



TIP/ix Utilities

IP-617

December 2014

This edition applies to TIP/ix 2.5 and revision levels of TIP/ix 2.5 until otherwise indicated in a new edition. Publications can be requested from the address given below.

Inglenet Business Solutions Inc reserves the right to modify or revise this document without notice. Except where a Software Usage Agreement has been executed, no contractual obligation between Inglenet Business Solutions Inc and the recipient is either expressed or implied.

It is agreed and understood that the information contained herein is **Proprietary** and **Confidential** and that the recipient shall take all necessary precautions to ensure the confidentiality thereof.

If you have a license agreement for TIP Studio or TIP/ix with Inglenet Business Solutions Inc, you may make copies of this documentation for internal use. Otherwise, you may not copy or transmit this document, in whole or in part, in any form or by any means, electronic, mechanical, photocopying, or otherwise, without the prior written permission of Inglenet Business Solutions Inc.

Inglenet Business Solutions Inc

Toll Free: 1-800-387-9391
Website: <http://www.Inglenet.com>
Sales: Sales@Inglenet.com
Help Desk: HelpDesk@Inglenet.com

TIP Studio, TIP/ix, and TIP/30, and are registered trade marks of Inglenet Business Solutions Inc:

This documentation occasionally makes reference to the products of other corporations. These product names may be trade marks, registered or otherwise, or service marks of these corporations. Where this is the case, they are hereby acknowledged as such by Inglenet Business Solutions Inc.

© Inglenet Business Solutions Inc, 1990-2014

Contents

TIP/ix Utility Programs	1
Terminfo.....	1
Basic Terminal Operation - HARDWARE.....	1
Terminfo - Terminal Capability Database - UNIX.....	1
curses - CRT Screen Handling Package - UNIX.....	8
aft - Active File Table	8
apb - All Points Bulletin	11
calendar - Display Calendar	11
Function Key Usage:	12
cobxref - COBOL Cross Reference Report.....	13
connect - Connect to another TIP System.....	14
dcheck - ISAM File Index Integrity Check.....	16
ddu - Disk Display and Update Utility	20
ddu Function Keys.....	24
Display a Record	24
Specify offset within record	25
Add a record.....	25
Update a Record	26
Delete a Record	27
Switch to Another File	28
Search for a String	28
Print Records from File.....	28
GDA Support	29
Using the Menu Bar.....	29
defkey - Define Function keys	29
dfu - Dump File Utility	31
die - Abort a Program.....	32
dsf - Display File Information	33
ebcasc - EBCDIC to ASCII Translation	35
ej - Schedule a TIP/ix Shutdown	36
fclose - Close Online Files.....	37
fcsdbug - Support Utility	40
fileview - browse, print, and delete files	40
Main Menu Screen	41
List of bookmarks:	44
List of opened files:	45
Screen of Deletion:.....	45
fin - Exit TIP/ix shell	45

fixperms - Adjust Permissions	46
fopen - Open Online Files	47
free - Release Active Files	50
freset - Close and Re-open Online Files	51
FSE - Full Screen Editor	54
Line Numbers	58
Column Ranges.....	59
Strings	59
Command Summary	60
AB - Add Lines Before.....	63
Ad - Add Lines	64
BX - Create Comment Box.....	66
CB - Copy Lines Before.....	68
CC - Copy Column Range.....	69
CO - Copy Lines After	70
DE - Delete Lines	71
DU - Duplicate Lines	72
En - End Full Screen Editor	73
EX - Execute TIP/ix Command Lines	73
F- Find Backward	75
F# - Define Function Key.....	75
FA - Find All Lines Containing a String	77
FI - Find Lines Containing a String.....	79
FM - Find Lines Containing a String.....	80
He - Help for Full Screen Editor	80
IB - Insert Empty Lines Before	80
In - Insert Empty Lines After.....	81
KF - Keep Commands Off	82
KN - Keep Commands On.....	83
Li - List Lines on Screen.....	85
LL - List Last Page	85
MA - Margin Set	86
MB - Move Lines Before.....	87
MC - Move Constant or Columns	88
MF - Marker Off	90
MN - Marker On.....	90
MO - Move Lines After	91
O - Set Language " "	92
P- - Peek Scroll Down	94
P+ - Peek Scroll Up.....	95
PE - Peek at Line	96
Pr - Print Lines.....	98

Qu - Quit Full Screen Editor	98
RC - Recall Last Command.....	98
Re - Read from TIP/ix Library, UNIX file, or Edit Buffer	98
SA - Sort Ascending	101
SD - Sort Descending.....	102
SE - Set Options.....	102
SH Execute UNIX Shell.....	107
SP - Substitute and Display Changes	108
Su - Substitute Text.....	108
SW - Switch exchange Two Lines	109
Up - Update Line Range.....	110
UX - Execute a UNIX Command	112
WE - Write Module and End.....	113
WN - Write No Overwrite Prompt	113
WQ - Write Module and Quit	113
WR - Write Module to TIP/ix Library, UNIX file, or Edit Buffer	114
WX - Write without Overwrite Check and End	116
WZ - Write without Overwrite Check and Quit	116
+ - Forward Space Lines	116
- - Backward Space Lines	117
= - Set Options	118
< - Shift Display Left	118
> - Shift Display Right.....	119
Call fse Recursively	120
% - Call fse with No Return - Not Implemented	121
#d - Saving Line Numbers.....	122
!d - Clear fse Registers.....	124
Function Key Usage	125
Pattern Matching	127
fxp - Printing Postscript Output	132
genmain - Write Unix main Program.....	134
groups - Change Users Active Groups.....	135
help - Display Help	140
Call Another Help Module	141
Chain to Another Help Module	142
Display Another Help Screen Full	143
Define a Module Title	143
Define Help Module Security.....	143
Sorted Call/Chain Table	143
Display Call/Chain Table	144
isonline - is file available	144

