecimal digits separated by spaces (a modulo 17 checksum is used for each block of 3 digits to enable transcription errors to be detected, so `G` may appear).

The letters may be entered in upper or lower case.

Note that for versions of **Xi-Text** prior to release 6, the checksums were not included, and the codes are 3 blocks of 8 hexadecimal digits.

6.10.1.1 Options

Note that this program does not provide for saving options in `.xitext` files.

6.10.1.1.1 -L option

`-L`

Specify non-network (local) licence.

6.10.1.1.2 -N option

`-N`

Specify networked licence.

6.10.1.1.3 -I option

`-I`

Try to download a licence automatically from the Xi Software website.

6.10.1.1.4 -n option

`-n`

Do not try to download a licence automatically from the Xi Software website.

6.10.2 `xt-checklic`

`xt-checklic`

`xt-checklic` reports the status of the **Xi-Text** licence, also noting whether the licence is for networks, is an emergency licence, and whether there is a limit of users or printers.

6.10.2.1 Diagnostics

Messages about invalid checksums or inodes probably indicate that the licence is not valid.

6.10.3 `xt-emerglic`

```
xt-emerglic [-N] [-L]
```

`xt-emerglic` installs an emergency licence in cases where, perhaps due to backup and recovery, the original licence was lost.

After use, users will get all sorts of messages about the need to install a proper licence.

The user will be asked whether the machine is to be networked or not, although `xt-emerglic` will guess and ask for confirmation.

6.10.3.1 Options

Note that this program does not provide for saving options in `.xitext` files.

6.10.3.1.1 `-L` option

```
-L
```

Specify non-network (local) licence.

6.10.3.1.2 `-N` option

```
-N
```

Specify networked licence.

6.10.4 `xt-triallic`

```
xt-triallic
```

`xt-triallic` installs a trial **Xi-Text** licence during installation.

It is not intended to be run more than once.

6.10.5 whatvn

```
whatvn [options] file ... bkm:Btuserdescribkm:Btuserdescr
```

`whatvn` discovers the location of installed software and extracts the version number for each module. It also displays certain compile-time options which need to be consistent.

There are inbuilt assumptions that to some extent that the directories installed to bear some resemblances to those used by Xi Software products or can be found in the master configuration file `/etc/Xitext-config`.

Typically, the program can be run thus:

```
$ whatvn spshed spq whatvn
spq      23.484, SHM, FLOCK
spshed  23.481, SHM, FLOCK
whatvn  1.62
```

This output would show that the installed versions of `spshed` and `spq` were 23.481 and 23.484 respectively and they were built with compile-time options to use shared memory and file-locking.

Minor version number differences in these probably will not matter but the compile time options should be the same.

It would also show that the installed version of `whatvn` was 1.62.

It is also possible to inspect the contents of a directory, uncompressing files on the fly, to discover the version numbers of the binaries in the directory.

6.10.5.1 Options

Note that this program does not provide for saving options in `.xitext` files.

6.10.5.1.1 -P option

```
-P path
```

Overrides the default path, as a colon-separated list of directories for which files are searched.

The directory `/usr/spool/progs` is added to the path in order to find internal programs.

If the master configuration file `/etc/Xitext-config` exists and defines `SPROGDIR`, then this is used in place of `/usr/spool/progs`.

Note that if **Xi-Batch** is present on the machine the file `/etc/Xibatch-config` will similarly be parsed and any definition of `SPROGDIR` in that added to the path.

6.10.5.1.2 -l option

`-l`

This ignores the search path and looks only at files in the current directory (or as overridden by `-D` and its subdirectories).

If a compressed version of one of the files is found, it will uncompress it on the fly and use the result.

6.10.5.1.3 -D option

`-D directory`

In conjunction with the `-l` option, specify the search directory.

6.10.5.1.4 -d option

`-d`

Display the build date of each file.

6.10.5.1.5 -v option

`-v`

Produce fuller output, in particular show the full path to each file examined and show the associated product.

6.10.5.1.6 -q option

`-q`

Suppress messages about files not found, which cannot be opened or are indecipherable.

6.10.5.1.7 -t option

`-t`

Try to produce nicely-aligned columns.

6.10.5.1.8 -F option

`-F file`

The installation suite used to include a file `/etc/whatvnrnc` which mapped the product name **Xi-Text** to a collection of files, so you could type

```
whatvn xibatch
```

to see a display of each component program and the version number, together with a calculated overall version number.

This is not necessary now that each program file incorporates the overall version number, but the old method is supported.

The default location may be overridden with this option.

```
# Note various alternate names precede the :
# This is not a definitive list!

xitext, Xi-Batch, spooler :      spq, spuser, spd,
                                spshed, spdinit

# Note continuation line

xibatch, Xi-Text, batch :      btq, btuserName, btsched
```

6.10.6 zinstall

```
zinstall [options] files
```

`zinstall` is not currently used for installation but is left in place as a potentially useful program.

It can search for a program file, possibly in a lower-level subdirectory, uncompress it if necessary, install it in the appropriate `bin` directory and set the modes and ownership.

It can check for the version currently

The software is copied or compared from the current directory, or optionally from subdirectories of the current directory to a specified directory. If it is compressed, it is uncompressed or un-zipped on the fly.

It can be used to verify that the permissions and ownership are correct, and optionally to reset them.

It can be used to check that the installed version is up-to-date with a version in the current directory, and optionally to install a newer version (this requires the version numbering to conform to Xi Software standards).

It can also be used to deinstall software.

The four modes are selected by means of the mutually exclusive options `-I`, `-V`, `-C` and `-U`. One of these options must always be given.

6.10.6.1 Options

Note that there is no mechanism for saving options in a `.xitext` file or anywhere else.

6.10.6.1.1 -I option

`-I`

Select the installation option.

6.10.6.1.2 -V option

`-V`

Select the verify option.

Verify doesn't actually change anything, it just returns an exit code of 100 if something is found to be inconsistent, unless the `-f` option is given.

6.10.6.1.3 -C option

`-C`

Select the check option.

Note that Xi Software version numbering standards are assumed, held in comment sections of the binaries.

Check doesn't actually change anything, it just returns an exit code of 100 if something is found to be inconsistent, unless the `-f` option is given.

6.10.6.1.4 -U option

`-U`

Select the uninstall option. Most of the other options are ignored.

6.10.6.1.5 -t option

`-t type`

Indicates the type of the file as being `type`.

Currently the only type which makes any real difference is `LINK`, which means that the second and subsequent named files are installed, or checked to be, hard links to the first named file.

6.10.6.1.6 -D option

`-D directory`

Specify the given *directory* as being the destination directory to install to.

If the specified directory does not have a `/` in it, then it is taken to be the name of an environment variable to expand, so for example:

```
BINDIR=/opt/products/bin
export BINDIR
zinstall -I -D BINDIR ...
```

will use the specified directory as a destination.

The environment variable must be defined, otherwise this will be an error, with the exception of the name `USERPATH`. If this is not defined, then the directories `/usr/local/bin`, `/usr/lbin` and `/usr/bin` are searched in order and the first of those which exists is taken.

If the `-D` option is not provided `USERPATH` is supplied anyway, with the above consequences.

6.10.6.1.7 -O option

`-O directory`

Select the given *directory* as the source directory for files, rather than the current directory.

No environment variable or other expansion takes place on this and it is, of course, an error if the directory does not exist.

6.10.6.1.8 -M option

`-M mode`

Create the destination directory if it does not exist, with the permissions given as an octal string.

When checking or verifying, precede the *mode* with a `+` sign to signify that the permissions on the directory should include at least the bits given, for example

```
-M +111
```

would ensure that the executable (search) permissions were set on for all users.

6.10.6.1.9 -p option

`-p prefix`

Provide a *prefix* to each installed file.

So for example to install the program `ripc` as `xt-ripc` and `cjlist` as `xt-cjlist` use

```
zinstall -I -p xt- ripc cjlist
```

6.10.6.1.10 -N option

`-N`

Ignore files which are non-executable in the list of files to operate upon.

This is handy if you have just built a directory of files and just want to install the programs, for example

```
make
zinstall -I -N *
```

will just install the built programs and ignore the `.o`, `.c` files and the `Makefile`.

6.10.6.1.11 -S option

`-S`

Skip files which look like source files in the list of files to operate upon.

This might be preferable to the `-N` option where library files are being installed, for example.

6.10.6.1.12 -m option

`-m mode`

Set the permissions on the file when installing to be *mode*, given in octal.

When verifying or checking, check that the permissions agree.

If this is not specified, then the file permissions are copied on first installation.

If a re-installation takes place and this is not specified, the permissions of the original file are copied.

6.10.6.1.13 -u option

`-u user`

Set the owner of the destination files to be `user`.

When verifying, check that it is that user.

If this is not specified, then the user will be that of the invoking user on installation.

If a re-installation takes place and this is not specified, the user will be that of the original destination file.

6.10.6.1.14 -g option

`-g group`

Set the group owner of the destination files to be `group`.

When verifying, check that it is that group.

If this is not specified, then the group will be that of the invoking user on installation.

If a re-installation takes place and this is not specified, the group will be that of the original destination file.

6.10.6.1.15 -f option

`-f`

In check and verify modes, attempt to fix anything found to be incorrect.

6.10.6.1.16 -v option

`-v`

Say more about what the program is doing. Without this, check and verify in particular do not say anything but just return non-zero exit codes if the files are out-of-date or incorrect respectively.

6.10.6.1.17 -q option

`-q`

Be silent about non-catastrophic errors like missing files.

Chapter 7

Text screen-based Programs

The following sections provide more information on the text screen-based programs `spq` and `spuser`, with attention to the screen commands. See the previous chapter regarding command-line options to `spq` and `spuser`.

7.1 `spq` – interactive spool queue manager

The following screens and screen-based commands apply to `spq`. Remember, however that all of the screen formats and the key bindings may be adjusted to suit your requirements, so the default commands listed here may have been changed on your system.

7.1.1 Position of cursor on entry

The cursor is placed in either the job section or the printer section on entry to `spq`.

If the user is not allowed access to the printer section, the cursor will always be placed in the job queue section on entry whatever options are specified.

Otherwise the job queue or printer list section may be selected using the `-j` or `-p` options respectively.

If neither option is specified, or `-d` is specified, the job queue will be selected if there are any jobs, otherwise the printer list will be selected.

7.1.2 Screen-based commands - cursor movement

Most of these commands apply everywhere, except when you are in the middle of updating a field such as the form type, where usually only the “help” and “screen refresh” commands apply.

We think that you may well want to assign function keys to some of these options, but you should have appropriate *terminfo* or *termcap* definitions to do this.

Command	Action
<i>j</i> or <i>down cursor</i> key	Move the cursor down one line, possibly scrolling down
<i>k</i> or <i>up cursor</i> key	Move the cursor up one line, possibly scrolling up
<i>Ctrl-f</i>	Move forward a screenful
<i>Ctrl-b</i>	Move backward a screenful
<i>Ctrl-d</i>	Move forward half a screenful
<i>Ctrl-u</i>	Move backward half a screenful
<i>B</i> or <i>home</i>	Go to top of screen; if already there scroll to top of list
<i>E</i>	Go to bottom of screen; if already there scroll to end of list
<i>Ctrl-l</i>	Redraw the screen
<i>?</i>	Display some sort of help message.

If the up and down cursor keys work on your terminal, these are equivalent to *k* and *j* and the latter should probably be removed from the valid keys for up and down movement, to minimise potentially ambiguous situations, see key definitions on page 292 on how to do this.

The standard response to an invalid instruction is to flash the screen, if possible, otherwise to ring the bell. This will be accompanied by an error message, displayed, highlighted if possible. Typical invalid instructions include:

- an attempt to move off the end of the display
- an invalid control character
- an invalid operation
- a lack of privilege, such as attempting to abort someone else's job.

7.1.3 Field entry

When a command to change a field is typed the cursor is moved to the start of that field. The existing field entry is undisturbed until the first character is typed. Characters are then accepted up until a carriage return or line feed is typed.

Special control characters in this context are:

Command	Action
<i>escape</i>	Abort the edit, restore the previous value, and quit the command
<i>kill</i> char	Kill all characters typed so far and start again
<i>erase</i> char	Delete the last character typed
<i>?</i>	For numbers - display meaning and range, for strings display possibilities starting with the characters typed so far.
<i>space</i> or <i>tab</i>	For strings only - insert a likely possibility. Successive spaces will cycle through all possibilities. If characters have already been typed the possibilities are restricted to those starting with the characters already typed.

7.1.4 Set and save program options

The program options with which `spq` was entered may be interactively displayed and modified, with immediate effect (except in 3 cases). To select this facility, type the `=` command. A screen like this should be displayed:

```
Setting program options for spq
Restrict to printer :
Restrict to user   :
Restrict to title  :
Limit jobs display: Jobs matching printer plus null
Display only      : All
Local jobs/printers only : No
Sort printers     : Unsorted
Class code       : ABCDEFGHIJKLMNOPabcdefghijklmnop
Confirm abort/delete jobs : Unprinted jobs
Help messages    : Inverse Video
Error messages   : Inverse Video
Clear Help message: Use next command
Page counts      : Yes
If job moves     : Follow job
Printer List size(on entry) : 3
Cursor on entry  : Don't care
Refresh Interval : 10
```

After setting the user is given the option to save the options, which may be done by creating or editing a `.xitext` file in the current directory (i.e. the directory when `spq` was entered) or in the user's home directory.

The printer list size and sort printers options are only effective if the options are saved and `spq` re-entered. Likewise the "cursor on entry" setting only makes sense on entry.

To cycle through all cases, move the cursor to the relevant field and press space. For the two numeric-valued cases just type in the required number. To exit, either type `q`, or else move past the last field. You should receive the message

```
Save current parameters?
```

Type `y` to save the parameters. You should receive a further message asking whether you want to save in the current directory, and if not in the home directory.

7.1.4.1 Restricting the display by printer

It is possible to restrict the display to a set of printers, and the jobs designated for those printers. This includes jobs which, by having no printer name included, may come out on one of the set. The set of printers may contain just one name, a list of names or a list of patterns for matching printer names.

The set of printers is specified on the line

```
Restrict display to printer :
```

They may be given as a comma-separated list of alternatives, including the use of shell-style wild-cards. This is the same as with the `-q` option on the command line, except that there are no quotes round the string. For example:

```
Restrict display to printer : ljet
Restrict display to printer : lp1,ptr3,laser7
Restrict display to printer : ptr*,lp[1-9]
Restrict display to printer : [a-c]?9
```

The wild-card options are:

<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

7.1.4.2 Restricting the display by user

The display may be restricted by username, in a similar fashion to restricting by printer name. The set of users may be a single name, a pattern, a comma separated list of names and patterns. The same shell style wild card options are available for pattern matching as are used for printers.

The set of users is specified on the line

```
Restrict display to user:
```

7.1.5 Job queue section.

There may be blank space at the top of the screen as jobs are deleted, but the cursor can not be moved into it. The size field is displayed in pages preceded by an `=` sign if the job appears to be on separate identifiable pages, otherwise by a `/` sign, in which case it is in bytes. The `-e` option to `spq` suppresses the page indication.

The following commands are available when the cursor is in the job list, in addition to the cursor movement and scrolling commands given above.

Commands generally apply to the job selected by having the cursor alongside it.

Key Action	Key Action
<i>A</i> Delete the job aborting if being printed.	<i>C</i> Copy job without deleting
<i>c</i> Change the number of copies	<i>F</i> Change post-processing flags, if displayed
<i>f</i> Change the form type for the job	<i>H</i> Change the hold time, if displayed
<i>h</i> Change the job title	<i>I</i> Display the job on the screen
<i>K</i> Change class code, if displayed	<i>L</i> Change “local only”, if displayed
<i>M</i> Change “mail attention”, if displayed	<i>m</i> Change “mail result”, if displayed
<i>O</i> Change other options for the job	<i>o</i> Switch to the printer list
<i>p</i> Change the priority of the job	<i>P</i> Change the printer destination
<i>q</i> Quit out of spq	<i>R</i> Edit page range, if displayed
<i>r</i> Set/unset “retain after printing”, if displayed	<i>s</i> Set/unset “no header”, if displayed
<i>T</i> Change the “delete if unprinted” time, if displayed	<i>t</i> Change the “delete if printed” time, if displayed
<i>U</i> Unqueue the job	<i>u</i> Change “post to” user, if displayed
<i>W</i> Change “write attention”, if displayed	<i>w</i> Change “write result”, if displayed
<i>=</i> Update/save program options	<i>:</i> Change “halted at” page, if displayed
<i>#</i> Change “odd/even pages”, if displayed	<i>,</i> Reassign job list display format
<i>X</i> Inspect system error log file	
<i>^</i> search forward for any string parameter	<i>\</i> search backward for any string parameter

Some commands, such as *H*, are only operative if the job display format has been changed to display the relevant job attribute on the main screen. These attributes are usually only edited from the supplementary job attribute screen, displayed using the *O* command. However attributes considered important, such as form type or printer may be edited with a suitable prompt even if they have been excluded from the main job list.

Most commands are simply ignored if the job is being printed. The main exceptions are abort/delete, and change copies.

To edit attributes of jobs, it will generally have to be your job, unless you have the privilege to edit other users' jobs. There is a separate permission to allow you to view the contents of other users' jobs without being able to edit them.

Other commands require you to have specific permissions as well as or in addition to those; you will need permission to select the printer list, to change the priorities of jobs which have been queued, or to “unqueue” a job.

7.1.5.1 Abort job

The *A* command deletes the selected job, aborting printing if it is in progress.

Confirmation is usually requested if the job has not been completely printed at least once, otherwise the job is deleted silently. You can choose between three levels of confirmation:

1. You can turn off all confirmation requests by specifying *-A*.

2. You can accept the default behaviour (possibly restored after a `-A` option with a single `-a` option), which requires confirmation for unprinted jobs only.
3. You can insist that confirmation always be requested with `-a` or multiple `-a` options.

If the job is being printed, then the appropriate action for abort handling for the printer is taken; by default the message `***ABORTED***` will be appended to the output, unless the output is being passed through a filter, in which case the filter process will be sent a `SIGTERM` signal.

Remember that output may not stop immediately if the printers have their own internal buffers to be emptied.

If a job has been retained on the queue via the `-q` option to `spr`, or because this is standard for the paper type, this command may be used to remove it, or it may be deleted automatically when the job has reached the “delete if printed” timeout.

7.1.5.2 Change the number of copies

The `c` command moves the cursor to the copies field of the job. You may change the number of copies from 0 to the maximum which you are permitted. If printing is in progress, the number of copies to be printed includes the current one.

You may reprint a job retained on the queue by resetting the copies field.

7.1.5.3 Change the form type

The `f` command changes the form type. The cursor moves to the job's form type field.

If the form type you want is loaded on one of the printers you may press *space* until it appears in this field, and you can always edit the result by changing a suffix, for example. Alternatively you may type the initial few letters of the form type to restrict the choice to form types starting with those letters.

Typing a `?` will display a list of form types loaded on the printers, or you may type this after typing one or more characters to list just the form types whose names start with those characters.

If you may only use your default form type, then the allowable forms here may be restricted.

7.1.5.4 Priorities

There are two privileges which apply here; you may or may not be allowed to change the priority of a job on the queue, and if you are you may or may not be limited to your standard upper and lower priorities. If you cannot change the priority, you will get an error message.

When in this field, you can type a `?` to see the range of priorities you may set.

7.1.5.5 Change the printer name

The `P` command changes the printer name. You may not be permitted to do this if you are limited to use of your default printer type.

As described in the `-P` option to the `spr` command, this field may be a pattern giving a range of printers which may print the job.

You may cycle through printer names using spaces in the same way as you may cycle through the form types.

7.1.5.6 Unqueue job

The `U` command enables a job to be taken off the queue, perhaps for editing before being returned. To protect secure environments (e.g., where accounts software log files are printed) the user must be granted the unqueue privilege.

When a job is unqueued, two files are created. The first file is a “shell script” or “command” file containing an `spr` command which if run unedited, would resubmit the job with the same options as it had when taken off the queue. The second file is a copy of the job itself, which is then deleted.

When the user invokes this command, `spq` prompts in turn for the directory to write in, the command file, and the job file. The directory to write in defaults to the current directory when `spq` was started.

Environment variables prefixed with `$` are expanded when the directory name is read. Also, certain directories may be specified by using the `~` character, thus:

Sequence	Meaning
<code>~</code>	Invoking user's home (<code>\$HOME</code>) directory
<code>~ user</code>	Home directory of specified <code>user</code>
<code>~+</code>	Current directory when <code>spq</code> started
<code>~-</code>	Value of environment variable <code>OLDPWD</code> if it exists on entry to <code>spq</code>

The directory being written to must be writable by the user invoking `spq` and must have search (execute) permission for user `spooler`.

The `C` command is similar to the `U` command except that the job is not deleted.