isreorg – Reorganize ISAM files	147
jrnsrap - Swap Journal Files	148
lopen - Open Connection to LOCAP	149
lclose - Close Connection to LOCAP	150
lreset - Reset Connection to LOCAP	152
mail - TIP/ix Mail System.....	153
Mail System Glossary.....	153
Mail User Guide.....	155
Mail Program Syntax	158
A - Access a Mailbox.....	159
C - Change Preferences.....	160
DEL - Delete Mail	160
DI - Mailbox Directory.....	161
E - End Mail Program	161
EX - Export Mail	162
FIN - End Mail and Logoff	163
F - Forward Mail	163
H - Help for Mail Commands	164
I - Import Letter Text - Not Implemented	164
L - Look at Mail.....	165
LOGOFF - End Mail and Logoff	165
N - Read Another Mailbox	165
P - Print Mail.....	166
Q - End Mail and Logoff	167
R - Read Mail	167
RES - Restore Letter from Library - Not Implemented	168
RET - Retract Letter	169
SA - Save Letter in Library - Not Implemented	170
S - Send Letter	171
U - Compose Letter with Editor - Not Implemented	172
US - List user ids	173
Mail Programming Guide.....	173
maillist - TIP/ix Mail List System	175
Command Summary	176
A - Add Mail List	177
C - Change Mail List.....	179
DEL - Delete Mail List.....	179
DIR - Display Mail List Directory.....	180
E - End Program.....	181
F - Find References.....	181
I - Inquire Mail List	182
M - Compare Mail Lists	183

P - Print Mail List	184
REM - Remove Name from Mail List.....	184
REN - Rename Mail List.....	185
RES - Restore Mail List - Not Implemented	186
S - Save a Mail List - Not Implemented.....	186
menu - TIP/ix Menu System.....	187
Menu System Requirements	188
Menu Processor	188
Menu Parameter Processing.....	190
Menu Help Requests	191
menuar - Menu Maintenance	193
C - Copy a Menu	193
DEL - Delete a Menu	194
DI - Display Menu Directory	194
E - End Program.....	195
H - Display Help Information	195
HDEL - Delete a Help Screen	195
HS - Define Help Screen	196
L - List Information about Menu.....	196
P - Print Menu	197
QP - Quick Print of Menu	197
Q - End and Logoff	198
REN - Rename a Menu	198
TI - Specify Timeout Value	199
menudef - Menu Definition	199
Phase 1 of Menu Definition	201
Phase 2 of Menu Definition	201
Menu Parameter Processing.....	204
Phase 3 of Menu Definition	205
Function Key Use	206
mfm - Message File Maintenance.....	207
AD - Add Messages	211
CH - Change Messages	212
DE - Delete Messages	213
EX - Write Messages to a Library Element	213
IM - Read Messages from a Library Element.....	214
LI - List Messages	215
LM - Language Code Maintenance.....	216
LO - Localize a Set of Messages	217
PA - Pack the Message File	219
PM - Product Code Maintenance	219
PR - Print Messages	220

RE - Read File from a Library Element	221
SH - Show Individual Messages.....	222
WR - Write File to a Library Element.....	223
msg - Send Terminal a Message	224
msgar - Message Format Utility	226
Features of MSGAR	226
MSGAR Commands.....	227
ALLON or ALLOFF	228
CObol name	229
CUrsor name row col.....	229
DElete name.....	230
DELETE pattern	230
HDR name.....	230
HDELETE pattern.....	231
HImport infile [name]	231
HList pattern [outfile]	232
HSetKey pattern Syskey Scrkey Fldkey.....	232
HUpdate name	233
HXport pattern [outfile]	233
K	233
List pattern [outfile]	233
Print name [outfile]	234
Rename oldname newname	234
Import infile [name].....	234
IBms infile [name].....	235
IDps infile [name].....	235
IDPSIMS infile [name]	236
IDPSFCA infile [name]	236
ISFG infile [name].....	236
Xport pattern [outfile]	237
XCport pattern [outfile]	238
XDps pattern [outfile].....	238
XSFG pattern [outfile].....	239
UXport pattern [outfile]	240
UXCport pattern [outfile].....	240
UXSFG pattern [outfile]	240
msgshow - Screen Format Testing	241
newpass - Change Password	242
note - Display Informational Message	243
pingtip - Report TIP/ix Processes	243
precob - COBOL Conversion Aid	243
purge - Remove Process.....	245

readjrn - Read TIP/ix Journal File	247
remove - Exit from Remote LOCAP	249
rfaxlt - Remote File Access Translation	250
rollfwd - Recover Lost Files	252
scratch - Erase Edit/Dynamic Files	257
script - execute a script from TIP/ix	259
seqreorg – Reorganize sequential files	260
SHUTDOWN - Shutdown Processing	261
smfile - File Definitions	261
Add a File Definition	263
Display a File Definition	278
Change a File Definition	279
Delete a File Definition	280
Get Security Records for File Definitions	281
Print File Definitions	284
Browse File Definitions with TQL	285
smgrpset - Group Set Definition	289
Add a Group Set Definition.....	290
List Group Set Definitions.....	293
Display a Group Set Definition	294
Change a Group Set Definition	295
Delete a Group Set Definition.....	296
Print Group set Definitions	297
Browse Group Set Definitions with TQL.....	299
smlocap - Define A LOCAP	301
Add a LOCAP Definition	302
List LOCAP Definitions.....	311
Display a LOCAP Definition	312
Change a LOCAP Definition.....	313
Delete a LOCAP Definition.....	314
Print LOCAP Definitions	315
Browse LOCAP Definitions with TQL	317
smprint - Define A Print File	319
Add a Printer Definition	321
Defining PRNTR - An Example	325
List Printer Definitions	326
Display a Printer Definition	327
Change a Printer Definition	328
Delete a Printer Definition	330
Get Security Records for Printer Definitions.....	331
Print Printer Definitions.....	333

Browse Print Definitions with TQL.....	335
smprog - Define a Program	337
Add a Program Definition	339
List Program Definitions	354
Display a Program Definition.....	355
Change a Program Definition	357
Delete a Program Definition	358
Get Security Records for Program Definitions	359
Print Program Definitions	362
Browse Program Definitions with TQL	363
smsec - Define System Security	367
Add a Security Record	369
List Security Records	376
Display a Security Record.....	382
Change a Security Record	382
Delete a Security Record	383
Get System Record for a Security Record	386
Print Security Records.....	387
Browse Security Definitions with TQL	389
smterm - Terminal Definition.....	392
Add a Terminal Definition.....	394
List Terminal Definitions By Terminal Name	399
List Terminal Definitions By Device Address	401
List Terminal Definitions By Normalized Device Address	402
Display a Terminal Definition.....	404
Change a Terminal Definition.....	405
Delete a Terminal Definition	406
Reset Physical Device Settings from /etc/hosts file	407
Print Terminal Definitions	407
Browse Terminal Definitions with TQL	410
smuser - Define a user id.....	412
Mechanics of TIP/ix security.....	414
Add a user id	415
List user ids	421
Display a user id Definition.....	422
Change a user id Definition	424
Delete a user id Definition	425
Print user id Definitions	426
Browse user id Definitions with TQL	428
sorter - Sort TIP/ix Edit Buffer	431
STARTUP - Startup Processing.....	432

status - Display System Statistics	432
c - Printer file report.....	435
d - Dynamic file report	435
e - Edit buffer report	436
f - File report	438
h - Command Help	439
i - Abbreviated I/O report.....	440
k - Active record key report	441
l - LOCAP information report.....	442
m - Memory information report.....	443
q - Queue information report	444
r - Run report	445
s - System status report	446
t - Transaction Response Time	447
STOP - Shutdown TIP/ix Immediately	448
tcm - TIP/ix Configuration Menu	449
tfd - Screen Format Definition Utility	450
Features of tfd	450
Starting tfd	450
Layout of the tfd Main Screen	452
Window Interface.....	455
Editing Modes.....	457
Field Definition Information.....	459
Performing File Operations	466
Setting Options.....	470
Performing Block Operations	479
Group Operations.....	484
Changing Field Characteristics	490
The Help System.....	493
tipctl - System Startup/Shutdown	500
Shutdown Security Requirements	502
Prompt Screen	502
a - system status	503
apb - Broadcast All Points Bulletin	504
b - boot TIP/ix	505
d - disable logins	507
e - enable logins	507
eoj - Schedule a TIP/ix Shutdown	508
k - kill user sessions	509
msg - Send a Message	510
restart - Shutdown then Boot.....	511
s - shutdown TIP/ix.....	512

stop - Shutdown Immediately	513
w - Recover then Shutdown	513
tipdump - Report Shared TIP/ix Memory	514
tipflg - Flag Manipulation	514
tipix - Command Shell.....	516
tipix - Command Line Syntax	521
tipix - Shell Commands	521
tipix - Command Recall	522
tippack - export TIP/ix files	522
Packing List	524
TIP/ix System File Records: SET	524
TIP/ix System File Records: RECORD Statement	525
Files: FILE Statement.....	527
Files: OBJECT Statement	529
Object Files	529
Comment.....	530
Echo	530
Example Packing List.....	530
tippager - Page Browsing Transaction	531
Commands	531
tippcstm - Thread Manager.....	532
tipsam - System Activity Monitor	534
Main Menu.....	535
tipsubp	539
tipupack - import TIP/ix files.....	539
tlib - Librarian Services.....	541
TLIB Commands	541
TLIB Options	542
TLIB Input and Output Specifications.....	543
COPY - COPY Data	545
DELETE - Delete Library Element.....	548
DIR - Display Library Directory	549
END - End TLIB Interaction.....	550
FDIR - Display Abbreviated Library Directory	550
HELP - Help for TLIB Commands	551
LIST - List Input at Terminal	551
PRINT - Print Input.....	552
QUIT - End TLIB and LOGOFF.....	554
SETOF - Set TLIB Option Off.....	554
SETON - Set TLIB Option On	555
users - List user ids	556

whoson - Who is on	558
wmi - Who am I.....	564
xfer - Data Transfer Facility	566
xfer a Library Element	567
xfer a TIP/30 Screen Format.....	568
 TIP/ix System Modules.....	 570

TIP/ix Utility Programs

Terminfo

TIP/ix uses the UNIX Terminfo terminal capability database and the curses screen-handling package to interface with all terminals.

The following sections provide brief explanations of the basics of terminal operation and how TIP/ix uses Terminfo and curses.

These sections are intended for use by the SYSTEM ADMINISTRATOR when solving problems with terminal definitions. For example if your delete key is not working or your screen display attributes appear to be wrong, these sections should help you solve these problems. Further information about Terminfo and curses can be found in your UNIX documentation or "man" pages.

Basic Terminal Operation - HARDWARE

Before explaining how TIP/ix interacts with Terminfo and curses, some basic knowledge of how a terminal operates is required.

In its simplest form, a terminal is a device for input and output of data. When data is received by the terminal, it is normally displayed on the CRT. When a key is pressed on the keyboard the terminal sends that keystroke.

However, not all data received by the terminal is for display on the CRT. Most terminals recognize sequences of characters, known as escape sequences, as instructions to perform some operation. For example, a DEC VT100 terminal will home the cursor when it receives the three characters ESC [H immediately following each other. These escape sequences control all operations of the terminal including cursor positioning, display attributes/colors, tabs and inserting or deleting characters.

Similarly, not all keys on the keyboard send single characters. Function and cursor keys generally emit sequences of characters when pressed. For example, a WYSE WY-60 terminal sends the characters CTRL-A @ CR when F1 is pressed.