7.1.5.7 Other options

Many of the other options for a job may be viewed and changed using the `O` option. The other option screen looks like:

```
Job number 6417 submitted by jmc on 13:10:35 Mon 10 Jan 1999

Title      : passwd
Suppress header : No
Form type  : a4
Post to user : jmc
```

```

Mail message      : No
Write message     : No
Mail attention message : No
Write attention message : No
Class code       : ABCDEFGHIJKLMNOPabcdefghijklmnop
P/proc flags     :
Retain after printing : No
Start page: 1      Last print halted at      : 5
End page        : <end>
Skip odd pages  : No
Skip even pages : No
Reverse skip    : No
Printed         : No
Delete time if not printed (hours) : 168
Delete time if printed (hours)    : 24
Hold job until      : <N/A>
Local job only     : No

```

The `Last print halted at:` field is only displayed if the job had started printing but not run to completion.

Note that the order of entries on this screen, and also the starting one, may be separately modified by adjusting the message file.

These options may be changed in just the same way as the options in the main screen are changed. For “flag” type options you may type `Y` or `T`, `N` or `F` and `~` to set, unset or “toggle” the flag. Note that the “this job has been printed” flag may be set or unset, however it will be set every time the job is printed, and if the “retain after printing” flag is not set and the printer setup file does not specify retain the job will be deleted then.

You may exit a field leaving it unchanged by using the up or down arrow keys, or `Enter` provided that nothing has been typed. You may quit back to the main screen by typing `q` provided that a string is not expected; otherwise move to another field and type `q`. Alternatively map this function onto some function key.

(To get the full benefit of using this and other screens in a smooth fashion, you should make sure that your terminal and **Xi-Text** are set up to use cursor and function keys, particularly for the “help” and “quit” functions. We’d like to distribute versions of the key definitions which include function keys, but terminals and *terminfo* descriptions seem to differ so widely in this area, often by not defining them at all).

You can set a “hold until” time by typing `S` on the field, or delete it by typing `U`. When setting the time you can move across the parts of the time and date using the left and right arrow keys, `h` or `l` respectively. To change the time, press `+` to “increment” the component which the cursor is on or `-` to “decrement” it. The digits of the time and the day of the month may be explicitly overtyped.

A separate help message may be displayed by typing `?` except where this may be legitimately part of a string. You can map the help function onto a function key (see chapter 10 on Configurability for more details) if you want to be able to display help here. The help message includes possible user names when editing the “Post to User” field. If one or more characters have already been typed, only user names starting with that string appear.

In the same way, a space typed in the “Post to User” field will suggest a possibility from the list of users, or from that subset starting with the character or characters already typed.

If the job is currently being printed, **Xi-Text** will display its progress as it is being printed. Should the job disappear whilst you are in this screen, you will receive a message. Just type *Enter* to continue.

If the job was halted whilst being printed, the page at which it was halted will be displayed after the startpage. You may reset this page number also if it is displayed. When the job is printed again, the first copy will commence at the page it was halted at, if applicable, and otherwise at the start page. Second and subsequent copies will commence at the start page. In all cases the last page to be printed will be the end page.

7.1.5.8 View job file

The *I* command displays the job on the screen. The job title and form type are displayed at the top of the screen, together with a “ruler” for measurement of character positions.

Since the whole of the printed page is unlikely to fit on the screen, the file is viewed in “screenfulls”, and the end of printed pages is indicated on screen by highlighted markers, like *End of page 1*. The page counts are based either on the specified delimiters given to *spr*, or if the job has been printed, and the page delimiters reassigned, then these new assignments. If no delimiters are given, then formfeed is taken as the delimiter.

The simplest commands here are *space* or *carriage return* to advance through the file, and *q* to quit back to the main screen. When you are looking at the start, end or halt page, an appropriate message will be displayed at the top right-hand corner of the screen. If there is text which has been truncated on the right to fit onto the screen, then such text is marked by a “>” in the rightmost column.

Facilities are available to:

- Set the start, end and halt pages for the job
- Scroll right and left to view the whole of the printed page on wide stationery
- Go to the beginning or end of the file
- Search forward or backward for a specified string.

The complete set of commands available is:

Command	Action
<i>space</i>	next page
<i>ENTER</i>	next page
<i>k</i> or <i>up cursor</i>	previous page
<i>j</i> or <i>down cursor</i>	next page
<i>^</i>	search forward for string
<i>\</i>	search backward for string
<i>l</i>	scroll left 30%
<i>r</i>	scroll right 30%
<i><</i>	scroll to left margin
<i>></i>	scroll to rightmost column in file containing text
<i>s</i>	set start page of job to current page
<i>e</i>	set end page of job to current page
<i>H</i>	set "halted at" page to current page
<i>q</i>	quit back to job list

The only commands requiring further explanation are the searching commands. These provide a limited ability to search forward and backward through the file for a specified string; they prompt for a string to be searched for; case is insensitive, and the only "wildcard" allowed is ".", which means "any character".

7.1.5.9 Inspect system error log file

The *X* command inspects the contents of the system error log file, *spshed_reps*, in the spool directory, by default */usr/spool/spd*. It may be useful to examine this in case something crashes, especially if a printer halts with error status.

The display starts on the final page of the log, to show the most recent messages. It is similar to the view job display and uses the same commands to move round the file.

7.1.6 Reassign format of job list display

The initial layout of the job list display is read from the message file. The format of the job list display of the main screen may be adjusted and the result saved in a new message file.

To select the display format, type a "," character. The display will change to something like:

```

" "      3   n  Number in queue
<
" "      6   N  Job number
" "      6   u  User submitting job
```

The numbers specify field widths, the code letters give the desired field and the text gives an explanation. Material in quotation marks designate separator characters. Lines prefixed by a < character denote fields which will be left-shifted to overwrite the previous field if a specific instance is too large to fit (e.g. job number prefixed by a machine name; this is inserted sacrificing the number in the queue). There is a similar > character which denotes "skip any fields to the right if too large", but this is mainly useful for printer "feedback" messages.

To adjust the formats, move to the relevant row and type:

Command	Meaning
<i>q</i>	Quit and prompt to save
<i>i</i>	Insert new field before current
<i>a</i>	Insert new field after current
<i>'</i>	Insert new separator before current
<i>"</i>	Insert new separator after current
<i>D</i>	Delete field or separator
<i>w</i>	Reset width
<i>c</i>	Reset format code
<	Toggle left overwrite field
>	Toggle right overwrite field
<i>S</i>	Reset separator text

When a separator is entered, there is a "minimum" of a single space which will be assumed if a null string is given.

The following format codes may be given:

Code	Meaning	Code	Meaning
<i>A</i>	Signify "write attention"	<i>a</i>	Signify "mail attention"
<i>C</i>	Class code	<i>c</i>	Number of copies
<i>F</i>	Post-processing flags	<i>f</i>	Form type
<i>H</i>	Hold time	<i>h</i>	Job title
<i>K</i>	Size in KB	<i>l</i>	Signify "local only"
<i>L</i>	Position reached in KB	<i>m</i>	Signify "mail result"
<i>N</i>	Job number	<i>n</i>	Number in queue
<i>O</i>	Signify odd/even pages	<i>o</i>	Display originating host
<i>P</i>	Printer	<i>p</i>	Priority
<i>Q</i>	Page reached	<i>q</i>	Signify "retain on queue"
<i>R</i>	Page range	<i>s</i>	Signify "suppress header"
<i>S</i>	Size in pages	<i>t</i>	Delete time if printed
<i>T</i>	Delete time if not printed	<i>u</i>	User submitting job
<i>U</i>	Post user	<i>w</i>	Signify "write completion"
<i>W</i>	Submission time		

Notes:

1. Job number includes the machine prefix if the job is on a remote host.
2. Display originating host includes Windows PCs and jobs submitted from remote hosts via the `rspr` command.

When you have finished, type `q` to return. You will be prompted as to whether you want to save the formats. If you do, you can opt to save in your home directory or the current directory. You would choose the former if you want to select this every time you use `spq` and the latter if you wanted it only from the given directory.

A "header" line to the job list section consisting of only a single letter `j` will be replaced by an appropriate header consisting of headings appropriate to the format selected.

7.1.7 Printer list section of `spq`

This section of the screen reflects the active printers. To access it the user must have the appropriate privilege. A further privilege enables access to the halt and restart commands `H`, `G`, `!` and `h`, and yet a further privilege to `N` and `D`.

Each printer known to the spooler may be in one of 8 states:

<code>Startup</code>	The printer has been started with <code>G</code> , or with <code>spstart</code> , and is being initialised with the <code>setup</code> string.
<code>Idle</code>	The printer is ready to print jobs with the given form type.
<code>Printing</code>	The printer is printing a job, whose number and user are shown.
<code>Shutdown</code>	The printer is in the process of shutting down, i.e. it was idle or printing, but has been halted, and is now shutting down the printer. This state is transient, probably not being seen on lightly loaded machines and printers with simple drivers / setup files.
<code>Halted</code>	No jobs are going to the printer. This is the initial state. You will need to change to this state to change the form type.
<code>a/w oper</code>	In this state the printer is awaiting a confirmation from the operator before continuing, either because it is in single job mode, or because an alignment page has just been printed. The spooler requires approval before continuing. The marker <code>N/A</code> (not aligned) appears on the right if it requires alignment.
<code>Offline</code>	The printer has been detected as being off-line. This state is actually a variation on the <code>Halted</code> state. A message may be displayed to the right of this column giving further details.
<code>Error</code>	An unrecoverable error has been detected on the printer. A message may be displayed to the right of this column giving further details.

Some causes of errors are:

- Syntax errors in setup files
- Inaccessible device special files
- I/O errors other than those caused by off-line detection.

For the `Error` state, a message is left in the file `spshed_reps` in the spool directory, by default `/usr/spool/spd`, describing the problem. You can inspect this with the `X` command.

The extra commands available in the printer section of the screen are, in addition to the cursor movement and scrolling commands described earlier:

Key	Action	Key	Action
!	Interrupt current job	A	Abort current job
c	Edit description field for printer	C	Reset the class code
D	Delete printer from list	f	Change form type
G	Start printer	h	Halt printer at end of current job
L	Impose lower limit on job size	U	Impose upper limit on job size
V	Edit device name for printer	H	Halt printer at once
N	Add a new printer	n	Reject alignment page
o	Switch to job queue	P	Change printer name
q	Quit out of spq	X	View system error log file
r	Restart current job	y	approve alignment page
=	Save program options	,	Reset printer list display format
^	Search forward by contents of a string field	\	Search backward by contents of a string field
?	display help message		

The ctrl-f, *j* etc commands also operate here. Often the user will set the size of this section of the screen to display all the printers by entering *spq* with the *-P* option.

7.1.7.1 Stop/start commands

Obviously the *G* command will only have any useful effect on a halted printer, and the *h* and *H* commands will only affect a running printer.

If an alignment file is specified for the given form type (see the Printer Definitions chapter 4 for more details) then the marker *N/A* will appear on the right. If there are one or more jobs to be selected, then the printer will output the contents of the alignment file and switch to *a/w oper* mode to enable the alignment to be approved (by typing *y*) or retried (by typing *n*). If the job has the mail or write attention flags set, then the owner will be sent an appropriate message. If there are no jobs to be selected, the printer will move to *idle* mode (unless there is an error). You can type a *y* if you are confident that the paper is correctly aligned; however you will be asked for confirmation.

An idle printer can have the requirement for alignment reinstated, if applicable, by typing *n*. The *N/A* marker will reappear, and the next job will invoke the *a/w oper* procedure.

The difference between *h* and *H*, is that *h* waits until the end of the currently printing job if any.

Note that the *H* and *h* commands do not always cause an immediate state change; they initiate a state change which may take some time to complete. So, in order to avoid confusion, the *H* command will mark the printer's line with "AB!!" (for abort) so you know it really is going to stop.

Similarly, the *h* command will mark the printer's line "(heoj)" (for "halt at end of job") while it is waiting for the job to finish.

If there is a large *docend*, *sufend* and/or *halt* string specified in the setup file, there may be some wait before the printer status changes from *shutdown* to *halted*. However you may force the issue by pressing *H* again, if you wish. The filter process group will be killed with a *SIGTERM* signal.

Beware of the fact that printers may send X-OFF characters when put off-line, and the reason why the `docend` or other string is being delayed may be because the port is waiting for an X-ON. Whilst `H` should nevertheless clear it, it is much better to only take it off-line after it has halted.

7.1.7.2 Printer size limits

You may select lower and upper limits for sizes of jobs to be printed on a given printer by typing `L` and `U` respectively. The sizes are in bytes, and a value of zero indicates no limit in either case.

If a current lower limit of job size applies on a printer, then a `<` will appear after the state in the display. If an upper limit applies, then a `>` will appear.

Jobs will only be selected for printing if the size in bytes falls between the limits set. The size is that for the whole job, disregarding any page ranges or odd/even page selection.

7.1.7.3 Restarting halted jobs

It is possible to “interrupt” a job while it is printing, then re-start it later from the last page printed.

This capability is especially useful when a small “rush job” arrives while a big job is printing. It is possible to interrupt the big job, print the rush job, then re-start the big job.

The `H` command may halt a printer in mid-job. If this happens, **Xi-Text** will note the page at which printing stopped. To find out at where it stopped, either use the `O` (other options) command in the job list, or look at the file with `I`. If the job was halted, the `O` screen will display a field “Last print halted at” followed by the page number at which it was halted. When viewing the job with `I`, the “halted at” page is noted by a highlighted message at the top of the screen.

If you re-start the printer, the job may continue from the “halted at” page. If, however, a higher priority job has appeared on the queue, this job will instead be printed, and the interrupted job will be printed when it again appears at the head of the queue.

To make interrupting jobs easier, the `!` command exists; this is equivalent to `H` followed by `G` on the same printer. This enables you to “rush through” a job interrupting a long job. Be careful; **Xi-Text** is not omniscient about interrupting escape sequences and graphics.

If you wish, you may update the “halted at” page while the job is halted, to re-start it at an earlier or later page.

If the job was assigned to a specific printer by the user, it will stay assigned to that printer. If, on the other hand, **Xi-Text** had assigned it to a printer automatically, it will be un-assigned and will start to print on the next available printer, with the correct form type loaded, when it gets to the head of the queue.

Again, beware of large printer buffers; these can limit the usability of this otherwise very useful facility.

7.1.7.4 Abort/restart current job

The abort command `A` is similar to that in the job queue, but no confirmation is asked for unprinted jobs. Note, the next job will be selected - it is usually better to halt the printer.

The restart command `r` causes the current print job to be restarted from the beginning. The action may be controlled for each printer by setting the `restart` string or command for the printer,

Both of these commands are limited in their effectiveness if the printer has a large buffer.

7.1.7.5 Change printer name

The command `P` moves the cursor to the printer name field of the selected printer, which must be halted.

The printer name may then be changed. As with other string-valued options, you may use spaces to put up various possibilities, and a `?` to list them. These are all derived from directory names in the directory `/usr/spool/printers`, or from a locally supplied help file.

7.1.7.6 Change form type

The command `f` moves the cursor to the form type field of the selected printer, which must be halted.

The form type may then be entered. Again, as with other string-valued options, you may use spaces to put up various possibilities, and a `?` to list them. The possibilities are derived from form types in the job queue starting with the first job, followed by the names of setup files in the directory for the current printer. Alternatively a local help file (see page 300) may be displayed.

If you want to avoid erroneously displaying or offering other files, such as record or alignment files at this point, these files must have names commencing with a `."` or a `"-`.

7.1.7.7 Delete printer

This command removes a printer from the list. It may only be executed on a halted printer.

7.1.7.8 Add a printer

The printer name, port, form type and description are accepted in turn.

Note that in this case the `escape` key aborts the whole operation, not just the field being edited as with other string-valued commands.

When putting the printer name in, as with the `P` command, spaces cycle through the names of directories in `/usr/spool/printers`.

When entering the line or device name, space cycles through currently unused (by **Xi-Text**) entries in directory `/dev` which start with `tty` or `lp` (but excluding `/dev/tty`), which have read-write access to user `/usr/spool/spd`.

When putting the form type in, spaces cycle through setup file names in the directory for the printer type just entered.

The class code assigned to the printer will be that obtained from the user's default class code and the `-C` option, if specified.

7.1.7.8.1 FIFOs as printers

When adding a printer, a “line” or device name starting with a `/` is not interpreted relative to the `/dev` directory. A FIFO or named pipe may be given instead of a character special device.

If the environment variable `FIFODIR` is defined when the scheduler `spshed` is started, then the contents will be prepended to line file names starting with `/`.

7.1.7.8.2 Terminal servers as printers

The line or device name is not used by **Xi-Text**, but can be used in the printer setup files. Typically a port number or unique identifier for the printer would be used. This port number is enclosed in angle brackets `<>`. The trailing `>` may be omitted when entering this but is inserted when displayed.

Note that the “guess” command doesn’t work here.

Note that if you don’t put the `<>`s in when you should, or do when you shouldn’t, the error will be detected when the printer is started and the state error will appear. However an appropriate message will be put in the system error message file `/usr/spool/spd/spshed_reps`, which you can display with the `X` command.

7.1.7.9 Reset class code

The `C` command causes a line to be displayed with the current class code. The user can then reset the code to a new value. However if the user does not have the override class privilege, this will be bitwise *anded* with their default class. A class code of zero (possibly as a result of this *and* operation) is an error and will not be allowed.

7.1.7.10 Operator attention commands

The commands `y` and `n` may be entered when the printer is in `a/w oper` status, as a result of printing an alignment page, or at the start of a single sheet. You can distinguish the two cases by the fact that `N/A` appears at the right of the line to indicate that the paper is not aligned.

If printing an alignment page, `y` accepts the page, and continues with the first job, whilst `n` causes another alignment page to be printed. The `a/w oper` state will not be re-entered unless the printer is halted and restarted, which should be done if the paper is wrecked.

In single job mode, both `y` and `n` have the same effect.

These commands may also be used when the printer is idle to bypass the requirement for alignment after the printer has been started, or to reinstate it. In either case confirmation is requested, type `Y` or `N` (for yes or no this time) as appropriate.

7.1.8 Reassign format of printer list display

The format of the printer list on the main screen may be adjusted and the result saved to suit the user's taste.

The initial layout of the printer list display is read from the message file, and this option will, if desired, save the desired format in a new message file.

To select the display format, type a “,” character. The display will change to something like:

```

" "
      14  p  Printer name
" "
      10  d  Device (or network address)
" "
      16  f  Form type

```

The numbers refer to field widths, the code letters give the desired field and the following text gives an explanation. Material in quotation marks designate separator characters. Fields prefixed by a < character denote fields which will be left-shifted to overwrite the previous field if a specific instance is too large to fit. There is a similar > character which denotes “skip any fields to the right if too large”; this is mainly useful for printer “feedback” messages.

To adjust the formats, move to the relevant row and type:

Command	Meaning
<i>q</i>	Quit and prompt to save
<i>I</i>	Insert new field before current
<i>a</i>	Insert new field after current
<i>'</i>	Insert new separator before current
<i>"</i>	Insert new separator after current
<i>D</i>	Delete field or separator
<i>w</i>	Reset width
<i>c</i>	Reset format code
<i><</i>	Toggle left overwrite field
<i>></i>	Toggle right overwrite field
<i>S</i>	Reset separator text

When a separator is entered, there is a “minimum” of a single space which will be assumed if a null string is given.

The following format codes may be given:

Code	Meaning
<i>a</i>	Signify "abort pending"
<i>c</i>	Class code
<i>d</i>	Device or network address
<i>e</i>	Description
<i>f</i>	Form type
<i>h</i>	Signify "halting at end of job"
<i>i</i>	Process id (only on current machine)
<i>j</i>	Job number being printed
<i>l</i>	Signify "local only"
<i>m</i>	"Feedback" message if off-line/error
<i>n</i>	Signify "non-aligned"
<i>p</i>	Printer name
<i>s</i>	State with <i>:message</i>
<i>t</i>	State without message
<i>u</i>	User owning job being printed
<i>w</i>	Combined abort / non-aligned message
<i>x</i>	Limit(s) set, display as < and/or >
<i>y</i>	Minimum job size
<i>z</i>	Maximum job size

When you have finished, type *q* to return. You will be prompted as to whether you want to save the formats. If you do, you can opt to save in your home directory or the current directory. You would choose the former if you want to select this every time you use *spq* and the latter if you wanted it only from the given directory.

A "header" line to the printer list section consisting of only a single letter *p* will be replaced by an appropriate header consisting of headings appropriate to the format selected.

7.2 spuser – user administration tool

The *spuser* command has 3 functions:

1. To enable any user to inquire about their permissions and charges.
2. To let a suitably privileged user change their own default form type or default priority within given upper and lower bounds.
3. To allow a fully privileged user to change system defaults and user details.

This program adjusts the user control file *spufile23* in the spool directory, by default */usr/spool/spd*.