Terminfo - Terminal Capability Database - UNIX

Terminfo is a database that describes the capabilities of terminals. To define a terminal in Terminfo, you must follow these steps:

1. Describe your terminals in Terminfo source files. You can use `artie` to produce the Terminfo source files.
2. Use the UNIX `tic` utility to produce the Terminfo database.

Types of Capabilities:

Each Terminfo source file contains three types of terminal capability definitions:

Boolean

Boolean capabilities specify terminal attributes that are true (non-zero) or false (zero).

Some examples: Does the cursor wrap from the leftmost column back to the last position on the previous line? Does a terminal use XON/XOFF handshaking?

Numeric

Numeric capabilities define characteristics of the terminal that can be expressed as numbers.

Some examples: What is the number of lines/columns on the display? How many colors does the terminal have?

String

String capabilities; define controlling escape sequences, such as the sequence that causes the terminal to delete a line on the display. The string capabilities also define the character sequences emitted when a key, like insert, is pressed.

In our discussion of terminfo, we will only be mentioning a small subset of the possible capabilities. For a list of all supported capabilities, see your UNIX documentation or "man" pages.

Header Line:

A terminfo source file begins with a header line. This line specifies the terminal name, any alternate names or aliases and a long descriptive name for the terminal. Each name is delimited by a vertical bar character and the line is terminated with a comma. This line **MUST** begin in column one.

Example:

The following is a sample header line from the terminfo source file for a 386 console:

```
AT386|at386|386AT|386at|at/386 console,
```

Capability Lines:

Following the header line are one or more lines of capabilities. Each Boolean/flag capability is specified with the name of the capability. Numeric capabilities are specified as the name followed by a hash

character (#) and then a numeric value. String capabilities are specified by the name followed by an equal sign character (=) and then a series of characters.

Special Characters:

Non-printing characters and characters with special meanings in terminfo source files can be specified as shown in the following table:

Character	ASCII (octal)	Use
BS	0x08 (010)	\b
HT	0x09 (011)	\t
LF	0x0A (012)	\n
FF	0x0C (014)	\f
CR	0x0D (015)	\r
ESC	0x1B (033)	\E
SP	0x20 (040)	\s
,	0x2C (054)	\,
:	0x3A (072)	\:
\	0x5C (134)	\\
^	0x5E (136)	\^
DEL	0x7F (177)	^?
NUL	0x80 (177)	\0

See the Appendix in the **TIP/ix Programming Reference** for ASCII/EBCDIC tables showing all special characters.

Control characters (ASCII Hex 0x00 through 0x1F) can be specified using carat notation, where CTRL-x is represented as ^X. For example, the BEL character (ASCII hex 0x07) can be specified as ^G. Alternately octal notation can be used for these characters. Using this notation the BEL character is \007.

Defining Capabilities:

Column one of a capability line must be blank.

If a line contains several capabilities, they are comma delimited.

Each capability line is terminated by a comma.

Example:

The following sample line specifies a terminal, which uses XON/XOFF handshaking, has 80 columns and 25 lines, deletes a line when it

receives the sequence ESC [1 M, and when the insert key is pressed emits the sequence ESC [@.

```
xon,cols#80,lines#25,dll=\E[1M,kich1=\E[@,
```

Delays

Some string capabilities require a delay in order for the terminal to complete processing a sequence. These delays may be specified in milliseconds by enclosing the delay in \$<...> and placing it anywhere in the capability. For example the capability el=\EK\$<3>, specifies that in order to erase a line the character sequence ESC K must be sent followed by a 3 millisecond delay.

Parameters:

String capabilities also require parameters in some cases. For example, the "rep" sequence (repeat character C, N times) takes two parameters (the character, and the number of repetitions). Parameters are handled using a stack and special % codes. Operations use postfix (Reverse Polish) notation.

A subset of the % codes is presented in the following table. For a complete description, see your UNIX documentation or "man" pages.

Code	Operation
%pN	push Nth parameter
%'C'	push character C
%{N}	push number N
%%	output %
%d	pop top item and output as number
%c	pop top item and output as character
%+	pop top two items and push #1 + #2
%-	pop top two items and push #1 - #2
%*	pop top two items and push #1 * #2
%/	pop top two items and push #1 / #2
%m	pop top two items and push #1 MOD #2
%&	pop top two items and push #1 AND #2
%	pop top two items and push #1 OR #2
%^	pop top two items and push #1 XOR #2
;%?C	if C
;%tB	then B (required by %?)
;%eC;%tB	elseif C then B

```
%eB      else B
%;       endif (required by %?)
```

A simple use of the param notation is:

```
cuf=\E%p1%dC,
```

which defines the cursor forward N columns sequence as:

```
ESC number C
```

A more complex sequence is "sgr" (set attributes). Its parameters are interpreted as follows:

```
pN      Attribute to turn on
p1      standout
p2      underline
p3      reverse
p4      blink
p5      dim
p6      bold
p7      invis
p8      protect
p9      altcharset
```

The following setting:

```
sgr=\E[0%?%p1%t;7%;%?%p2%t;3%;%?%p3%t;4%;m
```

would cause the set attributes sequence to behave as follows:

```
Always emit ESC [ 0
If parameter #1 is set emit ; 7
If parameter #2 is set emit ; 3
If parameter #3 is set emit ; 4
Always emit m
```

One final set of entries which require discussion, are the line drawing sequences. Four entries are required to support line drawing characters.

enacs

a sequence that sets up the terminal so that it can switch into an alternate character set.

smacs

switches to an alternate character set

rmacs

returns to the default character set

acsc provides the character mapping for the graphics characters.

The alternate character set mappings are based on a DEC VT-100 terminal's alternate character set. Here are the standard VT-100 characters and their associated glyph names.