7.2.1 spuser enquiry

The output of this is on the standard output, and is based on the user id of the requesting user. For a non-privileged user, this might take the form:

```
Default form A4 form types a4
Maximum Copies 10
Priority: min 100 max 200 default 150 Current Charge: 0
Default printer `` printer types ``
Class code A-Pa-p
You may Use other printers
You may Access queue options
You may Save new default options
```

7.2.2 spuser -c

If permitted this facility enables the user to change their default priority, default printer type or default form type. The current values are displayed on the screen in the following format:

```
Parameters for user: jmc
Default Priority: 150
Default form type: standard
Default printer type:
Priority: min 100 max 200 sys default 150
```

The user may type:

Command	Action
<i>ctrl-l</i>	Redraw the screen
<i>?</i>	Display help
<i>q</i>	Quit
<i>=</i>	Save program options
<i>d</i>	Change the default priority
<i>f</i>	Change the default form type.
<i>p</i>	Change the default printer type

The default priority may only be changed within the limits of the minimum and maximum priorities.

7.2.3 spuser -i

This option causes a full-screen display of all users and their privileges and limits to be shown. The display is in the order of user id (the numeric code used internally by Unix), unless the `-u` option is additionally given, whereupon the user names will be displayed in alphabetic order.

As with `spq` the display of help and error messages may be adjusted with the options `-b`, `-B`, `-m` and `-M`. Likewise the options may be saved in the current or home directories using the `=` command.

This mode of operation may only be entered by a user with the privilege to *Edit Admin File*. This privilege may only be set by someone else with this privilege; when **Xi-Text** is initially installed this privilege is granted to the users `root` and `spooler`. Beware of the fact that this can be used to deprive any or all users of the privilege, preventing re-entry to this mode of operation. However you will be able to carry on provided that you do not leave the program.

If you accidentally deprive all users of the “Edit Admin File” privilege, the only way to repair the situation is to stop the spooler (if no one can do that using `sstop`, then by brute force), delete the file `spufile23`, in the spool directory, by default `/usr/spool/spd` and restart. However all user and default details will be reinitialised to the installation defaults.

The format of the screen displayed is as follows:

```

User      def min max cps form          ptr          S/Priv

DEFAULT  150 100 200 50  a4

root      150 100 200 50  a4          >DP
daemon    150 100 200 50  a4
bin       150 100 200 50  a4
sys       150 100 200 50  a4
adm       150 100 200 50  a4
uucp     150 100 200 50  a4
muucp    150 100 200 50  a4
nuucp    150 100 200 50  a4
news     150 100 200 50  a4
spooler  150 100 200 50  a4          >DP
talk     150 100 200 50  a4
batch    150 100 200 50  a4
sync     150 100 200 50  a4
lp       150 100 200 50  a4
listen   150 100 200 50  a4
jmc      150 100 200 50  a4          >DP
dave     150 100 200 50  a4          >DP
susan    150 100 200 50  a4          N/S P
guest    150 100 200 50  a4          matrix <DP
accts    150 100 200 50  invoice

-- 3 more below --
Xi-Text spuser Copyright Xi Software Ltd 2000 (? for help)

```

After the user name is given the default, minimum and maximum priorities, the maximum number of copies, the current form type, printer type, the class code and an indication of the privileges. This is either blank to denote that the user has the default privileges, or has the marks “<DP”, “>DP” or “!=DP” to indicate that the privileges are respectively less extensive, more extensive, or different (less extensive in some respects and more in others) than the default.

The first row displayed gives the system-wide defaults and subsequent rows are in order of increasing numeric user ID or user name depending on whether or not the `-u` option is in effect. You can page through the user list using the cursor movement and scroll commands as for `spq` and set new values using:

Command	Action
<i>?</i>	display help
<i>c</i>	display current charge for user
<i>d</i>	change default priority for user
<i>D</i>	change system default priority
<i>l</i>	change minimum priority for user
<i>L</i>	change default minimum priority
<i>u</i>	change maximum priority for user
<i>U</i>	change default maximum priority
<i>f</i>	change default form type for user
<i>F</i>	change system default form type
<i>o</i>	change default printer type for user
<i>O</i>	change system default printer type
<i>r</i>	change allowed form type (pattern) for user
<i>R</i>	change system default allowed form type (pattern) for user
<i>g</i>	change allowed printer type (pattern) for user
<i>G</i>	change system default allowed printer type (pattern)
<i>n</i>	change maximum number of copies
<i>N</i>	change default maximum copies
<i>A</i>	copy system default priorities and form type to every user
<i>a</i>	copy system default priorities and form type to this user
<i>p</i>	change privileges for user using a further screen
<i>P</i>	change default privileges using a further screen
<i>q</i>	quit out of <code>spuser</code>
<i>=</i>	save program options
<i>s</i>	change class code for user
<i>S</i>	change default class code

The `DEFAULT` row will always stay in place as you scroll through the various users.

Type `q` to exit from the program.

7.2.3.1 Resetting system-wide defaults

When setting up the system for the first time, you should set the default privileges (upper case commands) and then set the privileges of users who are to differ in any way from the default. Note the commands to copy what you set to all other existing users.

The defaults are important not only because it is easier to set them up initially, but so that any new users are automatically added with the correct default privileges. Such new users are noticed and the details silently added to the user control file as they are encountered.

7.2.3.2 Setting default privileges

The privileges are not copied using the **A** command, but are set up separately. Note that the S/Priv column marks users whose privileges are different in some way from the default set, possibly by being less privileged.

First type the **P** command to enter the default privileges screen, which will look like this:

```

Default privileges
Default is to allow users to Edit admin file: No
Default is to allow users to Override class: No
Default is to allow users to Can masquerade as other users: No
Default is to allow users to Stop spooler: No
Default is to allow users to Use other forms: No
Default is to allow users to User other printers: No
Default is to allow users to Change priority on Q: No
Default is to allow users to Edit other users' jobs: No
Default is to allow users to Select printer list: No
Default is to allow users to Halt/restart printers: No
Default is to allow users to Add/delete printers: No
Default is to allow users to Set any priority on Q: No
Default is to allow users to Change own default prio/form: No
Default is to allow users to Unqueue jobs: No
Default is to allow users to View other users' jobs: No
Default is to allow users to Edit remote jobs: No
Default is to allow users to Edit remote printers: No
Default is to allow users to Access queue options: Yes
Default is to allow users to Save new default options: Yes

```

You can type

Command	Action
<i>y</i>	to allow a privilege
<i>n</i>	to disallow a privilege
<i>Enter</i>	to leave one as it is
<i>k</i> or <i>up cursor</i>	to move up
<i>j</i> or <i>down cursor</i>	to move down
<i>B</i>	to move to the beginning of the list
<i>E</i>	to move to the end of the list
<i>q</i>	to quit back to the main screen
<i>?</i>	to get a help screen

Note that changing some privileges always changes other ones, for example there is no point in allowing a user to halt printers but not to select the printer list, so a Yes will pop up in the latter row if you allow the former.

Likewise if you can edit the admin. file (what you are doing now), you can do anything anyway, so a Yes will pop up everywhere if you set this.

When you quit or finish the last privilege, you will be prompted as to whether you want to copy the default privileges to all users other than yourself. Use this facility carefully. It is possible to remove essential permissions from everybody including write administration file privilege. If you do this, you will have to stop the scheduler, possibly by brute force, and then delete the user control file `/usr/spool/spd/spufile23`.

7.2.3.3 Setting default class

To set the default class code, type `S`. The cursor moves to the class code field in the DEFAULT row, at the position corresponding to the letter `A`. The corresponding letter appears for any class code bit that is set, otherwise a "." is displayed.

The commands to set bits in the class code are as follows:

Command	Function
<code>+</code>	Turn on corresponding bit
<code>-</code>	Turn off corresponding bit
<code>!</code> or <code>~</code>	Toggle corresponding bit
<i>cursor left</i>	Move left
<i>cursor right</i>	Move right
<i>Enter</i>	Quit back to main screen
<code>A</code> to <code>P</code> and <code>a</code> to <code>p</code>	Move to bit and set it.
<code>&</code>	Clear all bits (but you must set some on before leaving)
<code>*</code>	Set all bits

When you have finished setting the default class, you will be prompted as to whether you want to copy to all users, however this time you will be included in the copy, as it is not as catastrophic as depriving yourself of the edit admin file privilege would be!

7.2.3.4 Setting individual user privileges

To set individual privileges, move the cursor to the relevant user, and use the `p` command. The screen shown is almost identical to that for the default privileges thus:

```
Privileges for user jmc

User jmc is allowed to Edit admin file:  No
User jmc is allowed to Override class:  No
User jmc is allowed to Can masquerade as other users:  No
```

```
User jmc is allowed to Stop spooler: No
User jmc is allowed to Use other forms: No
User jmc is allowed to Use other printers: Yes
User jmc is allowed to Change priority on Q: No
User jmc is allowed to Edit other users' jobs: No
User jmc is allowed to Select printer list: No
User jmc is allowed to Halt/restart printers: No
User jmc is allowed to Add/delete printers: No
User jmc is allowed to Set any priority on Q: No
User jmc is allowed to Change own default prio/form: No
User jmc is allowed to Unqueue jobs: No
User jmc is allowed to View other users' jobs: No
User jmc is allowed to Edit remote jobs: No
User jmc is allowed to Edit remote printers: No
User jmc is allowed to Access queue options: Yes
User jmc is allowed to Save new default options: Yes
```

The commands to set or unset privileges are the same as for the default privileges, with the addition of the command:

D - set to default privileges

However on exit, there is no prompt to copy before returning to the main screen.

Chapter 8

X/Motif Programs

8.1 xmspq – Motif GUI Spool Queue Tool

`xmspq` is a fully interactive Motif alternative to the standard print queue manager, `spq`. As with `spq` the format of the screen display, the help messages and even the command keystrokes can be easily altered to suit your requirements.

Unlike `spq`, there are no specific command line options to `xmspq`. The facility to change or specify resource settings for an X11 (and hence Motif) program on the command line can be used.

8.1.1 Options

The option `-xrm "str"` may be used to specify a resource `str` to override the resource given in the resource file.

Do not worry about the terminology it is sufficient to recognise what these resources look like. For example the resource which specified the title for the main window of `xmspq` is:

```
xmspq.title: xmspq - Xi-Text Spool Queue Viewer
```

The text before the colon “:” is the resource name (i.e. `xmspq.title`) and the text after is the data, in this case a string. The data can be boolean, integer or a string.

8.1.2 Useful Resources

All of the resources for program `xmspq` start with “`xmspq`” To keep the initial column width to manageable proportions just the text from the “.” or “*” is listed.

```
xmspq.Title
```

Specifies the title as it will appear on the main window of `xmspq`.

```
xmspq.IconName
```

Sets the text for the icon when the copy of `xmspq` is iconised.

```
xmspq.localOnly:
```

This is a boolean with the values True or False. When True the display will be restricted to show only local printers and jobs. When False printers and jobs on all co-operating **Xi-Text** hosts will be shown.

```
xmspq.unPrintedOnly:
```

Selects whether to show all print jobs, just those which have been printed or those awaiting printing. The selection is indicated by:

- 0 All jobs
- 1 Unprinted jobs
- 2 Printed Jobs

```
xmspq.inclJobs:
```

Selects which print jobs to show when the view is being restricted by printer, the options are:

- 0 Show only jobs which match the printer restriction, excluding those which do not have a specified printer.
- 1 Show only jobs which match the printer restriction, including those which do not have a specified printer.
- 2 Show all jobs disregarding any printer restriction

```
xmspq.onlyPrinter:
```

Limit display to the specified set of printers, and the print jobs associated with them. This includes jobs which, by having no printer name included, may come out on one of the set.

The set of printers may be one name, a list of names or a list of patterns for matching printer names. They may be given as a comma-separated list of alternatives, including the use of shell-style wild-cards. For example

```
Ljet
Lp1,ptr3,laser7
Ptr*,lp[1-9]
```

The wild-card options are:

- * Matches anything
- ? Matches one character
- [a-m] Matches one character in list or range
- [!n-z] Matches one char. not in list or range

```
xmspq*jlist.visibleItemCount:
```

An integer specifying the number of lines in the job list pane of the Main Window.

```
xmspq*plist.visibleItemCount:
```

An integer specifying number of lines in the printer list pane of the Main Window.

8.1.3 Examples

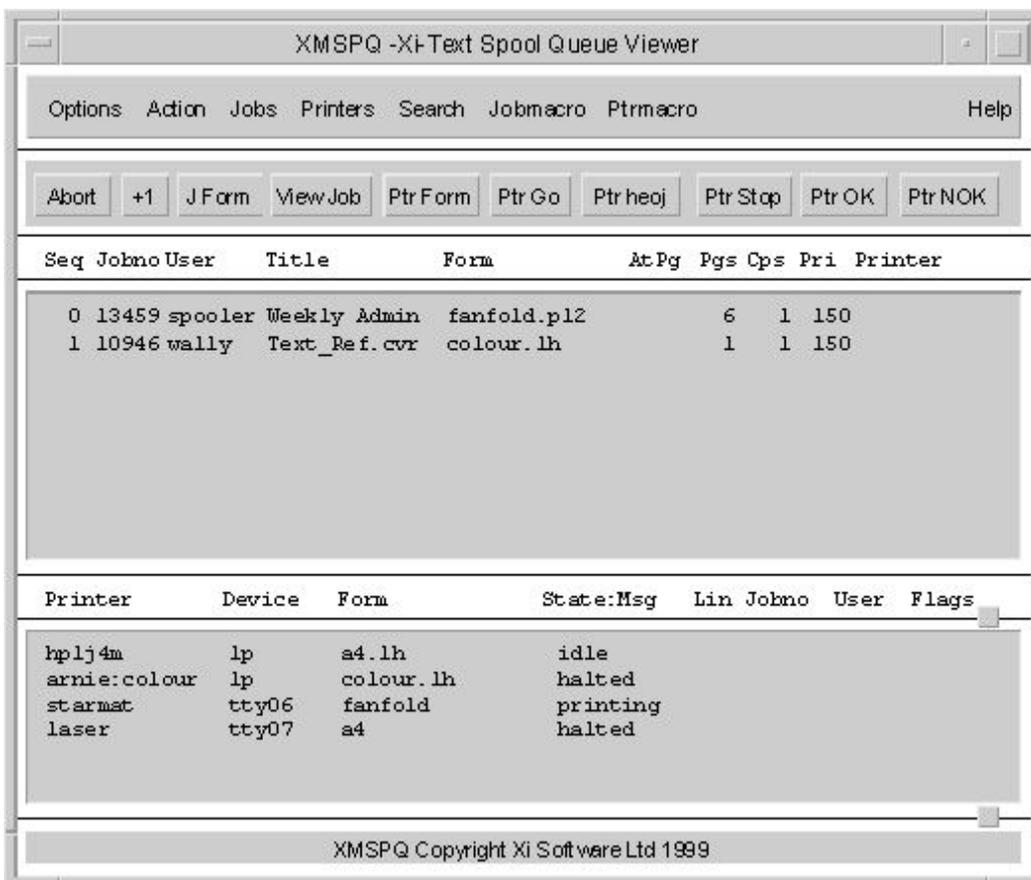
The display can be restricted to a particular user or set of users and the Window title changed to act as a reminder. One `-xrm` option is used to set the title and another is used to restrict the view by user:

```
xmispq -xrm "xmispq.title: Tony\'s Print Jobs" \  
-xrm "xmispq.onlyUser: tony" &
```

In this example the user is `tony` and the Window title is "Tony's Print Jobs". The command has been broken over two lines by preceding the newline character with a back slash. It is not uncommon to use several lines if specifying `-xrm` options.

8.1.4 The Main Window

When `xmispq` is invoked the main window will be displayed. By default it will look something like this:



The main screen is divided into two key functional areas. The top area contains menus and short cut buttons for issuing commands. The bottom area displays the print jobs and printers, which may be selected to have commands performed upon them.

The key features of the print jobs and printers area are:

Job List: Displays a re-definable view of the print jobs currently on the queue.

Scroll Bar for moving through list. Appears if there are more lines than can be displayed.

Seq	Jobno	User	Title	Form	AtPg	Pgs	Cps	Pri
0	13455	zole	Weekly_Admin	fanfold.pl2		6	1	150
1	10946	wally	Text_Ref.cvr	colour.lh		1	1	150
2	13470	jmc	passwd	A4.pl2		6	1	150
3	13471	spoole	Weekly-Admin2	fanfold.pl2		6	1	150
4	13473	spoole	Weekly-Admin3	fanfold.pl2		6	1	150
5	13477	spoole	Weekly-Admin4	fanfold.pl2		6	1	150
6	13490	spoole	Weekly-Admin5	fanfold.pl2		6	1	150
7	13492	spoole	Weekly-Admin6	fanfold.pl2		6	1	150

Printer	Device	Form	State:Msg	Lin	Jobno	User
hplj4m	lp	a4.lh	idle			
arnie:colour	lp	colour.lh	halted			
starmat	tty06	fanfold	printing			
laser	tty07	a4	halted			

X MSPQ Copyright Xi Software Ltd 1999

Control handles for adjusting the relative sizes of the job and printer lists

Printer List: Displays a re-definable view of the printers currently managed by the system

Printers and jobs can be operated upon by the appropriate menu options. The printer or job must first be selected. This is done by clicking on the line identifying it using the mouse. Once selected the line will be highlighted.

If you cannot see the printer or job that you want then you may:

- use the scroll bar or search menu options to find it
- change the view to add the item or remove unwanted items from the display.

The key features of the top area are:

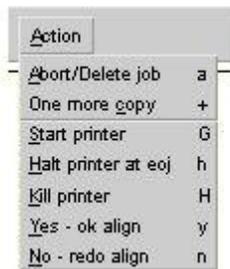
Menu bar with pull down menus providing access to all interactive queue management functions

X MSPQ -Xi-Text Spool Queue Viewer

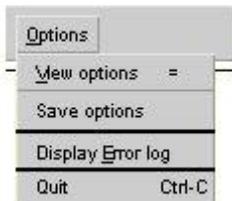
Options Action Jobs Printers Search Jobmacro Ptrmacro Help

Abort +1 J Form View Job Ptr Form Ptr Go Ptr heoj Ptr Stop Ptr OK Ptr NOK

Buttons providing short cuts to some of the more popular menu options



8.1.4.1 The Options Menu



For tailoring the look and feel of `xmspq`, saving the tailored settings, viewing the error log and quitting.

View options brings up the Display options dialogue, to tailor the look and feel. Pressing the = key also invokes this option.

Save options creates a local copy of the View options.

Display Error log brings up a Viewer showing any messages held in the **Xi-Text** system log file, `/usr/spool/spd/spd`

8.1.4.2 The Action Menu & Buttons

For high level actions: starting, stopping and aligning paper in printers; killing or printing another copy of a print job.

Abort/Delete job will stop the selected job if printing. Deletes it from the queue. This option is also available using the 'Abort' shortcut button.

One more copy increments the number of copies parameter for the print job. This option is also available using the '+1' button.

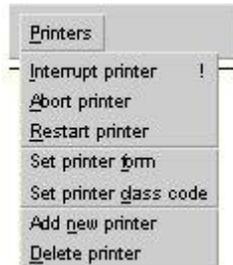
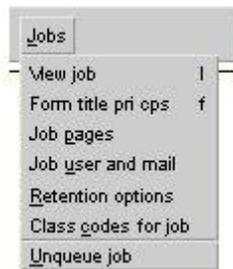
Start printer takes a printer out of the halted state, initialises it and enables printing. Any jobs pending for that printer start coming off as soon as the initialisation is complete.

Halt printer at eoj waits for the current print job (if any) to finish. It then sends any shutdown commands required and puts the printer into the halted state.

Kill printer abandons the current print job and puts the printer into the halted state.

Yes - ok align confirms that alignment is satisfactory after some test output has been produced.

No - redo align rejects the alignment. This allows adjustment of the output media and performs another alignment test.



8.1.4.3 The Jobs Menu & Buttons

Provides options for inspecting and managing print jobs which are currently visible in the job list.

View job opens a text browser showing the data to be output. No translation of the data is performed.

Form title pri cps brings up a dialogue of the jobs formtype, title, priority and number of copies.

Job pages displays the dialogue for selecting which pages are to be output.

Job user and mail opens the dialogue for setting user notification options.

Retention options brings up the dialogue to select how long a job may stay on the queue and if it is to be retained after printing.

Class codes for job shows the dialogue for setting class codes.

Unqueue job allows a print job to be taken off the queue and saved in an easy to re-submit form.

8.1.4.4 The Printers Menu & Buttons

Provides options for managing printers, including adding and removing printers from **Xi-Text** control.

Interrupt printer

Abort printer

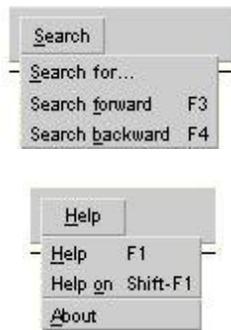
Restart printer

Set printer form brings up a selection browser to change the form type of the selected printer.

Set printer class code opens a dialogue for making changes to the class codes of the selected printer.

Add new printer sets up **Xi-Text** to use another printer.

Delete printer removes a printer from **Xi-Text** control.