Glyph Description	VT100 Character
arrow pointing right	+
arrow pointing left	,
arrow pointing down	.
solid square block	0
lantern symbol	l
arrow pointing up	-
diamond	`
checker board (stipple)	a
degree symbol	f
plus/minus	g
board of squares	h
lower right corner	j
upper right corner	k
upper left corner	l
lower left corner	m
plus	n
scan line 1	o
horizontal line	q
scan line 9	s
left tee	t
right tree	u
bottom tee	v
top tee	w
vertical line	x
bullet	~

The content of the "acsc" entry is a list of characters. To create this list start with the first VT100 character in the above list and alternate with the character required by your terminal. For example if your terminal uses "<" for the right arrow, ">" for the left arrow, "v" for the down arrow and "#" for the solid square block your "acsc" sequence would begin: acsc=+<,>.v0# ...

The comma character (,) needs a backslash character to escape its special meaning in the terminfo source.

If our terminal required the sequence

```
esc ( B esc ) 0
```

to set up its alternate character set, required a ^N to switch to the alternate character set, required a ^O to switch back to the default character set and followed VT100 character mapping conventions we would expect to see entries similar to the following in its terminfo source file.

```
acsc=++\,\,\,.00II--
` `aaffgghhjkkllmmnnooqqssttuuvvwvwx,
enacs=\E(B\E)0,
rmacs=^O,
smacs=^N,
```

The following shows a sample terminfo source file.

```
# Reconstructed via infocmp from file:
/usr/share/lib/terminfo/v/vt100
vt100|vt100-am|dec vt100 (w/advanced video),
am, mir, msgr, xenl, xon,
cols#80, it#8, lines#24, vt#3,
acsc=++,,.00II--` `aaffgghhjkkllmmnnooqqssttuuvvwvwx,
bel=^G, blink=\E[5m$<2>, bold=\E[1m$<2>,
clear=\E[H\E[J$<50>, cr=\r, csr=\E[%i%p1%d;%p2%dr,
cub=\E[%p1%dD, cub1=\b, cud=\E[%p1%dB, cud1=\n,
cuf=\E[%p1%dB, cuf1=\E[C$<2>,
cup=\E[%i%p1%d;%p2%dB$<5>, cuu=\E[%p1%dB,
cuu1=\E[A$<2>, ed=\E[J$<50>, el=\E[K$<3>,
ell=\E[1K$<3>, enacs=\E(B\E)0, home=\E[H, ht=\t,
hts=\EH, ind=\n, ka1=\EOq, ka3=\EOs, kb2=\EOr, kbs=\b,
kc1=\EOp, kc3=\EOn, kcub1=\EOD, kcud1=\EOB,
kcuf1=\EOC, kcuu1=\EOA, kent=\EOM, kf0=\EOy, kf1=\EOP,
kf10=\EOx, kf2=\EOQ, kf3=\EOR, kf4=\EOS, kf5=\EOT,
kf6=\EOu, kf7=\EOv, kf8=\EOl, kf9=\EOw, rc=\E8,
rev=\E[7m$<2>, ri=\EM$<5>, rmacs=^O, rmkx=\E[?11\E>,
rmso=\E[m$<2>, rmul=\E[m$<2>,
rs2=\E>\E[?31\E[?41\E[?51\E[?7h\E[?8h, sc=\E7,
sgr=\E[0??p1p6%|&t;1%;??p2&t;4%;??p1p3%|&t;7%;??p4
&t;5%;m??%
p9&t^N^e^O%;,
sgr0=\E[m^O$<2>, smacs=^N, smkx=\E[?1h\E=,
```

```
smso=\E[1;7m$<2>, smul=\E[4m$<2>, tbc=\E[3g,
```

Once you have created a terminfo source file, you are ready to compile the entry into the terminfo database. This is accomplished using the UNIX terminfo compiler, "tic". This utility is described in detail in the standard UNIX documentation or "man" pages.

To use "tic" type at your shell prompt.

```
tic FILENAME
```

This compiles the source file and places the definition in the terminfo database.

If you do not wish to modify the default terminfo database, the environment variable "terminfo" can be used to specify a directory in which to place the compiled definitions. For example, if the terminfo environment variable were set to "/u/tipsrc/terminfo" then "tic" would place any compiled definitions in /u/tipsrc/terminfo.

curses - CRT Screen Handling Package - UNIX

curses is a standard UNIX package which accesses the terminfo database in order to get the correct definition for a given terminal. TIP/ix uses curses to control the terminal and to receive input from the terminal.

Two environment variables are used by curses in order to find the correct definition for your terminal.

TERM

TERM should be set to the correct terminal type for your terminal.

terminfo

If you are using the system default terminfo database, terminfo does not need to be set.

Otherwise, terminfo should be set to the directory containing the terminfo database (as described in the previous section).

Your system administrator should make sure that these two environment variables are set correctly.

aft - Active File Table

The Active File Table (**aft**) program displays information about files currently assigned to a TIP/ix session and the TIP/ix screen formats that appear in that session's MCS cache (screen format pool).

Syntax:

```

aft [ [*!]terminal name ]
aft [ [*!]user id ]
aft [ [*!]program name ]
    
```

Where:
user id

Display the active file table for this user.

terminal name

Display the aft for this terminal.

program name

Display the aft for the users that are running this program.

* Match anything that starts with the prefix.

! Match everything that does not start with the prefix.

If no parameter is supplied, aft displays the Active File Table for the session issuing the aft command.

Example:

```
aft pts013
```

Results in:

```

Unixware7 - TIP WorkStation
Session Edit View Tools Help
$ aft _TCP431
TIP/ix ver 1999/02/23 2.2 RO - 0140 (c) 1991-1999 Allinson-Ross Corporation

4 Active Files for user SCOTTC on term _TCP431 Seg 19

LFN      File      Lvl      I/O
=====
TQL01000 TQL$CTL   1         3 ISAM  FCS-GETKEYED Q90, Pid=1719
TQL01001 TIP$SYS   1         1 ISAM  FCS-OPEN Q242, Pid=1715
TQL01002 TIP$SYS   1         1 ISAM  FCS-OPEN Q242, Pid=1715
TQL01003 TIP$SEC   1         1 ISAM  FCS-OPEN Q242, Pid=1715

MCS Screens cached: TF$TQRU_ LAST-MCS-NAME=TF$TQRU_

$
    
```

Where:

LFN The logical file name. A file name preceded by "*" indicates the file was automatically opened by the system (not by an application calling FCS-OPEN).

File Name of the file definition from TIP\$SYS. See smfile.

Lvl Program stack level where file was assigned.

I/O I/O count for the file.

other Information following the I/O count include: an indication of the type of file, and the queue id and process id of the FCS server handling the I/O for this file

An additional line may appear (as in the example) to list the TIP/ix screen formats that are present in the user's screen format pool (MCS cache). These screen formats are the ones most recently used at that terminal.

If you try to match a program name, a match only occurs if the program you want is active at the highest stack level of a TIP/ix session.

For example, run "tcm" then select "user id definition". In this case, tcm is running at stack level 1 and smuser is running at stack level 2.

So "aft smuser" would match this session but "aft tcm" would not.

apb - All Points Bulletin

The **apb** transaction broadcasts a message to all currently active terminals.

The message text is sent as an unsolicited message. A BEL character is sent to the terminal to indicate the presence of a message. If a recipient of a message is running TIP/fe then the bell can be seen (single flash) and/or heard (single beep) and the TIP/fe status bar will indicate a message is available (after any key at that TIP/fe session is pressed).

The message is displayed on the screen when MSG WAIT is pressed at the cursor location at the time MSG WAIT is pressed! It is a good idea to first move the cursor to an area of the screen that is not in use. The message will be prefaced by the user id and terminal name of the sender.

Syntax:

```
apb text
```

Where:

text The text of the message (64 characters maximum). You don't need quote marks.

Example:

```
apb System will shutdown at 2:15 for 30 min.
```

calendar - Display Calendar

The **calendar** program displays the calendar for a specified month and year. By default, **calendar** displays the current month and year.

Function keys may be used to move to preceding or following months. A function key is provided to toggle the display between Julian (day of year) format and standard Gregorian calendar format.

When the current month is displayed, the current day is highlighted.

Syntax:

```
calendar [ month [, year ] ]
```

Where:

month

The month desired (default is current month). This value must be in the range of 1 through 12 inclusive.

year The year desired (default is current year).

If the value specified is less than 100, the calendar program interprets the year as 1900 plus the value specified.

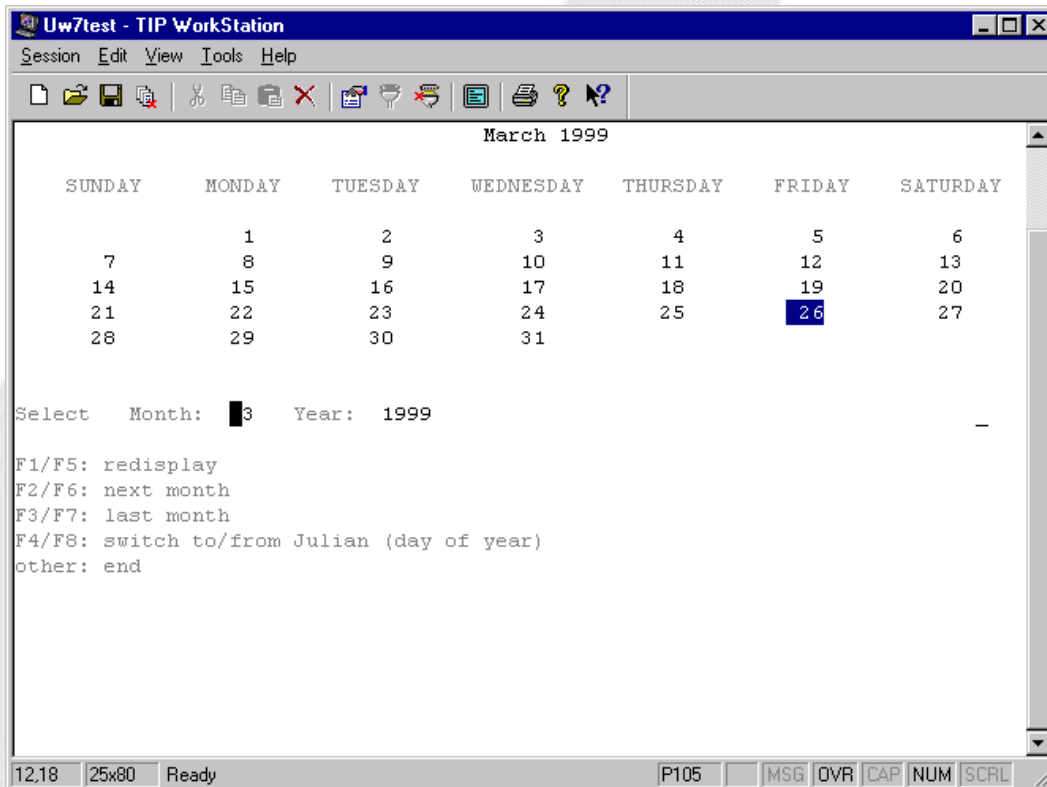
Years prior to 1753 are not accepted by this program.

Years after 3999 are not accepted by this program because the year 4000 represents the point at which an additional leap year error correction must take place.

Function Key Usage:

Key	Description
F1 or F5	Refresh screen.
F2 or F6	Display next month.
F3 or F7	Display previous month.
F4 or F8	Toggle between GREGORIAN and JULIAN modes.
Other	Terminate program.

Example of calendar output:



cobxref - COBOL Cross Reference Report

The **cobxref** utility scans COBOL source program files and generates a cross-reference report containing lists of included COBOL copybooks and symbols and the line numbers they appear on.