8.1.4.5 The Search Menu

Both the printer and job lists may be navigated by using search options to find an item of interest.

Search for selected item or pattern.

Search forward from the current position

Search backward from the current position

8.1.4.6 Help

Context sensitive help for using [xmispq](#).

Help displays help for the current window.

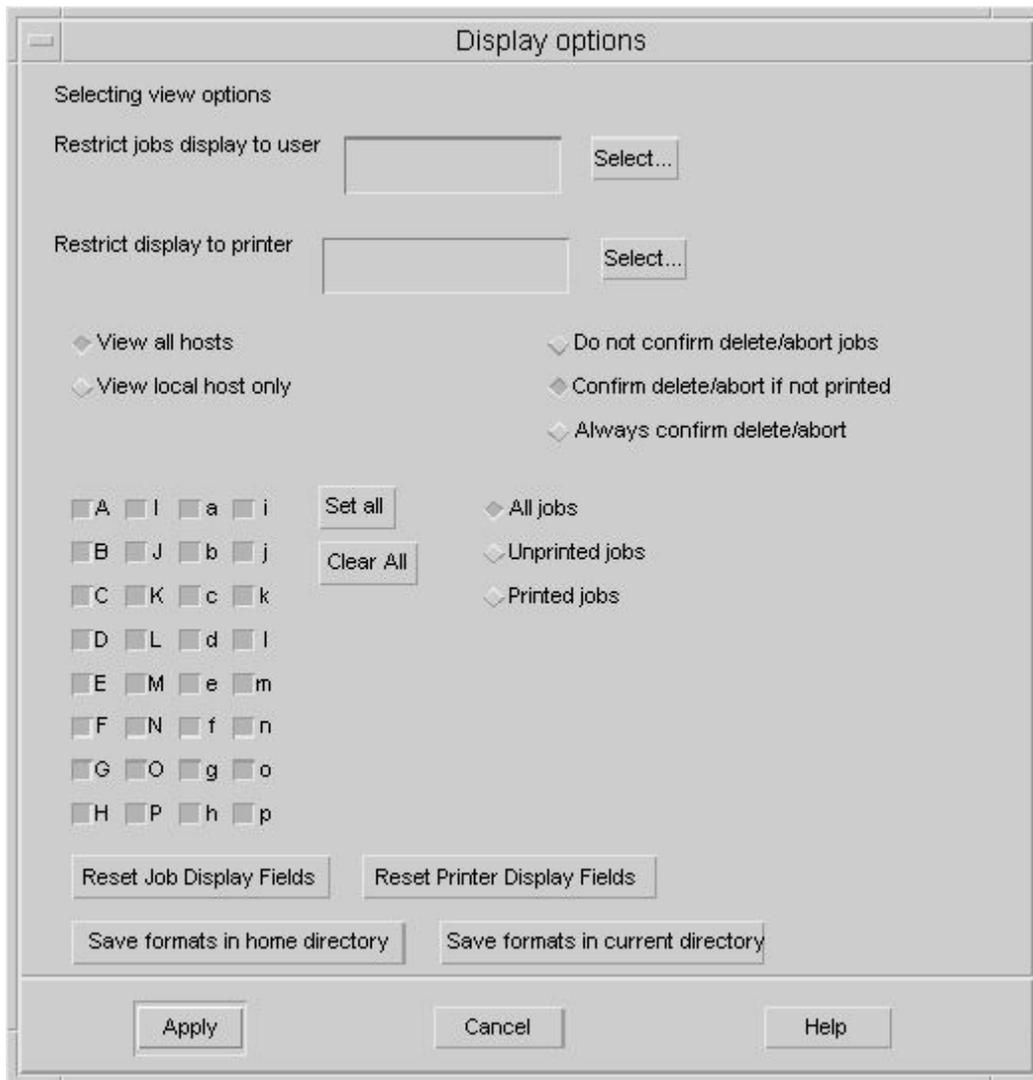
Help on changes the operating mode from taking commands to displaying help on any object (menu, button etc.) that is selected.

About displays information, such as release number, about the version of [xmispq](#) that is running.

8.1.5 Setting the View Options

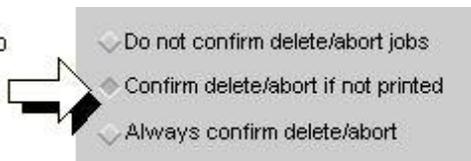
The content and format of information displayed by [xmispq](#) can be customised via the View options menu option under the Options menu. The level of confirmation for the more drastic commands may also be set under this option.

Selecting this option opens the following dialogue window.



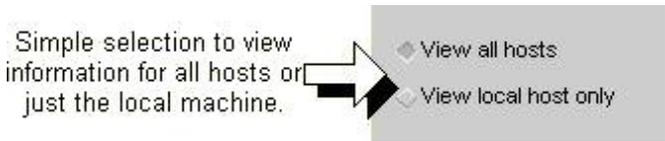
8.1.5.1 Setting the Confirmation level

Simple selection to specify level of confirmation for deleting print jobs



By default `xmspq` asks for confirmation before deleting an unprinted job from the queue. This may be relaxed to allow all jobs to be deleted without confirmation. Alternatively the rule may be tightened to ask for confirmation of all jobs before deletion even if they have been printed.

8.1.5.2 Restricting the display

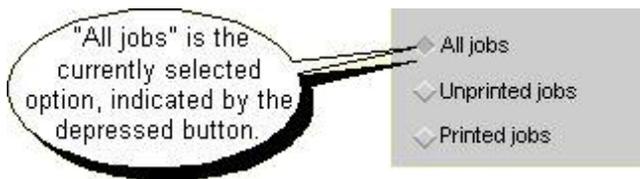


The display may be restricted by effectively filtering to only show information for selected users, printers, class codes, job status and local or all **Xi-Text** hosts.

8.1.5.2.1 Restricting the display to the local host

All hosts running **Xi-Text** in the networked mode can be treated as a single system. By default `xmspq` will show all of the externally visible jobs and printers. The view can be restricted to show just the local job queue and printers.

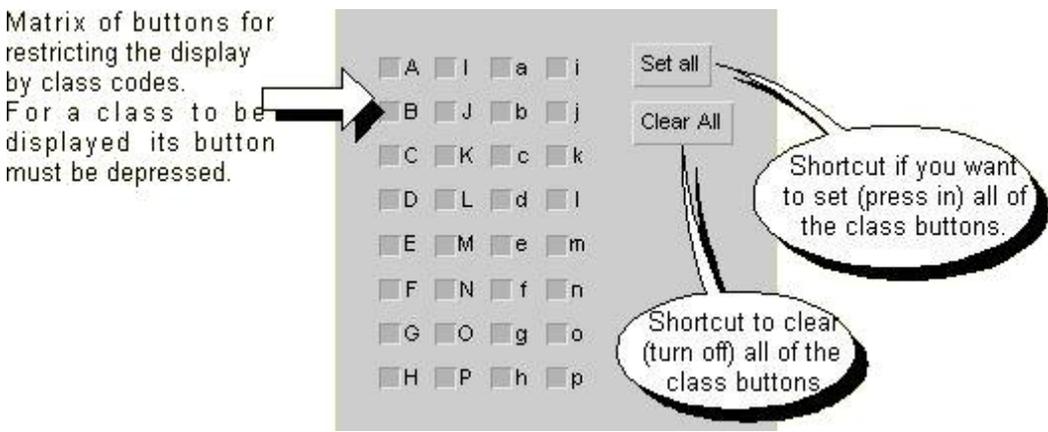
8.1.5.2.2 Restricting the display by job status



By default `xmspq` displays all print jobs. This can be restricted to show just the printed or unprinted jobs.

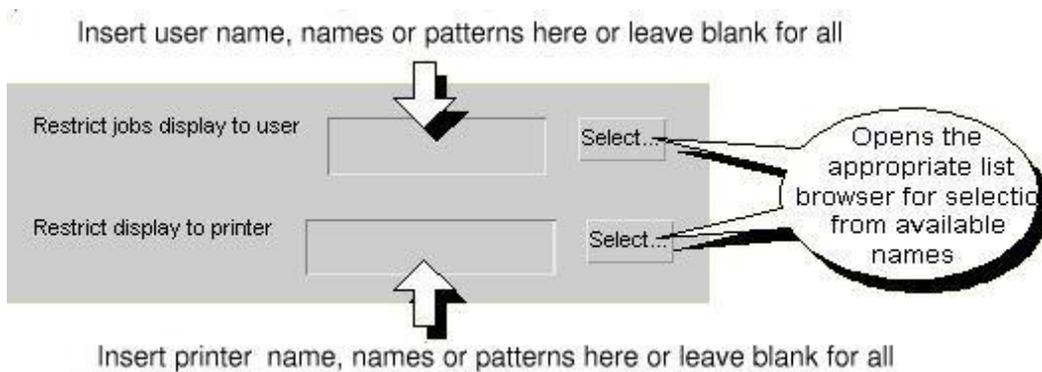
8.1.5.2.3 Restricting the display by class code

The display can be restricted by selecting a subset of class codes. Selecting just classes **A**, **B** and **C** will only show those printers and jobs which have at least one of those classes set.



For users with “Override Class” permission the Set all and Clear all buttons provide more functionality. Pressing the **Set all** button once adds all classes in the user’s set, then adds the rest if pressed a second time. Pressing **Clear All** once removes all classes not in the users set then clears everything on the second press.

8.1.5.2.3.1 Restricting the display by user & printer The display may be restricted to a user or set of users and their associated jobs. Similarly it may be restricted to one or more printers, and the jobs for those printers. When used together they restrict the display to the intersection of the sets of selected jobs and printers.



Sets of users or printers may contain just one name, a list of names or a list of patterns for matching names. The list of jobs displayed when restricting by printer name includes jobs which, by having no printer name included, may come out on one of the selected printers. There is no corollary for user names, since all jobs are tied to the user who owns them.

The printer and user names may be given as a comma-separated list of alternatives, including the use of shell-style wild-cards. For example

Users

```
Fred
jmc,tony,ukops_jmc,ukops_wal
ukops*,ukadmin[1-5]
[m-z]*
```

Printers

```
ljet
lp1,ptr3,laser7
ptr*,lp[1-9]
[a-c]?9
```

The wild-card options are:

- * Matches anything
- ? Matches one character
- [a-m] Matches one character in list or range
- [!n-z] Matches one char. not in list or range

8.1.5.3 Changing the fields displayed and their format

There is far more information available for both printers and jobs than could be displayed in the main window of xmspq. Different columns of information may be displayed as required. The field widths and handling of field overflow may also be adjusted.

For example: If your print jobs often have a hold time and long titles, and you do not need the printer column in the job display, then you could do the following:

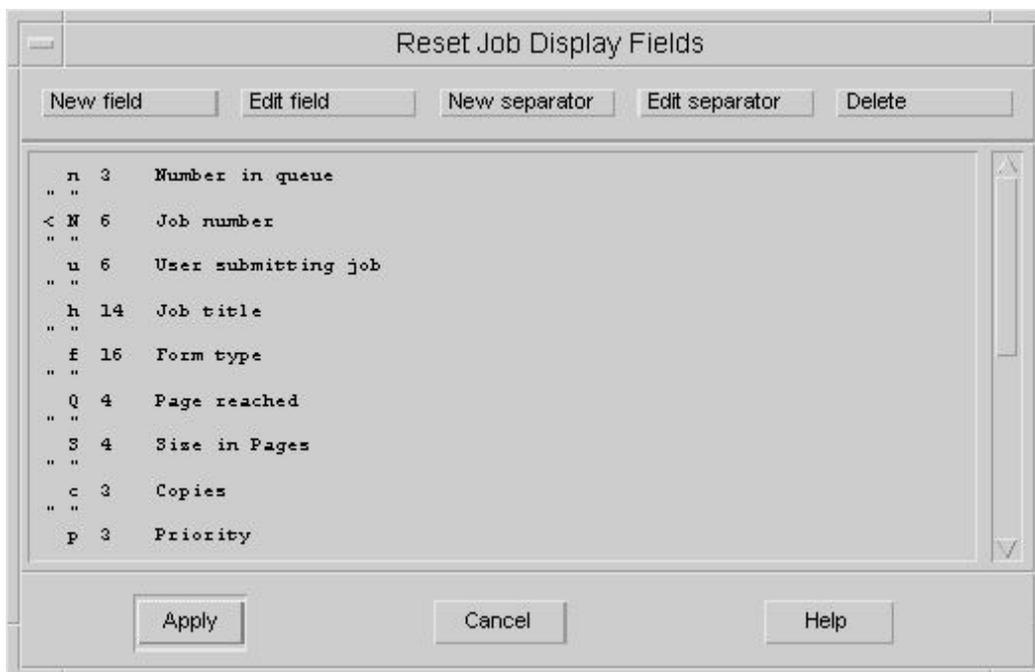
- Delete the printer column
- Make your xmspq main window wider by dragging it with the mouse.
- Add a job hold time column
- Increase the Title width from 14 to say 25 characters.



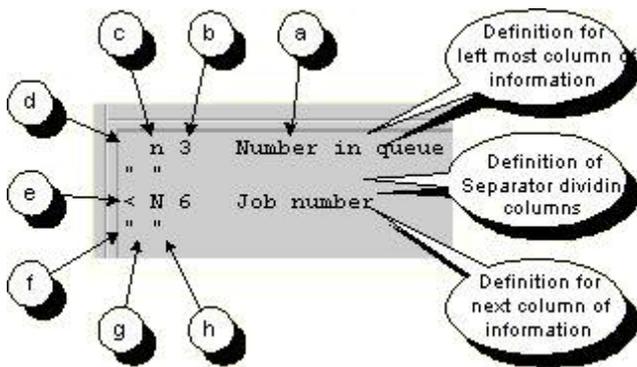
To edit the job or printer displays click on the appropriate button:

8.1.5.3.1 Changing the Job Display

Clicking on the **Reset Job Display Fields** brings up the following window. The row of buttons at the top are for adding, changing and deleting fields or separators. The fields are the columns of information and the separators are the column dividers.

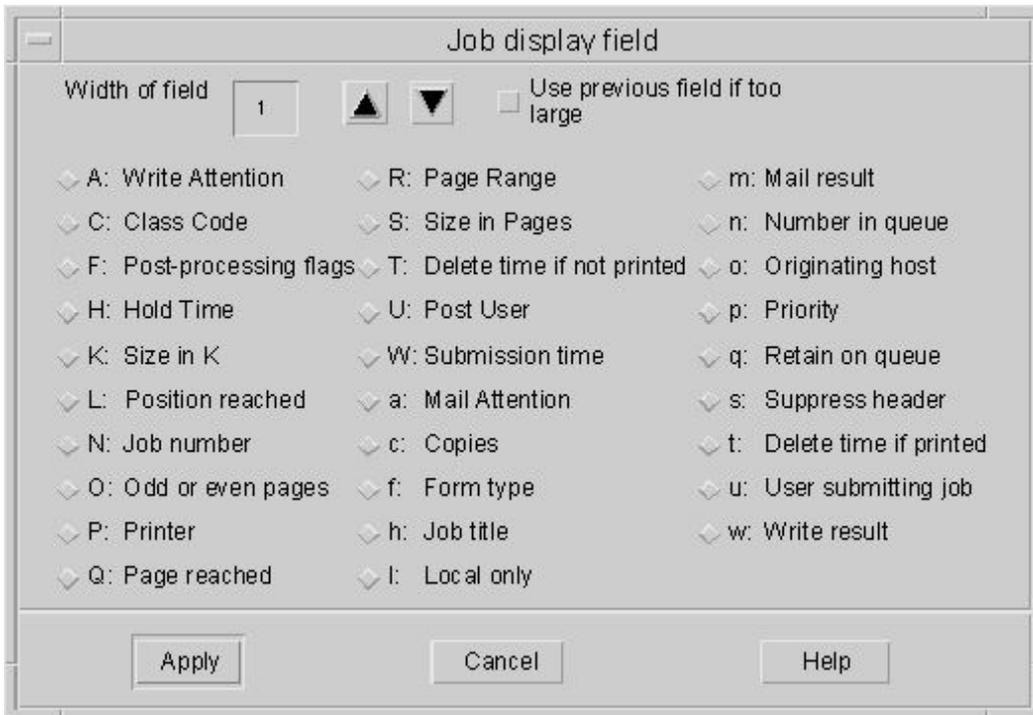


Underneath the row of buttons is a scrollable text window showing the display format. Each line holds the specification for one column or column divider, as follows:



1. Field description / title
2. Width in characters
3. Field Identifier
4. No action on field overflow
5. Overflow onto left hand field is permitted
6. Open quote before separator
7. Separator character(s)
8. Close quote after separator

To edit an existing field select the line showing the specification for that field and click on the Edit field button. To insert a new field select the line underneath the point at which you want to insert it and click on the New field button. Either of these actions will bring up a “Job display field” window looking something like this:



The width of field can be adjusted by typing in a new value or using the arrow buttons. The “Use previous field...” button is to allow a field to overflow into the field on its left. Pressing the button changes it from the deselected to selected state and vice versa.

The other buttons allow selection of what information will be displayed in the column. Only one of these can be in the selected state. When a button is selected the field width is set to a suitable value, which may then be adjusted as required.

8.1.5.3.1.1 Changing the Printer Display The principles for the printer display are the same as those for the job display, with one addition. If there is too much data in a field it may overflow into the previous field (as in the job display) or into the next field. The same windows are used, only the display information inside is different.

8.1.5.4 Saving the Format Changes

These are saved in local copies of the file `xmspq` helpfile which is given a user specified name at the time of saving. If a local help file has been set up, `xmspq` will be told to use it by the `XMSPQCONF` parameter in the local `.xitext` file.

8.1.5.5 Saving the View Options (excluding Format Changes)

All of the view options, excluding job and printer display formats, are saved in a users local copy of the resource file. This file is called `XI` and is kept in the user’s home directory. The customised options are added to the end of the file.

The view options may be overridden by entering the Motif parameters on the command line using the `-xrm` option for the window manager. For example to restrict the job display to just those belonging to users `kate` and `tom` give the option:

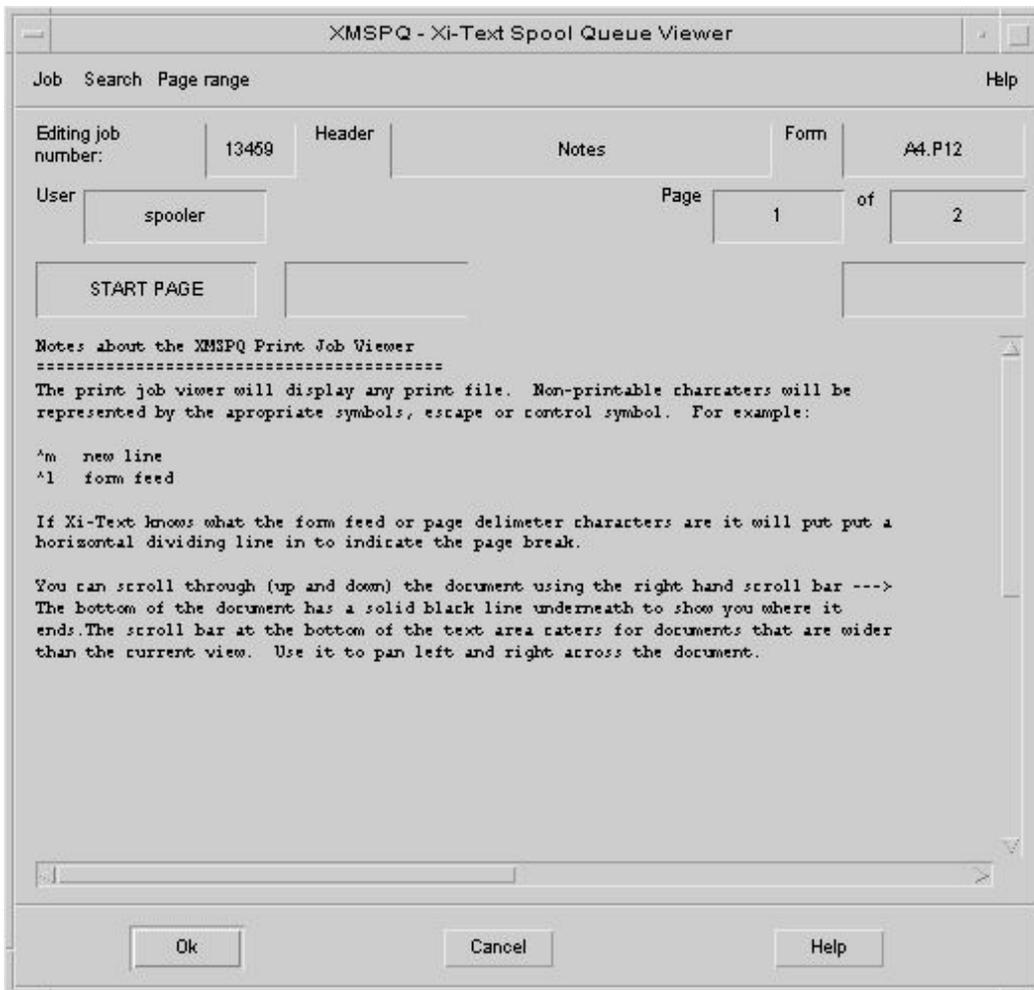
```
xmspq -xrm "xmspq.onlyUser: kate,tom"
```

The complete set of view options, held at the end of a user’s `XI` file, look like this:

```
!! xmspq User-defined options 15:02:29 98/07/10
xmspq.keepJobScroll: False
xmspq.localOnly: False
xmspq.unPrintedOnly: 0
xmspq.inclJobs: 1
xmspq.onlyClasscode: -1
xmspq.confirmAbort: Unprinted
xmspq.onlyUser:
xmspq.onlyPrinter:
xmspq.onlyTitle:
xmspq.sortPtrs: True
xmspq*jlist.width: 594
xmspq*jlist.visibleItemCount: 10
xmspq*plist.visibleItemCount: 6
```

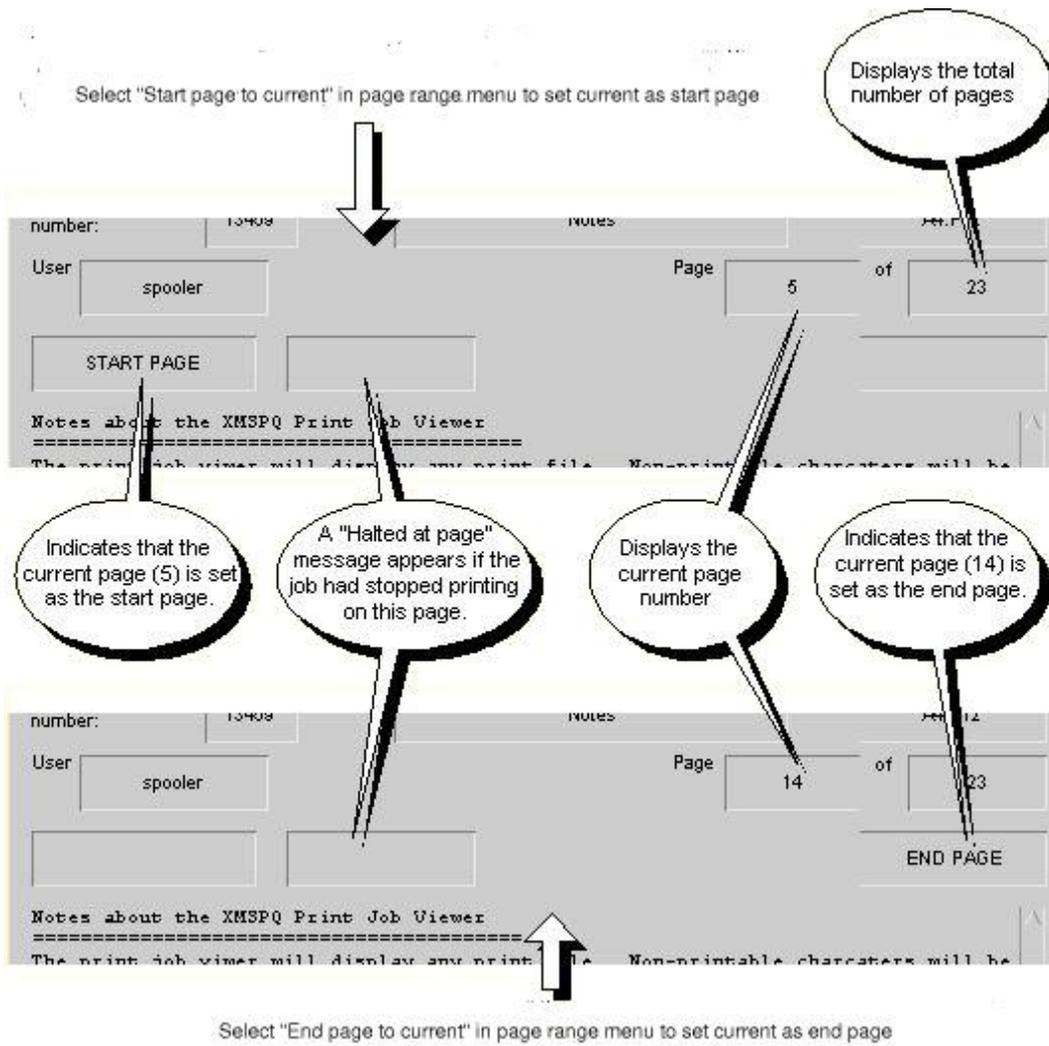
8.1.6 Viewing a Print Job

Selecting a print job from the display then selecting the **View job** option from the **Jobs** menu opens this window.



The display may be scrolled through the document and panned across the text using the vertical and horizontal scroll bars. The Search Menu provides options for specifying and finding text strings within the print job.

A selected range of pages from a document may be printed. The start and end pages may be viewed and set from within this window. If a print job has stopped during printing the page on which it halted is also shown.



8.1.7 Changing Job and Printer parameters

A job may be deleted, changed, reviewed by clicking on the line representing it in the job list and then selecting the required menu option or short cut button. Printers may operated upon in exactly the same way. Some menu options and short cut buttons will have an immediate effect. Others will open a dialogue window for additional information, such as this one for changing the form type, header, priority and number of copies for a print job.

Editing job number: 4932

Form: label

Title: j_smith Suppress header page

Printer:

Priority: 150

Copies: 1

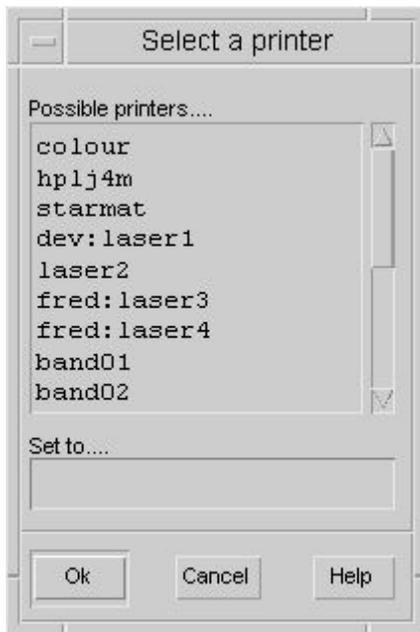
Fields like Priority and Copies take a numeric value which may be typed in or adjusted using the increment and decrement buttons.

Plain text fields like Title simply allow any free text to be entered or modified.

Square buttons represent options which can be enabled or disabled (true or false) like the Suppress header page. The Diamond shape buttons (not shown in this example) are used for selecting one option out of a set of 2 or more possible options.

Some fields like the Printer and Formtype in this example have a Select button next to them. These support two methods of parameter entry, straight text entry as in the Title field or by selecting one option from a list using a selection dialogue. (With some versions of the Motif library, this may be a "Combo Box").

Clicking on the Select printers button, for example, opens this selection dialogue.



The information displayed is context and configuration sensitive, showing only the permitted and/or appropriate information.

In this example `xmispq` is looking at all printers and jobs on a set of networked **Xi-Text** machines. `laser1` is on host `dev`, `laser3` and `laser4` are on host `fred`.

Printers which the user is not allowed to access due to class restrictions will not appear in their view of this list.

8.2 xmispuser – Motif GUI User Administration Tool

`xmispuser` is a fully interactive Motif alternative to the standard user permission manager (invoked using the command `spuser -i`). It is provided to maintain the list of user privileges, charges and default modes for jobs and printers.

Unlike `spuser` there are no command line options to `xmispuser`, it is always in the interactive mode (similar to `spuser -i`). The facility to change or specify resource settings for an X11 (and hence Motif) program on the command line can be used.

8.2.1 Options

The option `-xrm "str"` may be used to specify a resource `str` to override the resource given in the resource file.

Do not worry about the terminology it is sufficient to recognise what these resources look like. For example the resource which specified the title for the main window of `xmispuser` is:

```
xmispuser.title: xmispuser - Edit user options
```

The text before the colon “:” is the resource name (i.e. `xmspuser.title`) and the text after is the data, in this case a string. The data can be boolean, integer or a string.

8.2.2 Useful Resources

All of the resources for program `xmspuser` start with “`xmspuser`”. To keep the initial column width to manageable proportions just the text from the “.” or “*” is listed.

- `.Title:` Specifies the Title as it will appear on the main window of `xmspuser`.
- `.IconName:` Sets the text for the icon when `xmspuser` is iconised.
- `.sortAlpha:` This is a boolean with the values True or False. When True the list of users will be sorted into alphabetical order by username.

8.2.3 Examples

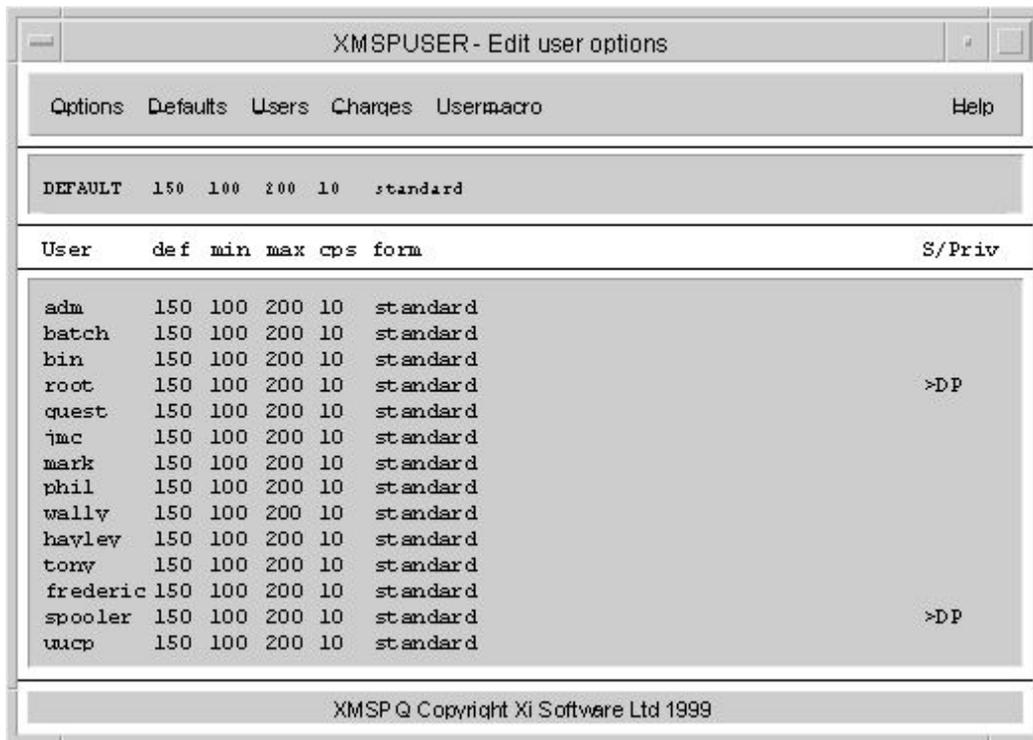
When using an X-Terminal connected to several **Xi-Text** machines it can be very useful to include the hostname in the Window and Icon titles. The display can be restricted to a particular user or set of users and the Window title changed to act as a reminder. One `-xrm` option is used to set the title and another is used to restrict the view by user:

```
xmspuser -xrm "xmspuser.title: xmspuser - $HOSTNAME" \  
-xrm "xmspuser.IconNAME: $HOSTNAME" &
```

In this example the machine’s name will be picked up from the environment variable `HOSTNAME`. It is not uncommon to use several lines if specifying `-xrm` options. Of course this can all be put in a shell “wrapper” to save typing each time.

8.2.4 The Main Window

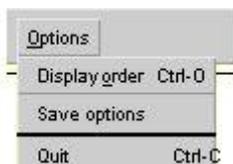
When `xmspuser` is invoked the main window will be displayed. By default it will look something like this:



The bottom area contains a list of all users in the password file with their **Xi-Text** permissions and settings. It will have a scroll bar if there are more users than can fit on the screen. Above this is a pane containing the default settings.

At the top of the screen is the menu bar supporting all of the `xmspuser` commands. Each menu option opens a dialogue or operates on the specified data immediately. With a few exceptions these are straight forward and easy to understand.

8.2.4.1 The Options Menu



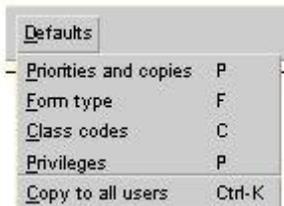
This is for changing the ordering of users in the display, saving the settings and quitting.

Display order brings up the Display options dialogue, to tailor the look and feel. Pressing the = key also invokes this option.

Save options saves a local copy of the resources file with the user's choice of Display order.

Quit saves any changes to the default and user accounts. N.B. This is the only way of saving these changes. Quitting by any other method will discard the changes.

8.2.4.2 The Defaults Menu



Administers the default account parameters.

Priorities and copies opens the dialogue for setting up maximum number of copies, priority range and defaults.

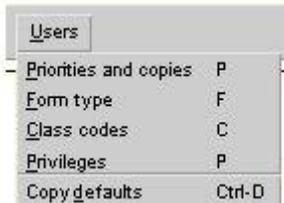
Form type opens the form type selection dialogue.

Class codes opens the class code selection dialogue.

Privileges opens a dialogue showing and allowing changes to the privileges (e.g. starting and stopping printers).

Copy to all users Copies the default settings to all users - use with care.

8.2.4.3 The Users Menu



Administers the accounts of individual users. **Priorities and copies** opens the dialogue for setting up maximum number of copies, priority range and defaults.

Form type opens the form type selection dialogue.

Class codes opens the class code selection dialogue.

Privileges opens a dialogue showing and allowing changes to the privileges (e.g. starting and stopping printers).

Copy defaults resets the users account to the default settings - use with care.

8.2.4.4 The Charges Menu



Displays and manages the printing charges.

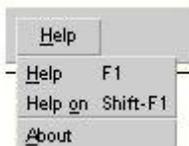
Display Charges operates for all selected users.

Zero charges for selected users

Zero charges for ALL users

Impose fee on selected users.

8.2.4.5 The Help Menu



Context sensitive help for using [xmspuser](#).

Help displays help for the current window.

Help on changes the operating mode from taking commands to displaying help on any object (menu, button etc.) that is selected.

About displays information, such as release number, about the version of xmspuser that is running.

8.2.5 Selecting multiple users for menu options

Many of the menu options can be carried out on more than one user at a time. First select all of the relevant users from the list. Then select the required menu option for the operation you wish to perform. The set of users remains selected after you have completed the operation in case there are other options that you want to use on them.

You can scroll up and down a long list of users without losing those you have selected so far. The following mechanisms allow you to select a group of users:

- Hold the selection mouse button down and drag it across a contiguous set of users.
- Using the mouse click on the first user in the required set. Move to the last user in the set, hold the shift key down on the keyboard and click on this user.
- Individual users may be added or removed from the set by clicking on them whilst holding the Control key down on the keyboard.

8.2.6 Copying defaults to all users

Available via the **Copy to all users** option under the **Defaults** menu. It copies the default settings to all users except the one you are logged in as. This command should be used with care. It is possible to remove essential permissions from everybody including write administration file privilege.

8.2.7 Resetting a user to the default

This is a very useful mechanism for setting permissions for a group of users. Set the default to the required value then apply it to the required users. If you do not want new users to inherit this default setting remember to return it to the original state.

You cannot remove administration file privilege from `root` or `spooler`.

Chapter 9

Terminal Server support commands

The following support commands for terminal servers are provided as standard with **Xi-Text**, installed in the internal programs directory, by default `/usr/spool/progs`.

9.1 Xtlpc

```
/usr/spool/progs/xtlpc [options] [file]
```

`xtlpc` is a terminal server program for use by **Xi-Text** where the destination is an LPD protocol server. Input is usually taken from standard input, or from the supplied file name.

For information on the format of the control file for `xtlpc`, please see below at page [270](#).

9.1.1 Options for xtlpc

The following command line options are available for `xtlpc`.

Note that there is no mechanism provided for saving the options in a file as with the user-level programs.

9.1.1.1 -H option

```
-H hostname
```

This option supplies the host name or IP address for the destination printer.

It is normal to pass `$SPOOLDEV` so that the host or IP address given when the printer is installed is passed. However you may want to pass `$SPOOLPTR` here if that printer name can be decoded to the IP address of the destination printer.

This option should always be given.

9.1.1.2 -f option

-f ctrlfile

This specifies the location of the control file, typically `xtlpc-ctrl` in the internal data directory, usually `/usr/spool/progs`.

This option should always be given.

9.1.1.3 -S option

-S hostname

This option sets the “sender” IP to the address given. This may be necessary in some configurations where a variety of IP addresses from the outgoing server are possible and the printer is “fussy” about which server jobs are received “from”.

9.1.1.4 -P option

-P ptrname

The LPD protocol supplies a name to the printer. By default, this is taken from `SPOOLPTR`, i.e. the name given as the name of the printer under **Xi-Text**. Some terminal servers require a specific name to be given to the printer, perhaps to select which printer or printer option of several is to be specified. This option overrides this printer name in the protocol.

9.1.1.5 -F option

-F fifo

Take input from the named fifo rather than standard input or the supplied file. This is useful in situations where the LPD protocol has to be run from a command line, i.e. a console session, and files sent to it by having **Xi-Text** write to the fifo.

9.1.1.6 -N option

-N

Avoid sending zero length files, which confuse some servers.

9.1.1.7 -U option

`-U`

Do not insist on a reserved port locally (i.e. numbered under 512).

9.1.1.8 -l option

`-l n`

Make the given number of attempts to connect to the terminal server before giving up. The default is 3.

9.1.1.9 -L option

`-L n`

Wait the given number of seconds between attempts to connect using the `-l` option. The default is 1.

9.1.1.10 -I option

`-I n`

Set the input timeout, waiting for acknowledgements from the printer, as specified. The default is 5.

9.1.1.11 -O option

`-O n`

Set the output timeout, waiting for packets to the printer to be accepted, as specified. The default is 5.

9.1.1.12 -R option

`-R n`

Retry transmission or reception the given number of times before giving up. The default is 0.

9.1.1.13 -s option

`-s n`

Set the linger time of the protocol as specified, which may be fractional. The default is 0.

9.1.2 Exit codes from xtlpc

Normal termination is denoted by an exit code of zero.

1. Usage error, invalid option etc.
2. No control file given.
3. No destination host given.
4. Cannot open input file.
5. Unknown host name.
6. Cannot open control file.
7. Syntax errors in control file.
8. No spool directory defined.
9. Cannot select spool directory.
10. Control card format not given in control file
11. Invalid format control card in control file
12. Cannot write to spool directory
13. Cannot create spool file
14. Cannot allocate file name.
15. Cannot open FIFO.
16. Name given is not a FIFO.
17. Cannot specify input and FIFO.
18. Cannot create sequence file
19. No port name in control file
20. TCP protocol missing
21. Cannot find named port in service file
22. Cannot open socket.
23. Cannot bind port
24. Cannot allocate TCP device

25. Cannot make connection
26. Cannot allocate TCP space or change stream
27. Write failure writing to socket
28. Lost spool file
29. Read failure ack packet.
30. Error return from remote.
31. Cannot determine own host name
32. Invalid host name
33. Exit on signal.

9.1.3 Format of control file

The control file sets out:

1. What is sent.
2. The protocol to send it.

In the following sections we outline these. The control file itself is a text file that mirrors the protocol described below and uses a simple interpreted format to construct the files required and transmit them using the protocol.

9.1.3.1 Description of LPD Protocol

Each LPD job consists of one or more “D-files” of job data, and a “C-file” saying what to do with it.

The C-file consists of lines of text referred to as “control cards”. The first character of the control card gives the action, such as “`f`” to print the file name following, “`H`” to set a host name etc.

The C-file is given a name starting with `cf` and followed by digits and a host name. The D-file is given a name starting with `df` and followed by digits and a host name.

The following is an example of the contents of a C-file:

Control Card	Explanation
<code>Hxisl.co.uk</code>	Set host name to xisl.co.uk
<code>Pjmc</code>	Set user's login name
<code>Jmydata</code>	Set job name
<code>Cxisl.co.uk</code>	Set class name
<code>Ljmc</code>	Set user's name for banner
<code>fdfA058xisl.co.uk</code>	Print file name given
<code>UdfA058xisl.co.uk</code>	Unlink (delete) file name
<code>N/home/jmc/mydata</code>	Name of file

Some of these constructs do not have direct equivalences in **Xi-Text**, for example the file name is not relevant, and the class name does not mean the same thing.

The form type (or equivalent) is selected by substituting some other letter for “f” in the print command. Copies are specified by duplicating the print command the appropriate number of times. There are options for indentation, and the automatic invocation of “pr” with appropriate arguments.

Each file is sent across by means of a protocol of lines of text including a single-byte code denoting the action, and the name and size of the file being sent.