Syntax:

```
cobxref [options] [file1] [file2] [etc..]
```

Where:

options

can be selected for the following list:

-f{file}

The output cross-reference report filename.
The default is x-refrnc.arc

-h Display cobxref help.

-p Specify copybook search path.

-c Specify copybooks to search for (separate copybooks with commas.)

-d Finds every copybook.

-s Specify symbols to search for (separate symbols with commas.)

-t Finds every symbol (in all included copybooks.)

-u Specify copybooks to search for. This finds nested symbols (separate copybooks with commas.)
-u overrides (disables) any other searching option.

file 1...file n

Specify COBOL files to search. At least one file must be specified.

If no search options are specified the **cobxref** will search for all COPY modules and symbols referenced by the specified programs.

The report created will have at least one line for each copy module and symbol that is being cross-referenced for each module that is being cross-referenced.

The format of the lines in the report is:

```
<format><copy module or symbol name><source module><line #><ref flag>
```

Where:

format

A single digit indicating the nature of the reference to the copy module or symbol.

- 1 This line records a single reference to the copy module or symbol within the PROCEDURE DIVISION of the source module.
- 3 This line records a single reference to the copy module or symbol prior to the PROCEDURE DIVISION of the source module. The reference on this line number will be the definition of the symbol (data type and size).
- 6 Copy module or symbol is not reference in this source module.

copy module or symbol name

Name of the copy module or symbol name.

source module

Name of source module that this reference refers to.

line # Line number in the source module where the copy module or symbol was referenced. If the format type is “6” then no line number will be displayed (since this indicates no reference occurs within the source module).

ref flag

is one of the follwong:

D This reference is prior to the PROCEDURE DIVISION of the source module. The reference on this line number will be the definition of the symbol (data type and size).

<blank>

This reference is within the PROCEDURE DIVISION of the source module.

Examples:

```
cobxref *.cbl
```

This example finds every copybook and every symbol referenced in the any file with an extension of “.cbl” in the current working directory.

```
cobxref -sACCT-NAME,ACCT-AMOUNT -facctxref *.cbl
```

This example finds all references to the symbols ACCT-NAME and ACCT-AMOUNT in any file with an extension of “.cbl” in the current working directory writes the resulting report to the file “acctxref” in the current working directory.

connect - Connect to another TIP System

The **connect** program establishes a remote session with another TIP system. A *LOCAP* name is used to identify TIP systems when there are

multiple TIP systems in a network. Each TIP system establishes its own *LOCAP* name with the parameter *LOCAP* in the **tipix.conf** file.

The **connect** utility can only establish remote sessions with other TIP systems in the network if their *LOCAP* identifiers have been correctly defined via the [smlocap](#) system maintenance program. A remote TIP system could be on the same UNIX host or on another UNIX system in the network.

The remote session established by **connect** allows the terminal user to execute a number of subsequent transactions on another TIP system. The remote session remains in effect until the user exits the remote session (**fin**) or executes the **remove** transaction (the opposite of **connect**).

The security level and active groups settings for a remote session, depend on the *LOCAP* definition (on the remote *LOCAP*) of your local *LOCAP*. See the [smlocap](#) utility for details.

Syntax:

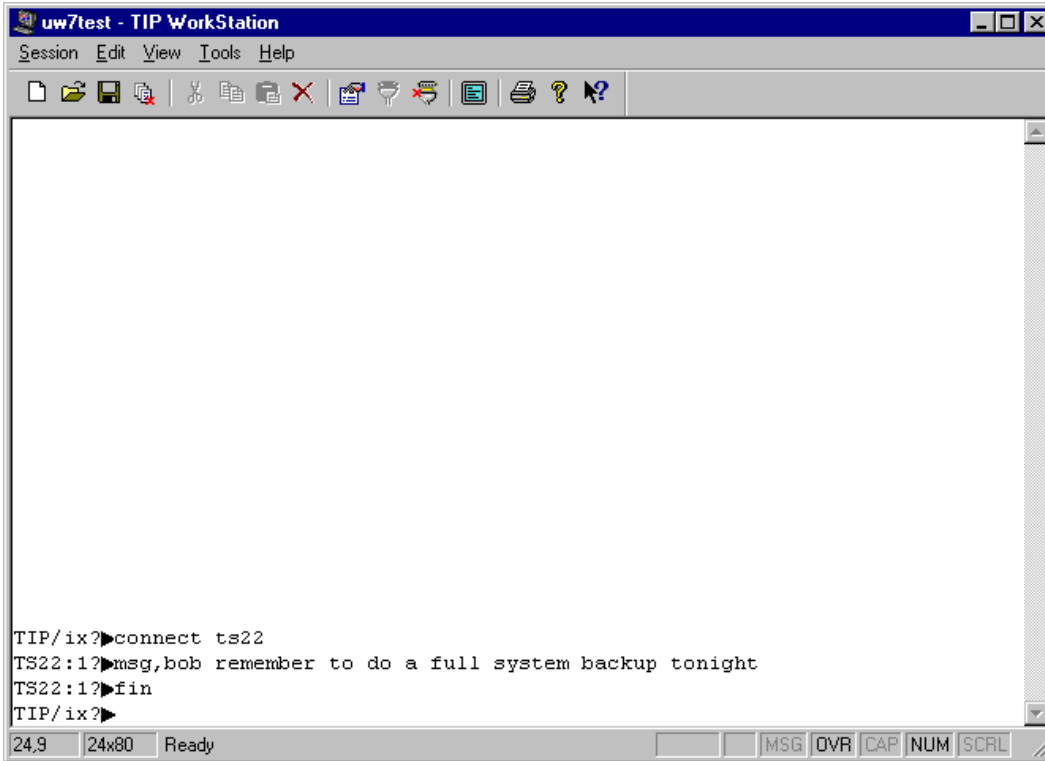
```
connect LOCAP
```

Where:**LOCAP**

The *LOCAP* identifier of the TIP system that a remote session is to be established with.