9.1.3.2 Multiple copies

In principle the C-file could provide multiple copies by repeating the print command the appropriate number of times, i.e. in the above sequence the “card”

```
fdfA058xisl.co.uk
```

could be repeated the appropriate number of times.

Some printers do print multiple copies if the “card” is repeated in this way. Others ignore the contents of the C-file.

As supplied, the assumption is printers do not print multiple copies in this way and a separate LPD protocol job is created for each copy requested.

If your printer does print multiple copies in this way, you modify the control file as shown in the comments so as to repeat the “card” according to the value of the `SPOOLCPS` environment variable, which gives the number of copies left to print, and also include the keyword `onecopy` in the setup file, to make sure that the job is only sent to the printer once.

9.2 Sp.lpq

```
sp.lpq [ -l ] [ -s ] printer
```

`sp.lpq` obtains and outputs a list of pending jobs on an external print server supporting the LPD protocol.

9.2.1 Options

This is a user-level program that interfaces with the LPD interface software and the option-handling mechanism is supported.

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

9.2.1.1 `-?` or `+explain` option

```
-?  
--explain  
+explain
```

This option causes a summary of the other options to be displayed without taking further action.

9.2.1.2 `-l` option

```
-l
```

Produce a long listing.

9.2.1.3 `-s` option

```
-s
```

Produce a short listing (default).

9.2.1.4 `+freeze-current` option

```
--freeze-current  
+freeze-current
```

Save all the current options in a `.xitext` file in the current directory with keyword `SPLPQ`.

Processing of the remaining command options will proceed even if this is specified.

9.2.1.5 `+freeze-home` option

```
--freeze-home  
+freeze-home
```

Save all the current options in a `.xitext` file in the user's home directory with keyword SPLPQ.

Processing of the remaining command options will proceed even if this is specified.

9.2.2 sp.lprm

```
sp.lprm printer job ...
```

`sp.lprm` deletes pending jobs on an external print server supporting the LPD protocol. The specified printer is that known to **Xi-Text** and the details are obtained from the printer setup file.

9.3 Xilp

```
/usr/spool/progs/xilp [options] [ file ... ]
```

`xilp` is a terminal server program for use by **Xi-Text** where the destination is an LPD server.

It has been largely superseded by `xtlpc`.

9.3.1 Options

9.3.1.1 -H option

```
-H hostname
```

This is the host name or IP address to be used to access the printer. In **Xi-Text** setup files, it may be appropriate to substitute `$SPOOLDEV` or `$SPOOLPTR` (if the name of the printer is set up as a host name) here.

9.3.1.2 -p option

```
-p portnumber
```

This is the port number or service port name to be used if not the standard of 515 (printer).

9.3.1.3 -h option

```
-h
```

Enable banner page.

9.3.1.4 -r option

`-r`

Remove file on completion (inapplicable to **Xi-Text**, which sends data on standard input).

9.3.1.5 -# option

`-# n`

Set number of copies as specified. If you use this option under **Xi-Text**, you should put `$SPOOLCPS` as the argument here and put `onecopy` in the setup file, see page 70.

9.3.1.6 -u option

`-u user`

Set user name to `user`.

9.3.1.7 -P option

`-P ptr`

Set printer name to as given.

9.3.1.8 -C option

`-C class`

Set printer class as given.

9.3.1.9 -J option

`-J file`

Synonym for `-F`

9.3.1.10 -F option

`-F file`

Set file name in protocol as given.

9.4 Xtelnet

```
/usr/spool/progs/xtelnet [options]
```

`xtelnet` is a terminal server program for use by **Xi-Text** where the destination is a server which accepts “reverse telnet” protocol. Data is accepted on standard input.

9.4.1 Options

There is no facility for saving these options in a file.

9.4.1.1 -h option

`-h hostname`

This is the host name or IP address to be used to access the printer. In **Xi-Text** setup files, it may be appropriate to substitute `$SPOOLDEV` or sometimes `$SPOOLPTR` here.

9.4.1.2 -p option

`-p portnumber`

This is the port number (or name in `/etc/services`) to be used, if not 9100. In **Xi-Text** setup files, it may sometimes be appropriate to substitute `$SPOOLDEV` here where the printer on a server with multiple printers with each selected by port number is in use.

9.4.1.3 -l option

`-l n`

Make the given number of attempts to connect to the terminal server before giving up. The default is 3.

9.4.1.4 -L option

`-L n`

Wait the given number of seconds between attempts to connect using the `-l` option. The default is 1.

9.4.1.5 -t option

`-t n`

Sleep `n` seconds before exiting, after closing the connection to the server. This is sometimes necessary with some servers, as **Xi-Text** operates too fast for them. However you may prefer to use the `postclose` option in the setup file, see page 50.

9.4.1.6 -s option

`-s n`

Set the linger time of the protocol as specified, which may be fractional. The default is 0.

9.4.1.7 -d option

`-d`

Debug. Displays activities on standard error.

9.4.2 Exit codes

Normal termination is denoted by an exit code of zero.

Errors are signified as follows:

1. Device appears to be offline.
2. Some kind of device error, failure to connect etc.
3. Incorrect usage, unknown option etc.
4. System error, cannot find host name etc.

9.5 Xtlhp

```
/usr/spool/progs/xtlhp [options]
```

`xtlhp` is a terminal server program for use by **Xi-Text** where the destination is an HPNPF server. Alternatively other printer devices which accept input using “reverse telnet” and allow status to be interrogated via SNMP (or other means) may be driven.

After each block is transmitted, the status of the printer is interrogated using a supplied control script.

The SNMP are done directly, rather than relying on an external program being available.

9.5.1 Options

There are no facilities for remembering the options, which must be supplied each time.

9.5.1.1 -h option

```
-h hostname
```

This is the host name or IP address to be used to access the printer. In **Xi-Text** setup files, it may be appropriate to substitute `$SPOOLDEV` or `$SPOOLPTR` here.

9.5.1.2 -p option

```
-p port
```

This is the port number or service port name to be used, if not the standard of 9100.

9.5.1.3 -f option

```
-f configname
```

This specifies the definitions file, if not `xtsnmpdef` in the **Xi-Text** internal programs directory, usually `/usr/spool/progs`.

9.5.1.4 -c option

```
-c file
```

This specifies a control file for obtaining the printer status. See below at page [279](#) for the format of this file.

9.5.1.5 -l option

`-l logfile`

Specify error message destination if not standard error.

9.5.1.6 -b option

`-b n`

Specify the output block size. The argument may be just a number, or it may be followed by **b**, **k**, **m** or **g** to denote blocks, kilobytes, megabytes or gigabytes.

Otherwise a default block size of 10K is assumed.

An SNMP enquiry is made after each block of data of this size has been completed.

Do not confuse this with the `outbuffer` of setup files described on page 49.

9.5.1.7 -d option

`-d n`

Debug. Displays activities on standard error, if *n* is greater than zero. Higher values of *n* will give more output, including tracing of SNMP operations.

9.5.1.8 -D option

`-D n`

As for **-d**, except that no output is attempted, the control script is executed once only and the appropriate exit code returned. Use an argument of zero to just invoke the script and exit.

9.5.1.9 -c option

`-c name`

This option specifies the community in SNMP operations. The default is `public`.

9.5.1.10 -T option

`-T timeout`

This option specifies the timeout to wait for SNMP replies. The default is 1 second.

Note that this value may be fractional.

9.5.1.11 -S option

`-S snmpport`

Specify the SNMP port name to use, if not `snmp`.

9.5.2 Configuration file

The configuration file `xtsnmpdef` provides a series of mnemonics for the SNMP definition of the form:

```
system 1.3.6.1.4.1.11.2
gdStatusBytes system.3.9.1.1.1
```

The function of these definitions is to “macrogenerate” constructs which appear in the control file, for example (with appropriate definitions)

```
gdStatusPaperOut.0
```

may be used as an abbreviation for

```
1.3.6.1.4.1.11.2.3.9.1.2.8.0
```

Previous versions of `xtlhp` expected quotes around the columns of the file, but this is no longer expected and they will be discarded, provided that they are properly matched. The file may define names in terms of previously-defined names and may redefine names. If a name is redefined, names already defined using the previous definition are unaffected.

9.5.3 Control file format

The control file consists of a series of command structures involving options to obtain SNMP values, run commands, string and numeric comparisons and assignments.

Constructs are based upon that of the shell, but control names are in upper case.

9.5.3.1 Expressions

```
'string'
```

A single-quoted string is taken literally, as per the shell.

```
"string"
```

A string enclosed in double-quote characters is subject to name expansions of macros defined in the configuration file and variables assigned in the script.

```
SNMPVAR(objid)
```

The specified *objid*, which may be specified using macros, is returned, treated as a numeric value.

```
SNMPSTR(objid)
```

The specified *objid*, which may be specified using macros, is returned, treated as a string value.

```
`string`
```

A back-quoted string is a shell command to execute and the output taken as a string. Variable names are expanded within the string before execution.

```
[string]
```

A string enclosed in square brackets is a shell command to execute, with the output taken to be a decimal numeric value, the whole returning a numeric value. Variable names are expanded within the string before execution.

```
{string}
```

A string enclosed in braces is a shell command to execute, with the output assumed to be a hexadecimal value. (This is required for `snmpinfo(8)` on AIX systems, which returns hexadecimal values). Variable names are expanded within the string before execution.

```
number
```

a numeric value.

```
name
```

The value currently assigned to *name*, if any, is the value of the expression.

```
LASTVAL
```

The last SNMP value obtained (possibly using `DEFINED` or `UNDEFINED`) as a numeric or string value.

9.5.3.2 Constructs

The following constructs or “statements” are used in the control file.

9.5.3.2.1 Assignments

```
X = "ABC"
```

Assignment, the value on the right is assigned to the name on the left.

```
X := expression
```

One time assignment. As per assignment, but the operation is turned into a no-op after the first execution. (This does not prevent a subsequent assignment to the variable).

For example

```
H = `hostname`
```

would be evaluated each time the script is run (after every block), but

```
H := `hostname`
```

would only be evaluated the first time.

9.5.3.2.2 Comparisons and tests

Comparison of strings is performed with alphabetic operators `LT`, `LE`, `EQ`, `NE`, `GE` and `GT`, and numerics with symbolic operators `<` etc.

```
DEFINED (objid)
```

Return a true value if the given object id (which may be specified using macros) is defined, otherwise false. If the value is defined, `LASTVAL` is set to whatever value that was to save further fetches.

```
UNDEFINED (objid)
```

As for `DEFINED` but the other truth value is returned. `LASTVAL` is still set if the value is defined.

```
ISNUM value
```

Return a true value if the value is a number.

```
ISSTRING value
```

Return a true value if the value is a string.

Tests can be combined using `AND` `OR` `&&` `||`. Combine boolean operations in tests.

`AND` is more binding than `OR`. "Short circuit" evaluation is performed, so beware of relying on `LASTVAL` in such cases.

```
IF .. THEN .. [ELIF .. THEN ] [ELSE .. ] FI
```

Conditional construct with optional `ELIF` and `ELSE` parts.

```
MSG STRING
```

Display message on standard error.

```
EXIT number
```

Exit `xtlhp` with specified exit code.

```
FLUSH
```

Flush output to socket.

9.5.3.3 Example

This is the control file supplied by default, with definitions for `gdStatusLineState` etc being provided in the definitions file supplied.

Note that if no `EXIT` is reached, then script file has “succeeded” and output continues.

```
# Test the status to see if there are problems

IF SNMPVAR(gdStatusLineState.0) != 0
THEN
    # Look at paper out indication and exit
    # message

    IF SNMPVAR(gdStatusPaperOut.0) != 0
    THEN
        MSG 'Out of paper'
        EXIT 100

    # Likewise paper jam

    ELIF SNMPVAR(gdStatusPaperJam.0) != 0
    THEN
        MSG 'Out of paper'
        EXIT 101

    # Likewise toner low

    ELIF SNMPVAR(gdStatusTonerLow.0) != 0
    THEN
        MSG 'Out of toner'
        EXIT 102

    # Give up just give general message

ELSE
    MSG SNMPSTR(npSysStatusMessage.0)
    EXIT 103

FI

FI
```

9.6 Xtftp

```
/usr/spool/progs/xtftp [options]
```

`xtftp` is a terminal server program for use by **Xi-Text** where the destination is a server which accepts FTP protocol. Data is taken from standard input, unless overridden by the `-i` option.

9.6.1 Options

No facility for remembering options is provided.

9.6.1.1 -h option

```
-h hostname
```

This is the host name or IP address to be used to access the printer. In **Xi-Text** setup files, it may be appropriate to substitute `$SPOOLDEV` or sometimes `$SPOOLPTR` here.

9.6.1.2 -A option

```
-A hostname
```

Specifies the sending hostname or IP address to be as given. This may be required when sending from hosts with multiple IP addresses.

9.6.1.3 -p option

```
-p portnumber
```

This is the port name or number to be used, if not `ftp` or 21.

9.6.1.4 -u option

```
-u user
```

This is the user name to be quoted to the FTP server.

9.6.1.5 -w option

-w password

This is the password to be quoted to the FTP server.

9.6.1.6 -o option

-o outfile

This is the output file to be stored at the server end. In some cases this may designate the actual printer on a multi-server device, and it may be useful to use `$SPOOLDEV` as the argument to be expanded in **Xi-Text** setup files to be supplied here.

9.6.1.7 -D option

-D directory

This is the directory in which the output file is to be stored at the server end, if not the “home directory”. In some cases this may designate the actual printer, and it may be useful to use `$SPOOLDEV` as the argument to be expanded in **Xi-Text** setup files.

9.6.1.8 -T option

-T timeout

This gives the timeout in milliseconds to await for responses from the server.

9.6.1.9 -i option

-i input

Specify input file if not standard input.

9.6.1.10 -t option

-t

Specify text mode.

9.6.1.11 -b option

-b

Specify binary mode.

9.6.1.12 -d option

-d

Debug. Displays activities on standard error. Increasing information is given if more than one -d option is specified.

9.6.2 Exit Codes

Normal termination is denoted by an exit code of zero.

Errors are indicated with the following exit codes.

1. Usage error, invalid option etc.
2. No host name or ip address given.
3. Cannot open input file.
4. Expecting password as well as user name.
5. Incorrect password.
6. Transmit failure - file rejected.
7. Data accept failed.
8. Cannot create/bind control socket.
9. Cannot connect control socket.
10. Cannot create data socket.
11. Cannot bind data socket.
12. Cannot listen data socket.
13. Cannot get socket name data socket.
14. Unknown service name
15. Invalid IP address
16. Invalid host name
17. Socket write error.
18. Socket read error.

19. No response code from server.
20. Port or store command failed.
21. Generated command too long (excessive length file name etc).
22. Aborted with a signal.

Chapter 10

Configuration of user interfaces

This section looks at the various options available to user programs such as `spr` and `spq`. In some cases, particularly with `spr` and `sqchange`, there is a perhaps bewildering array of options. It is not intended that users should have to remember them all as they can be specified by default and overridden as required.

This is done by using *configuration files* and environment variables.

10.1 Configuration files

A configuration file is a file called `.xitext`, which may appear in the current directory, or in the user's home directory. This file is a text file, containing environment variable type assignments. It may also be used, as described below, to specify alternative help message files.

Options to the user programs enable these files to be generated and edited automatically.

Hopefully you will be familiar with the `-f form` and the `-c n` options to `spr` to set form type and copies. Suppose you want to print a number of documents with form type `a4.lh`, and with 3 copies. Rather than specifying each time

```
spr -f a4.lh -c3 .....
```

it would be easier to put these options into the environment by means of:

```
$ SPOOLOPTS='-f a4.lh -c3'  
$ export SPOOLOPTS
```

and then they would be automatically specified each time, until you set the environment variable to something else. If you wanted only one copy for a particular job, or a different form type, then you could re-specify the option on the command line, this taking precedence. Note the general rule that for every option which “does” something, there is a corresponding option to “undo” it, to provide for this case.

This may be inconvenient in some cases. If you move to a different directory to do alternative work, you may forget to reset the environment variable to something more appropriate. Or you may log out and come back after lunch and forget to set it again. Hence we provide an alternative mechanism using `.xitext` files.

You can plant a `.xitext` file in the current directory to imply options for the relevant programs appropriate to files in that directory. The format of the file is similar to setting environment variables, but without the quotes or export, for example:

```
SPOOLOPTS=-f a4.lh -c3
```

The file may contain comment lines commencing with `#`. In fact any lines not understood are silently ignored).

As with environment variables, you can override the `.xitext` file on the command line.

This technique of planting `.xitext` files in directories is appropriate when, as you move from directory to directory, you want to specify options suitable for the files in that directory; e.g. `invoice` form type in one directory, `letterhead` in another etc.

If you want a given option to apply whatever happens, you can put a `.xitext` file in your home directory. For example if you want to reorganise the screen of `spq` to provide for 6 printers to be displayed at once, rather than specifying `-P6` every time, you can put in your home directory a `.xitext` file containing the line:

```
SPQ=-P6
```

You could instead set an environment variable `SPQ` in your `.profile` or `.login` file; the choice is up to you.

If you specify options in `.xitext` files in your home directory and current directory, and in an environment variable, then they are handled in the following order:

1. Any system-wide defaults are taken (e.g. the user's default form type).
2. The home directory `.xitext` file is processed.
3. The environment is processed.
4. The current directory `.xitext` file is
5. Options on the command line are processed.

Conflicting options encountered later completely override what came first, so that options specified on the command line will take priority whatever else was encountered. As mentioned above, there is a "reset" type option for every "set" type option, so for example the option `-r` resets the `spr` option `-s` to suppress the banner page.

As described below, you can re-assign the functions of the option letters using alternative help message files specified using the `.xitext` files.

10.1.1 Environment variable or keyword names

For all programs except for `spr`, where the environment variable name is `SPOOLOPTS` the environment variable name is the same thing in upper case letters. If you invoke any command under a different name the environment variable names won't change, with the exception of `spstart`, where the environment variable name is tied to that of the program.

To set the environment variable, the format is just the same as for the options in the corresponding command. If you are using the Bourne or Korn shells, type (for example)

```
$ SPOOLOPTS='-f letterhead -c2 -wi'
$ SPQ='-A -P2 -p'
$ export SPOOLOPTS SPQ
```

(The quotes are required if spaces are included, which they usually are).

With the C shell type (for example)

```
% setenv SPOOLOPTS '-f letterhead -c2 -wi'
% setenv SPQ '-A -P2 -p'
```

There isn't any hard and fast rule about whether to use home or current directory .xitext files, or environment variables.

In practice people put “comfort” type options, such as help display options and the display of job numbers in the home directory, options specific to files in a given directory, such as form type or retention on the queue, in that directory and transient requirements in the environment. The intention, however, is to give you some flexibility.

Finally note that the search order for options may be changed as described later.

10.2 User reconfiguration

To allow maximum flexibility, all strings, such as screen headers, error and help messages and prompts used in **Xi-Text** are taken from a set of files, and are thus re-definable

1. On a system-wide basis by editing the files.
2. On an individual basis for each user by allowing each user to have access to his or her own version or versions of the files.

All the keystrokes understood by `spq` and `spuser` are “soft”; the functions may be redefined as required.

The following are examples of the kind of applications these facilities have:

1. Producing customised versions of the product incorporating site names, help messages etc. on the basic screen formats.
2. Providing “seamless joins” between **Xi-Text** and other software with different function key sets.
3. Providing interfaces appropriate to different terminals, in particular taking advantage of function keys provided on those terminals.
4. Providing support for national languages; possibly with different languages on different terminals on the same machine.

10.2.1 Message files

The standard message files all live in the help files directory, by default `/usr/spool/progs`. The files involved are:

<code>spuser.help</code>	The configurable information for <code>spuser</code>
<code>spq.help</code>	The configurable information for <code>spq</code>
<code>splpd.help</code>	The configurable information for LPD interface programs.
<code>int-config</code>	containing the configuration information for the programs internal to Xi-Text : (<code>spd</code> , <code>spdinit</code> , <code>spwrite</code> , <code>spshed</code> , <code>xtnetserv</code>).
<code>xmispq.help</code>	containing the configurable information for <code>xispq</code> and <code>xmispq</code>
<code>xmispuser.help</code>	containing the configurable information for <code>xispuser</code> and <code>xmispuser</code>
<code>rest.help</code>	Which contains all the configurable information for all the rest of the user-accessible utilities and the CGI programs for the web browser interface (this needs to be available for the remote version of the browser interface).

The files are found by default from `/usr/spool/progs` as specified.

If you want to specify an alternative file, then use the configuration file or environment variable mechanism previously described. The following table lists the environment variables and/or keywords used for various user programs:

Environment var/kw	Description
<code>SPQCONF</code>	<code>spq</code> message file
<code>SPUSERCONF</code>	<code>spuser</code> message file
<code>SPLPDCONF</code>	<code>sp.lpq</code> and <code>sp.lprm</code> message file
<code>XMISPQCONF</code>	<code>xmispq</code> message file.
<code>XMISPUSERCONF</code>	<code>xispuser</code> and <code>xmispuser</code> message file.
<code>SPRESTCONF</code>	message file for other utilities including CGI programs for Web Browser interface

For example if you had an alternative message file for `spq`, you could specify its use by means of the environment variable setting:

```
SPQCONF=`pwd`/my-spq-file
export SPQCONF
```

It is important to specify the full path name, otherwise the file will be searched for in whatever directory you move to.

Alternatively you can specify the message file in a configuration file (`.xitext` file) located in your home directory or in the current directory. If it is located in the home directory, it will apply regardless of what directory you are in, otherwise it will only apply whilst you are in the current directory.

The format of lines in the `.xitext` files is similar to that to set environment variables in the shell. For example:

```
SPQCONF=$HOME/lib/myspq.help$TERM
```

Note that environment variable names are also expanded here, so in this example the user is intending to specify a different file according to the setting of the `TERM` environment variable. Environment variables which contain further environment variable names (prefixed with `$`) will be recursively expanded up to 10 times.

There are 3 facilities in the expansion of these lines intended to assist the user to supply defaults etc.:

Firstly, as with the shell, sequences of the form `${VAR-default}` are replaced by the value of environment variable `VAR` if it exists, and otherwise the default string specified.

Secondly the sequence `$0` is replaced by the name (or the last component of the file name) by which the program was invoked, so if you were to put

```
SPRESTCONF=$HOME/lib/helps/$0.help
```

in your `.xitext` file, then the file searched for would be `spstart.help` if you were running `spstart`, `sqlist.help` if you were running `sqlist`, etc. You could likewise provide a completely different-looking version of `spq` with a different name which automatically selected a different message file.

Finally, if it cannot find the file specified, then an attempt will be made to find a file name constructed by deleting the last part of the file name from the path and substituting the standard name (`spq.help`, `rest.help` etc.), before giving up. For example in the above case, if the files `spstart.help` or `sqlist.help` could not be found in the directory `$HOME/lib/helps`, then a further try would be made with the standard name `rest.help` to give the file name `$HOME/lib/helps/rest.help`. (Note that the order of search may be changed, see later).

As for the argument keywords and environment variables, the current directory `.xitext` file takes precedence over environment variables, if any, which in turn take precedence over the home directory `.xitext` file. However once the keyword is found in one of those places, the remainder are not searched; this means that if a non-existent file is specified, say in the current directory `.xitext` file, then the program will abort without looking in the other places.

10.2.1.1 File format

It is easiest to work on these files by copying one of the supplied ones and starting from there. The files are text, and you can use your favourite editor to amend it.

In general a line which doesn't make sense is ignored. However we recommend that you adopt the convention of using comments denoted by the `#` symbol. Blank lines are ignored everywhere.

Example lines within the files might be:

```
K400:?
1K500:o,*
100P:Ok to delete (Y/N)
E100:Unknown command - expecting job queue control
H400:Please reply Y or N
```

The general form of the lines is:

1. An optional *state code*, denoting the state of the program in which the line applies.
2. A key letter, denoting the function, i.e. key definition, prompt, error or help message etc.
3. A code number for which the program looks when it requires a particular line.
4. A colon preceding the definition of the item.

The state and code numbers are arbitrary and compiled into the relevant programs. Hopefully the context and the comments in the supplied files will make the situations in which they are used clear enough.

The types of entries given by the key letters are as follows:

- K** a key mapping
- H** a “help” or similar type message
- E** an error or other information message
- P** a “prompt” or other screen field
- N** a numeric parameter
- A** an option definition

In addition there are screen heading lines, which have a slightly different format using other letters, defined later.

10.2.1.2 Key definitions

Key definitions are read in when the program is first started. As a result conflicts, where a key is defined for two or more different uses, and other errors are detected before further processing is done.

There are two types of key definition; *global* definitions which apply everywhere within the program, and *local* definitions, which only apply in a particular context, enabling you to use the same key in another context for a different purpose.

A global key definition looks like this:

```
K400:?
```

whilst a local definition looks like this:

```
1K500:○
```

The first number is referred to as a *state code*. The number after the **K** gives the internal code into which the key sequence is translated and used to select the program action.

In all the contexts where state codes apply, there is a help message supplied with the same code as the state code. For example state 1 corresponds to the state where the cursor is in the job queue, keys relevant to this state only are prefixed with this state code, and the help message lines for this state start with **H1** : .

The balance of the key definition gives the character sequences which make up the key definition; these will be translated to the internal code. In the above examples the character **?** is globally translated to the internal code 400 and **○** to 500 in state 1.

You can represent control and non-printing characters using `\` and `^` as described for setup files together with the extensions described below. In particular you must represent comma as `\,` and spaces and tabs as `\s` and `\t` respectively.

To give two or more keys for a given command, separated them by commas, as follows:

```
K400:?,*
```

It is because of this that you must represent comma using `\,` as just mentioned.

Many keys, in particular cursor and function keys, generate multi-character sequences. The sequences can be just written out thus:

```
K406:\e[A
```

The whole of the sequence up to the end or to a comma will be searched for and translated to the given internal code when it occurs. The arrival of the characters will be timed so it should be possible to distinguish between key sequences with similar prefixes (often the case with cursor and function keys) and the effect of typing the component characters separately. This particularly affects the escape character, which on its own is commonly used to abort input, but which also invariably prefixes function and cursor key sequences.

You can if necessary tune the timing of the arrival of characters by means of the environment variable `KEYDELAY`, which is set to the number of 1/10ths of a second to wait between the characters of a suspected function key.

If another character is not received within this time, the character will be assumed to be a single key, otherwise it is taken as the start of a function key. The default for `KEYDELAY` is 3/10ths of a second, which should work correctly for the vast majority of terminals.

If this value is too low, then function keys may not always be recognised, whilst if it is too high, then the response to a single escape character, when required, will appear to be slow.

10.2.1.2.1 Special key sequences

As well as the control and non-printing characters which may be inserted using the sequences `\n` and `^c` etc., additional constructs introduced by `\k` are provided to import the user's choice of keys for erase, kill etc., and the *termcap* or *terminfo* definitions of some of the special function keys on the terminal.

There are two problems which can arise when you do this:

1. Choices of single-character keys often conflict with existing uses of the same key, but not in a consistent way for each user or terminal. For example:
 - (a) The character control-U (`^U`) is defined in the supplied files as "scroll-half up". Many people use this as a kill character.
 - (b) Some terminals have cursor keys which generate a single character. A notable example is Wyse 60 terminals on which the left cursor key generates a backspace (*ctrl-h*) character which therefore cannot be distinguished from a "real" backspace character (or indeed *ctrl + h*).

2. *Termcap* and *terminfo* files often tell lies about what sequences are generated by cursor and function keys. This may not always be the fault of the files themselves if the user has extensively used soft key-setup facilities on the terminal.

If you do successfully get past these problems, then the way to import these values is to use the sequence `\k` followed by the symbolic name of the required item. The items available are:

Control keys	
Symbolic Name	Meaning
<code>\kINTR</code>	user's interrupt key
<code>\kKILL</code>	user's kill key
<code>\kERASE</code>	user's erase key
<code>\kQUIT</code>	user's quit key

which are looked up on entry to `spuser` or `spq` from the *termio* or *stty* settings, as set by `stty`, and:

Terminal keys	
Symbolic Name	Meaning
<code>\kUP</code>	string sent by up arrow key
<code>\kDOWN</code>	string sent by down arrow key
<code>\kLEFT</code>	string sent by left arrow key
<code>\kRIGHT</code>	string sent by right arrow key
<code>\kHOME</code>	string sent by home key
<code>\kFnn</code>	string sent by function key <i>Fnn</i>

which are looked up in the *terminfo* or *termcap* database for the terminal type specified in the environment variable `TERM`.

To use these sequences, put the relevant one in place in the key definition for example:

```
K400:?,\kF1
K406:k,\kUP
```

If something does go wrong due to one of the two problems given above, you will find that either `spq` or `spuser` does not start, displaying a message such as:

```
State 5 setup error - clash on character 08 with previously-given value 420 and
new value 530.
```

In this case you can pinpoint the problem by looking for a key definition with 420 in (this is the code for erase last character) and with 530 in (left cursor in some fields). This particular message often appears on Wyse 60 terminals where left cursor is the same as backspace, in other words `\kLEFT` and `\kERASE` produce the same result - control-H or hexadecimal 08.

The other thing which might happen is that the function keys do not work properly or the message "Undefined key sequence" is displayed. This is because you have an incorrect *termcap* or *terminfo* file, or because you have reset function keys.

A common problem on VT100-based terminals is that there are four settings depending upon which combination is selected of “normal” or “application” cursor keys and “numeric” or “application” keypad. You can be reasonably sure that your *terminfo* file will assume one of the combinations whilst the terminal will be set to one of the other three.

This particularly affects cursor keys which are often defined in the *terminfo* file to be `\eOA`, `\eOB`, `\eOC` and `\eOD` (application keys) for up, down, right and left cursor respectively whilst the terminal when switched on generates the “normal” cursor keys `\e[A`, `\e[B`, `\e[C` and `\e[D` respectively. If you run into this problem we suggest that you spell out the combinations explicitly, thus for example

```
K406:k,\e[A,\eOA
K407:j,\e[B,\eOB
```

If you do manage to traverse all these snags and catches, we do encourage you to move over to function keys rather than printing characters as printing characters are thereby “lost” for the purpose of typing strings etc.

Don't forget to adjust your help and error messages to reflect the keys you have set up.

10.2.1.3 Help and error messages

Help and error messages are treated almost identically. The only difference really is that they are displayed slightly differently.

The text to be displayed is given on as many lines as required with the same prefix, for example:

```
E100:Unknown command - expecting job queue control
E100:Type ? for help
H1:? - help (this file)
H1:^L - redraw screen
```

Now in many cases there are specific variables or strings which it is helpful for the help or error message to quote in the message. These can be inserted using “parameters” introduced by `%` as follows:

Sequence	Meaning
%%	"%"
%P	program name (last component)
%F	full path name of message file
%p	process id as a decimal number
%E	<i>errno</i> , interpreted like <code>perror</code>
%U	user name corresponding to effective UID
%R	user name corresponding to real UID
%un	argument <i>n</i> ($0 < n < 9$) interpreted as a UID
%gn	argument <i>n</i> ($0 < n < 9$) interpreted as a GID
%G	group name corresponding to effective GID
%H	group name corresponding to real GID
%dn	decimal representation of argument <i>n</i> ($0 < n < 9$)
%xn	hex representation of argument <i>n</i> ($0 < n < 9$)
%on	octal representation of argument <i>n</i> ($0 < n < 9$)
%cn	character (or hex if non-printing) value of argument <i>n</i> ($0 < n < 9$)
%s	string argument
%t	secondary string argument

Some of the information (program name, process ID, error code, real and effective user & group IDs) is always available. The rest of the information (the 2 string arguments, the 10 parameters) is set up as required for each error message by the calling program. The only way to discover what has been passed is to examine the error message as given in the supplied and to use the same parameters, possibly with a different format.

10.2.1.4 Option syntax

There are two alternatives to using the standard single letter command-line options:

1. You can invoke options with keywords such as `+priority` or `--priority` instead of or as single-letter options such as `-p`.
2. Alternative option letters and keywords may be specified by use of different message files.

Option syntax is defined by means of lines of the form

```
A119:p,priority
```

As with other parts of the file, there is an internal state code, in this case 119, to denote the action to take. Alternatives are specified after the colon and separated by commas. They may consist of either:

1. Any single printing character, as for `p` in the above example, which denotes a “-” type argument, thus `-p`. The character is taken exactly as given, and can be any printing character (a comma or `\` must be escaped using a `\` thus `\,` and `\\` but these, especially the latter, are very strongly discouraged, for obvious reasons).
2. A keyword consisting of alphanumeric characters, “-” and “_” (underscore) starting with a letter, denoting a `+keyword` or `--keyword` type argument, thus `+priority --priority`. Letters in the keyword are case insensitive.

You do not have to define either single-character or keyword type options for any function. If you leave one out then only the other type will be available, or the option will not be available at all.

If you leave out option syntax definitions out of the file altogether, then a default set of options consisting only of the single-character variants of the options as defined in this manual for each program will be supplied.

If you change the options for `spr`, then when you unqueue a job from `spq`, the program `spjobdump` will be run and will pick up message files as for `spr` starting with configuration files in the current directory from which `spq` was invoked, using the options specified to generate arguments for a `spr` command. The same considerations apply to options saved using `+freeze-current` and `+freeze-home` options to various commands.

Of course this means that if you have completely different option definitions in various places, this may be inappropriate and you, and the software, will get totally confused! We recommend some care in using this facility, which was mainly intended for producing national-language versions of **Xi-Text**, with appropriate option names.

10.2.1.5 Prompts

Strings introduced with sequences such as `100P:` are *prompts*. These are transient messages which are generated in various places such as the state codes on printers and the messages at the bottom of the screen indicating so many jobs or printers below.

Some prompts have parameters in (such as the “n jobs below” message), which are introduced with a `%` sequence such as `%s` or `%d` very much like the C `printf` function (which it uses).

A special set of prompts is provided in the internal program file `/usr/spool/progs/int-config`. These provide hexadecimal strings from which the bitmaps for the large characters in the banner page are constructed. These may be adjusted as required, also to provide for additional characters - see the comments in the file.

Another prompt is defined to make `standard` the initial form type assigned everywhere when **Xi-Text** is first installed. If you should want to change this, please remember that it appears in all 4 files and should be changed consistently; the first program to discover that the User control file `/usr/spool/spd/spufile23` does not exist will recreate it with this form type as the default.

10.2.1.6 Numeric parameters

There are some sequences of the form

```
1000N1001
```

in the files. These are used in 3 places:

1. In the `spq` “job options” screen, to configure the order of prompts and the prompt to start with.
2. In the `spuser` display or set privileges options, to configure the order to present the privileges.
3. In the “save options” screens in both `spq` and `spuser` to configure the order in which the prompts are presented.

Please see the comments in the supplied files for how to adjust these parameters.

10.2.1.7 Titles

The screen titles used by `spq` and `spuser` are handled slightly differently. Although they can be multi-line, each line of the title is taken from a different message number. For example, `Dn` is the format for the n th line of the title for the `spq` printers list. The default is:

```
D1: Printer line form status jobno user
```

Similarly the title for `spq`'s job list is taken from `Jn`, the default is:

```
J1:Seq jobno User file form posn size cps pty printer
```

Perhaps you might want to make this more elaborate and incorporate your company name as follows:

```
J1:Bloggs & Sons Ltd -- Print queue --- ? for help
J3:Seq jobno User file form posn size cps pty printer
J5:=====
J6:
```

The height of the title is given by the highest-numbered line specified - 6 in the above case (this will give a blank line). Note that lines 2 and 4 will be generated as blank lines also. There is also an optional footer section, using the same principle, with lines denoted by `T`.

The title for the `spuser -i` screen is taken from `Tn` in the same way, and the trailer from `Un`.

It is permissible to omit any or all of these titles completely if desired, leaving more space on the screen for jobs and printers, or users, as the case may be.

10.2.1.7.1 Enhancements and line drawing in headers

It is possible to specify screen enhancements and line drawing characters within headers. To do this, the following sequences are interpreted within header lines.

`\B` Set bold mode
`\D` Set dim mode
`\F` Set flashing mode
`\U` Set underlined mode
`\I` Set inverse video mode
`\S` Set standout mode

The effect of the above sequences are cumulative and apply until the end of the line or until a terminating `\N` to reset enhancements in encountered.

For example the line

```
\BExample\n Header
```

would be displayed as

```
Example Header
```

Do note that not all *terminfo* files define all of the various enhancements, also that “standout” mode is usually some arbitrary combination of the others.

The line-drawing set, if available, is separately invoked by use of sequences thus;

`\L` Turn on line drawing mode and display upper left corner line drawing symbol.
`\l` Turn on line drawing mode and display lower left corner line drawing symbol.
`\R` Turn on line drawing mode and display upper right corner line drawing symbol.
`\r` Turn on line drawing mode and display lower right corner line drawing symbol.
`\|` Turn on line drawing mode and display vertical edge line drawing symbol.
`\-` Turn on line drawing mode and display horizontal edge line drawing symbol.
`\+` Turn on line drawing mode and display internal intersection line drawing symbol.
`\<` Turn on line drawing mode and display intersection of horizontal line and left edge 'T' line drawing symbol.
`\>` Turn on line drawing mode and display intersection of horizontal line and right edge 'T' line drawing symbol.
`\^` Turn on line drawing mode and display intersection of vertical line and top edge 'T' line drawing symbol.
`\v` Turn on line drawing mode and display intersection of vertical line and bottom edge 'T' line drawing symbol.

Once line-drawing mode is set, then any of the above characters which immediately follow are interpreted without needing to have a “\” in front, so for example

```
\-----
```

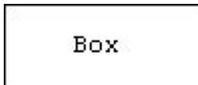
would generate a horizontal line.

Any character not in the above set of line-drawing characters will cancel line-drawing mode. So that you can put a line drawing character in a box next to a line, then the sequence

`\.`

cancels line-drawing mode without inserting a period.

For example to produce a display like this;



use the sequence

```
T1:  \L-----R
T2:  \|  Box   \|
T3:  \l-----r
```

This applies to all headers of screens in `spq` and `spuser`.

If you want an actual “\” in a header you should double it thus “\\”.

10.3 Changing message files

A few tips may be useful:

- Tabs don't work in messages, since they may displayed anywhere on the screen; use multiple spaces instead.
- Keep the messages reasonably short, this avoids having to redraw large areas of the screen just to display a message and obliterate everything else going on.
- When re-defining keys, don't forget to adjust the help and error messages to correspond, and make sure that they are all consistently re-defined throughout all the states of all the programs.

Please note that there is nothing “magic” about the code numbers for global and local keystrokes; as distributed the state codes 400 upwards are assigned to global keystrokes, and 500 upwards to local ones, but if you want a different help key, for example, in different places by defining `1K400` etc. please go right ahead.

Likewise you can change around, for example, the order of access to screens in `spq`, by redefining local versions of the quit key (405) and selection of job and printer screens.

10.4 Local help files

The message file facility copes with most of the problems of customisation. One extra facility, however, is available to assist the operator/user in selecting form types and printers in `spq`.

Normally, if the operator/user asks for help while filling in a printer or form name, a list of names will be supplied by scanning the appropriate directory (either the *printers* directory for the printers list, or the specific directory in for the form type list for the given printer) and using the names contained therein to format a list of possible names. If this is not useful, either because there are other files being stored in the directory, or because more help is needed, a help file can be provided.

If a file called `-Help` exists in the appropriate directory, this will be assumed to contain a suitably formatted list of printer names and other help information to be output instead of the internally generated list.

10.5 Variation of search order

The order in which `.xitext` files are scanned, either to locate message files, or to read options, may be varied from the default if required.

A keyword, optionally placed in the Master configuration file `/etc/Xitext-config`, in each case may be used to vary this.

10.5.1 Search order for message files

The default search order, for example with `spq`, which searches for a file named in the variable `SPQCONF`, is to look:

1. As specified in a `.xitext` file in the current directory.
2. As specified in the environment variable of that name.
3. As specified in a `.xitext` file in the user's home directory.
4. In a standard place, namely `spq.help` in the system help files directory, by default `/usr/spool/progs`.

This order may be respecified by assigning a "PATH variable" style value to the environment variable `XI_HELPPATH`, consisting of directory names separated by colons. The user's home directory may be denoted by `~` and any other user's home directory by `~user`. Any environment variables are fully expanded.

If a directory name is relative (not starting with `/`), it is taken as being relative to the directory from which the program was started. The current directory should be represented as just `.`.

Finally the environment is represented as a `!`.

Thus the default search order is represented as:

```
XI_HELPPATH=". : ! : ~ $LIBRARY"
```

which causes the search to start in the current directory, then the environment, then in the home directory, possible modified if the environment variable `LIBRARY` is set to some string such as `/lib`. (This is included for backward compatibility as some users object to having hundreds of file names starting with `.` in their home directories). Note that there is no way of suppressing the fall-back to the standard location in the system help files directory.

It might be relevant for example, for a French user, with copies of the help files in a separate directory, to set a different search path, thus:

```
XI_HELPPATH=". : ! : /usr/spool/xihelp/Francais"
```

This will affect all **Xi-Text** user-level programs, which search for a help file named by a keyword. (**NB** If this is put in the environment rather than in `/etc/Xitext-config`, it will also affect **Xi-Batch** if it is installed).

10.5.2 Search order for program options

An almost identical system is used for program options, with the keyword `XI_CONFIGPATH` except that the search is in the opposite order. Also the additional symbol “-” is used for command-line options.

The default is therefore:

```
XI_CONFIGPATH=~$LIBRARY:!:.:--"
```

Indicating the search first in the home directory, possibly modified by `$LIBRARY`, then the environment, then the current directory, and finally the command line options.

Note that it is possible to turn off the interpretation of command line options, if required, in this way, by omitting the “-”, limiting the command line arguments to, for example, just job numbers for `sqchange` etc.

Assignments to this will affect all **Xi-Text** user-level programs (and also **Xi-Batch** programs unless the assignment is in `/etc/Xitext-config`).

10.5.3 Freezing options

The `+freeze-home` and `+freeze-current` options in the user level commands and equivalent on-screen facilities in `spq` and `xmspq` are not affected by these options, the software does not attempt to “backwards-interpret” these paths. In particular please note that these facilities save all of the current values of these options regardless of whether they are default values or where they were read from, so you may want to “prune” the values of the options saved.

Chapter 11

Extension of user interfaces

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 such as the 'C' programmer's API for Unix and the API for Windows PCs.

This chapter covers the facilities which are built into the basic product and the Motif GUI option as standard. They consist of hooks where **Xi-Text** can invoke custom built user, administrator and internal programs. Such programs are usually shell scripts, but compiled programs can be used just as easily.

11.1 Message Handling

It is possible to override the message handling invoked when the user receives messages about the status of jobs and printers.

This might either be by mail (with the `-m` option to `spr` and similar, or by a message on the user's terminal, via the `-w` option to `spr` and similar.

The user can also be advised when a printer halts abnormally.

11.1.1 Overriding mail and write options

The messages are despatched via the internal program `spmdisp`. Without intervention, the text of the messages are taken from the internal message file `/usr/spool/progs/int-config` and sent either by the system mail program for "mail" style messages or one of the two "write to terminal" programs `spwrite` or `dosspwrite`.

However `spmdisp` can be instructed to invoke an alternative message program.

This can be done either globally for all users, or locally for individual users.

11.1.1.1 Global overrides

To override message handling globally, the following three environment variables can be assigned. This is most easily done by putting assignments in the master configuration file `/etc/Xitext-config`.

The assignments are to the variables `MAILER`, `WRITER` and `DOSWRITER` for mail options, write message and write message to clients respectively, and name a program or shell script to be invoked.

For example:

```
MAILER=/usr/bin/mymailer
WRITER=/usr/bin/mywriter
DOSWRITER=/usr/bin/mydoswriter
```

The only arguments passed are the user name for `MAILER` and `WRITER`, and the host name and user name for `DOSWRITER`. The text of the message, extracted with substitutions from the system message file, is passed on standard input.

Other information is passed in environment variables as follows:

Variable	Meaning
<code>SPOOLJOB</code>	Job number
<code>SPOOLHDR</code>	Job title
<code>SPOOLUSER</code>	Numeric user id
<code>SPOOLJUNAME</code>	User name submitting job
<code>SPOOLPUNAME</code>	User name job "posted to"
<code>SPOOLFORM</code>	Form type
<code>SPOOLFLAGS</code>	Post-processing flags
<code>SPOOLPTR</code>	Printer name printed on
<code>SPOOLHOST</code>	Host name on which printed

The variables `SPOOLHDR`, `SPOOLPTR` and `SPOOLHOST` are not set if there is no job title, or the message does not involve a printer, or does not involve a printer on a remote host respectively.

11.1.1.2 Per-user overrides

Instead of or in addition to the global overrides, a user can put assignments in his or her home directory, in a configuration file `.xitext` (the same file as can hold default option settings for various user-level programs).

The assignments are exactly as for the assignments described above for the master configuration file, but in addition the user can provide an assignment to the variable `SYSMESG`, to specify alternatives to the actual text of the messages contained in the system message file `/usr/spool/progs/int-config`.

If this file is used, most of the other messages used by other internal programs will probably not be referred to, but it is probably easiest to copy the entire file and just change the job completion messages.

The assignments to `MAILER`, `WRITER` and `DOSWRITER` are treated in exactly the same way as those in the master configuration file described above.

11.1.2 Customising printer error handling

It is possible to insert a custom-defined action to be automatically invoked whenever a printer local to the host goes into error or offline.

This is done by putting a reference to (usually) a shell script in the master configuration file, `/etc/Xitext-config`, with the keyword `SPPTRMSG`, for example:

```
SPPTRMSG=/usr/spool/progs/ptr_alert
```

The command is invoked with 3 arguments:

1. Name of printer.
2. Device name.
3. Description of printer (as a single argument).

11.2 Macros for Interactive User Programs

Macro command facilities are available in `spq`, `spuser`, `xmispq` and `xmispuser` and `xmispuser`.

Up to 9 macro commands relating to jobs and printers in `spq` and `xmispq`, can be set up. Similarly up to 9 macro commands can be defined for users in `spuser`, `xmispuser` and `xmispuser`.

In addition you can prompt for any arbitrary command in all of those places. The command doesn't have to be anything to do with the context of jobs, printers, variables or users. It will, however, always be passed the identity of the currently-selected item, or user as a parameter or parameters.

Except for `xmispq` and `xmispuser`, which have a more convenient system forThe actual commands are placed in the relevant help files, which by default are `spq.help`, `spuser.help` etc. in `/usr/spool/progs`.

11.2.1 Inserting the commands

Firstly you must put the commands in the format:

```
27000P:Prompt for prompted command
27001P:cmd1
27002P:cmd2
27003P:cmd3
27010P:Prompt for prompted command
27011P:cmd1
27012P:cmd2
```

The `2700nP` prompts are the commands for fixed command `n`, `n` 1 to 9 for jobs in `spq`, `xmspq`, users in `spuser` and `xmspuser`, the 0th entry being the prompt for the prompted-for command. The `2701nP` prompts are the commands for fixed command `n`, `n` 1 to 9 relating to printers in `spq` and `xmspq`.

The command involved is any arbitrary shell command which may take a list of jobs, printers, variables or users as the case may be. (The command may be given zero users by `xmspuser` and `spuser` if none are selected).

The commands should be preceded by “!” where appropriate to indicate that they may produce output, and the screen should be refreshed when they are completed.

11.2.2 Binding the keys

This only applies to `spq` and `spuser`. Menu entries are automatically created in `xmspq` and `xmspuser`. The names given to the pulldown entries can be set by editing the lines:

```
xmspq*menubar*macro1.labelString:
```

etc in the resources file.

To bind the keys, you need to assign key codes 600 to 609 in `spq.help` etc, for example:

```
K600:0
K601:\kF1
```

would assign the “0” key to the prompted-for command and F1 to “fixed” command 1.

11.2.3 Example

You want to provide 3 extra function keys in `spq`.

1. Reset the number of copies on a given print job to 3 by pressing F1.
2. ‘Hold’ a print job by setting the printer name to “hold” by pressing F2.
3. ‘Release’ a print job by setting the printer name to blank by pressing F3.

First we create 3 shell scripts, which must be in every user’s path, these are `copies3`, `holdprint` and `relprint` and are one-liners as follows:

```
#!/bin/sh
# Copies3 # Set copies to 3
sqchange -c3 $1
```

For `copies3`,

```
#!/bin/sh
# Holdprint # Set printer to 'hold'
sqchange -Phold $1
```

For `holdprint` and

```
#!/bin/sh
# Relprint # Release job
sqchange -P- $1
```

For relprint.

Next insert the following lines in `spq.help`

```
K601:\kF1
K602:\kF2
K603:\kF3
```

(you might want to make the key definitions specific to state 1 by giving them as `1K601:\kF1` etc)

```
27001P:copies3
27002P:holdprint
27003P:relprint
```

To do the same thing with `xmspq` you just have to add the last 3 lines to `xmspq.help`. Specify the full pathname to `copies3` etc.

Chapter 12

XTLPD

`Xtlpd` is an alternative interface for providing an LPD server to receive jobs from devices expecting LPD protocol and submit jobs to **Xi-Text**.

A script, by default `xtlpd-ctrl`, in `/usr/spool/progs`, specifies the translation from LPD-style control cards to **Xi-Text** job parameters, together with implementations of the long and short listing and remove commands.

To start `xtlpd`, issue the following command, possibly substituting your own control file for the argument.

```
/usr/spool/progs/xtlpd /usr/spool/progs/xtlpd-ctrl
```

This should start as a “daemon process” and return control to the command prompt.

You may wish to add this line to the system startup files to start up **Xi-Text** when the system is booted.

12.1 Format of XTLPD control file

The control file consists of 2 basic types of line apart from comments, introduced by a # sign.

1. Variable assignments, looking very much like shell variable assignments, which are performed as the file is read. Some assignments provide additional options to the process, such as log file generation and debugging. Assignments have 3 variants, =, += and =+ which replace, append after and prepend before any existing values respectively.
2. Specifiers, which are “remembered” and interpreted when the time comes to read a C-file.

The following are examples of assignments:

```
LOGFILE=xtlpd-log  
DEBUG=2  
PrCmd = "pr -f -l66"
```

The following are examples of specifiers:

```
S<s>:
```

```
H<s>: sprflags += " -o <s>"
W<n>: propts += " -w<n>"
f<f>*: | spr $sprflags $cpsflags -f $Form
p<f>*: | $Prcmd $propts | spr $sprflags $cpsflags -f $Form
```

The first case describes the handling of an “S” control card. The rest of the line is a string, which we choose to ignore.

The next case describes the handling of an “H” control card. In this case we do an assignment, to append the given string to the variable `sprflags`. The `<s>` in the right hand side of the assignment is replaced by whatever string followed the “H” on the control card.

Next we describe how to handle a “W” control card. The `<n>` denotes that a number is expected, and is inserted in the right hand side of the assignment.

The next two lines give two alternative file handling commands. The `<f>` denotes that a file name is specified. The “*” indicates that the control card may be repeated to denote the number of copies required (we explain how this is handled shortly).

The pipe symbol “|” denotes that we now want to run a command with the relevant file or files “piped into” it. The various variables are substituted first. The file name is either the name read from the control card, or a list of space-separated file names taken from a variable name given in parentheses.

There are also two internal commands are `<indent>` and `<unlink>`. These are available to implement the I and U control cards respectively.

There may be more than one specifier for each control letter. Each is treated in the order given.

12.1.1 Assignment format

As previously stated, assignments may take the forms:

```
Variable = String
Variable += String
Variable =+ String
```

These commands assign a value to the variable, appends the string to the current value and pre-pends the string to the current value respectively.

The string may be “as is” (leading spaces are deleted), or delimited by single or double quotes. It may also, except in an assignment in a specifier, be a shell command enclosed in backquotes with the same meaning as with the shell. `$variable` constructs are expanded except where the string has internal quotes or characters escaped with a `\`. Note that, unlike the shell, this applies regardless of whatever kind of quotes the string has around it. The quotes just specify the limits of the string.

Further examples of assignments might be:

```
Form = "-f "
Form += `spuser -d|grep Default|sed -e 's/Default form \(.*\)$/\1/'`
```

12.1.2 System Variables

The following are so-called “system variables”, which the Control file may, and in some cases must specify.

Variable	Meaning
<code>XTLPDPOOL</code>	Spooler directory. This must be specified.
<code>PORTNAME</code>	Name or number of tcp port name to use. This must be specified.
<code>LOGFILE</code>	Append messages to specified log file (relative to the spooler directory). If omitted, messages appear on standard error.
<code>DEBUG</code>	Debugging level. 0 = none, 1 = major ops, 2+ all operations.
<code>SHORTLIST</code>	(More or less compulsory). Shell command to execute to generate short listing of jobs.
<code>LONGLIST</code>	(More or less compulsory). Shell command to execute to generate long listing of jobs.
<code>REMOVE</code>	(More or less compulsory). Shell command to execute to delete jobs.

There are additional system variables which are redefined at the start of each job.

Variable	Meaning
<code>HOSTNAME</code>	Host name which the job came from.
<code>PRINTER</code>	Printer name which lpr thinks it's talking to.
<code>PERSON</code>	(In remove command) the "person" parameter
<code>CMDLINE</code>	(In list and remove commands) the users and jobs list.

The last three of these are “exported” to the environment.

Note that the host name assigned to `HOSTNAME` may not necessarily be the same as on the control cards, as the former comes from the `/etc/hosts` on which `xtlpd` is run, and the latter from that from which the job originated. The host names which `spr` and other **Xi-Text** programs use come from the file `/etc/Xitext-hosts` and may be different again.

12.1.3 Handling Repeats

The C-files in the LPD protocol specify multiple copies by just repeating the f commands the requisite number of times. If that is good enough, and you want a separate **Xi-Text** job for each repeat, then forget this section.

It is more efficient, however, to provide for the “-cn” parameter to `spr` to be specified. To do this what we do is to say that the specifier is repeatable, by use of the “*”, thus

```
f<s>*: . . .
```

Next we specify a repeat specifier. This looks like this:

```
*<n>:cpsflags =-c <n>
```

The effect of this is that whenever a control card is repeated, the repeat specifier is executed first, and then the control card specifier is executed once. The `<n>` is of course replaced by the number of copies involved.

A refinement of this allows you to distinguish between repeats of more than one copy, and provide for code to be executed then:

```
*<n>: cpsflags=
*<n>*: cpsflags =-c <n>
```

The second “*” signifies work to be done when the repeat is more than one. The first line is thus always obeyed, setting `$cpsflags` to blank, and the second is only obeyed when the copies is greater than 1.

If you want to specify actions to take place at the end of the block of repeats, you should use the special “.” command thus:

```
.<n>:cpsflags=
```

12.1.4 Beginning and Ending actions

You may want to specify actions to take place at the beginning and end of each C-file, as several C-files may be passed across in each transmission and you may want to reset variables at the start, or to defer commands until the end.

The following specifiers provide for this:

```
!: . . .
@: . . .
```

Specify actions at the beginning and end respectively.

Again, multiple occurrences of these may appear and are treated in order.

12.1.5 Internal commands

Two internal commands are provided:

- `<indent>` followed by a number or `<n>` provides for indentation as `lpr` expects by prepending spaces to each line (empty lines do not have any spaces added). The number of spaces can be fixed, in which case a number should follow, or as read from the control card.
- `<unlink>` is used to implement the unlink command. This is followed by the file names to unlink.

Examples of these are:

```
I<n>:<indent><n>
U<f>:<unlink><f>
```

In the last case you might want to be sure of doing the “unlinks” only at the end thus:

```
U<f>:unlinks += " <f>"
@: <unlink> $unlinks
```

12.2 Example Control File

```

# Define variables used to configure the daemon process

PORTNAME=printer
XTLPDPOOL=/var/spool/incoming-jobs
LOGFILE=xtlpd-log
DEBUG=2

# Commands to invoke to produce short and long lists

SHORTLIST=/usr/spool/progs/shortlist
LONGLIST=/usr/spool/progs/longlist
REMOVE=/usr/spool/progs/remove

# Initialise "pr" command options

PrCmd = "pr -f -l66"

# Get default form type

Form = `spuser -d|grep 'Default form'|sed -e 's/^.* \([^ ]*\)$/\1/'`

# Control character interpretation
# First give things we want to initialise
# Note use of special spr options -E and -o, and further down -U

!: sprflags=-E BSD -o $HOSTNAME
!: propts=
!: cpsflags=
!: <indent>0
!: plainlist=

S<s>:
J<s>: header = "-h '<s>'"
C<s>:
L<s>: sprflags += " -U <s>"
T<s>: propts += " -h '<s>'"
H<s>:
P<s>:
W<n>: propts += " -w<n>"
I<n>: <indent><n>
1<s>:
2<s>:
3<s>:
4<s>:
N<s>: header = "-h '<s>'"
M<s>: sprflags += " -m"

# Pre-repeatable.
# Flush out any pending files from the last command
# We do it this way to absorb Ns which come after the files.

```

```

*<n>: (plainlist) | spr $sprflags $cpsflags $header -f $Form
*<n>: (prlist) | $Prcmd $propts | spr $sprflags $cpsflags -f $Form
*<n>: (pslist) | spr $sprflags $cpsflags $header -f $Form.ps
*<n>: (trofflist) | spr $sprflags $cpsflags $header -f $Form.troff
*<n>: (ditrofflist) | spr $sprflags $cpsflags $header -f $Form.ditroff
*<n>: (dvilist) | spr $sprflags $cpsflags $header -f $Form.dvi
*<n>: (plotlist) | spr $sprflags $cpsflags $header -f $Form.plot
*<n>: (rasterlist) | spr $sprflags $cpsflags $header -f $Form.raster
*<n>: (cifplotlist) | spr $sprflags $cpsflags $header -f $Form.cifplot
*<n>: plainlist =
*<n>: prlist =
*<n>: pslist =
*<n>: trofflist =
*<n>: ditrofflist =
*<n>: dvilist =
*<n>: plotlist =
*<n>: rasterlist =
*<n>: cifplotlist =

# The next line sets the copies number to default
# The line after is only executed if the number of copies > 1
# and sets to the number of copies

*<n>: cpsflags =
*<n>*:cpsflags = " -c <n>"

# Repeatable commands

f<f>*:plainlist += " <f>"
p<f>*:prlist += " <f>"
l<f>*:pslist += " <f>"
t<f>*:trofflist += " <f>"
n<f>*:ditrofflist += " <f>"
d<f>*:dvilist += " <f>"
g<f>*:plotlist += " <f>"
v<f>*:rasterlist += " <f>"
c<f>*:cifplotlist += " <f>"
U<f>: unlinks += " <f>"

# End of file, flush out any pending files and unlink

@: (plainlist) | spr $sprflags $cpsflags $header -f $Form
@: (prlist) | $Prcmd $propts | spr $sprflags $cpsflags -f $Form
@: (pslist) | spr $sprflags $cpsflags $header -f $Form.ps
@: (trofflist) | spr $sprflags $cpsflags $header -f $Form.troff
@: (ditrofflist) | spr $sprflags $cpsflags $header -f $Form.ditroff
@: (dvilist) | spr $sprflags $cpsflags $header -f $Form.dvi
@: (plotlist) | spr $sprflags $cpsflags $header -f $Form.plot
@: (rasterlist) | spr $sprflags $cpsflags $header -f $Form.raster
@: (cifplotlist) | spr $sprflags $cpsflags $header -f $Form.cifplot
@: <unlink> $unlinks

```

12.3 Example short and long list commands

The following shell/awk scripts implement the short and long list commands referred to in the above example control file.

12.3.1 Lpq short format

```
# Turn sqliist output into equivalent of lpq short format output

qlim=${PRINTER--}
count=1

sqliist -z -q $qlim -l -F '%u %N %K %h' | awk '{
if (count == 0)
print "Rank  Owner Job  Files  Total Size"
count++
if (int(count/10)%10 == 1) th = "th"
else if (count%10 == 1) th = "st"
else if (count%10 == 2) th = "nd"
else if (count%10 == 3) th = "rd"
else th = "th"
num = sprintf("%d%s", count, th)
printf("%-7s%-11s%-5s%-38s%s bytes\n", num, $1, $2, $4, $3);

END { if (count == 0)
print "no entries"
}'
```

12.3.2 Lpq long format

```
# Turn sqliist output into equivalent of lpq long format output

qlim=${PRINTER--}
count=1

sqliist -z -q $qlim -l -F '%u %N %K %c %o %h' | awk '{
count++
if (int(count/10)%10 == 1) th = "th"
else if (count%10 == 1) th = "st"
else if (count%10 == 2) th = "nd"
else if (count%10 == 3) th = "rd"
else th = "th"
num = sprintf("%d%s", count, th)
printf("%s:  %-35s[job %s%s]\n", $1, num, $2, $5)
if ($4 != 1)
num = sprintf("%-3d copies of %s", $4, $6)
else
num = $6
printf("  %-33s%s bytes\n\n", num, $3)
```

```

}
END { if (count == 0)
print "no entries"
}'

```

12.4 External Queue Specification

Three additional options are used in the the `spr` command to support `xtlpd`. Exceptionally, these may only be used by `spooler` and `root` regardless of the permissions setting via `spuser`. Attempts by other users to access them will be silently ignored, as will invalid parameters.

The option `-En` will set an external queue number of `n`. The queue number may be given as a number, or a name as specified in the external queue definition file.

The option `-User` overrides the authorship of the job, setting it as though it had been queued by the given user. (This is in distinction to the option `-uuser` which sets the user name on the banner page, but leaves the owner of the job unaffected).

The option `-ohostname` overrides the "originating host" name of the job, as though it had been submitted from the given host.

The purpose of these options is to enable `spr` to be run via daemon processes with the parameters set appropriately. None of these options are intended to be invoked by general users.

The external queue number has no effect other than changing the disposition of messages. There are currently no options to display this, although the number can be obtained from the field `apispq_extn` in the job structure of the API.

12.5 External queue definition file

This file is a text file, with `#` comments, consisting of 4 fields per line.

1. An external queue number, 1 to 65535.
2. A name, e.g. BSD, Novell.
3. A "mailer" program or shell script.
4. A "writer" program or shell script.

The mailer or writer programs are invoked in circumstances where mail or write messages are called for. The arguments passed are the user name and the originating host name, together with the environment variables as specified for `spmdisp`.

The two programs should be given as absolute path names, or relative to `$SPROGDIR` (which defaults to `/usr/spool/progs`).

Either or both programs may be omitted, or given as `-` to completely suppress the relevant messages.