Example:

The following example illustrates a brief conversation with the TIP system with a *LOCAP* identifier of "TEST". Because `PROMPT=LOCAP` was specified in the **tipix.conf** file on the remote TIP system, the *LOCAP* name "TEST" appears in the in the TIP prompt. The ":1" indicates that the connect level (hop count) is one.



```

uw7test - TIP WorkStation
Session Edit View Tools Help
TIP/ix?▶connect ts22
TS22:1?▶msg,bob remember to do a full system backup tonight
TS22:1?▶fin
TIP/ix?▶
24.9 24x80 Ready MSG OVR CAP NUM SCRL

```

Error Conditions:

The CONNECT program outputs an error message if the attempt to make a connection fails. Some of the error conditions are:

- Invalid LOCAP name.
- Currently connected with remote LOCAP.
- LOCAP is currently closed.
- Error occurred at remote LOCAP.

dcheck - ISAM File Index Integrity Check

The **dcheck** program may be used to display the configuration of a file, check the integrity of a file's index, repair a faulty index, rebuild an index from the file data, and view the index contents. This program is part of the D-ISAM (C-ISAM compatible) file system (from Byte Designs Inc) that is used by TIP/ix to implement keyed files. This utility can only be used with files created by TIP/ix using the default FCS server (**tipfcs**) or by files created by batch Micro Focus COBOL programs.

This program may be useful in displaying a file's configuration information (record size, key structure, ...) in preparation for defining the file to TIP/ix (via **smfile**).

There is a utility called 'dpack' which can be used to rebuild an ISAM file. Also check out 'isreorg' which can rebuild ISAM files and change the file format.

Syntax:

```
dcheck - isam file integrity check/repair utility

usage - dcheck [-hifbB] <isamfile> [...]

options - h - display isam header information only
          i - just check indexes, ignore data file
          f - fix corrupt indexes
          b - rebuild all indexes
          B - rebuild specific index
```

Worldwide Copyright (c) Byte Designs Ltd (2009) - version 7.0

Example:

The following example shows the file structure for the file "tiptrm" in the current working directory (/tipix/tipfiles). It reports that the file has six keys and proceeds to display the details of each key. The description "UNIQ" means that no duplicates are allowed for the key. The description "DUPS" means that duplicate values are allowed for the key. To the right of the description are three numbers indicating the key location, key length, and key type respectively. For files created with TIP/ix, the key type will be "0".

```
>dcheck tiptrm
tiptrm structure
  data record length: 100
  index block size: 1024
  index dup width: 2
  index 1: uniq char@0/8
  index 2: uniq char@8/25
  index 3: uniq char@33/24
  index 4: dups char@57/4
  index 5: dups char@61/8
  index 6: dups char@69/8
  data file: 902 slots allocated, 886 free
  index file: 146 slots allocated, 133 free

checking data..ok
checking index 1..ok
checking index 2..ok
checking index 3..ok
checking index 4..ok
checking index 5..ok
checking index 6..ok
```

Error Conditions:

The **dcheck** program will display a message (on stdout) if an error occurs. The format of the message is:

```
error(<linenum>) <function> = <errornum>
```

Where:**linenum**

Source code line number in dcheck where error occurred.

function

The function that returned the error.

errornum

The error number returned by the function. Error numbers less than 100 normally emanate from UNIX and can be found in /usr/include/sys/errno.h. The descriptions of the D-ISAM error numbers are listed in the following table.

D-ISAM Error Number	Description
100	An attempt was made to (re)write a duplicate where duplicates are prohibited, or an attempt was made to REWRITE(F) where the primary key permitted duplicates.
101	The fd parameter does not reference an opened file.
102	One of the arguments has a value with no defined meaning.
103	The values of key are not valid.
104	All ISAM file descriptors are used, you cannot open any more files
105	The ISAM file is corrupted; it must be repaired with DCHECK.
106	Exclusive access to the file is not possible.
107	Another process has a read-only lock on the requested record.
108	The value of key has already been established as a key.
109	The requested function may not be performed on the primary key, as requested.
110	The beginning or end of the file has already been reached.
111	No record was found to match your request.
112	There is no "current" record set at this time.
113	The file has been exclusively locked by another process, or if trying to establish an exclusive lock, another process is using the file.

- 114 The name given for the file is too long or contains unacceptable characters.
- 115 The lock file cannot be created. Presently not used by D-ISAM.
- 116 malloc() cannot allocate the request. Usually means out of memory, but possibly the allocation list is corrupted.

Note :

Since build 2012/09/19 2.5 R0 - 0228, TIP/ix is using D-ISAM version 7. If you are using a recent Micro Focus COBOL compiler some customers have found that MF COBOL ISAM and D-ISAM 7 do not work well together and sometimes result in ISAM data file corruption. We recommend that you compile your batch programs and include the D-ISAM library instead of using the MF COBOL default ISAM file handler.

To use the D-ISAM binary library released with TIP/ix for your batch MicroFocus COBOL programs use the following compile options:

```
-L $TIPROOT/lib -m ixfile=cixfile +l disam
```

If you use these parameters in a 'makefile' then it is coded as:

```
-L $(TIPROOT)/lib -m ixfile=cixfile +l disam
```

The D-ISAM software comes with two utilities: dcheck & dpack

dcheck can be used to verify an ISAM file

dpack can be used to rebuild an ISAM file

Do not use the Micro Focus 'rebuild' utility on ISAM files which will be accessed by TIP/ix. Use the 'dpack' utility to verify and rebuild the ISAM file when needed.

ddu - Disk Display and Update Utility

The disk display utility (**ddu**) allows you to view and update the contents of files or the TIP/ix **GDA** (global data area).

The following types of files can be accessed by ddu:

- TIP/ix data files: Indexed, Direct (Relative)
- TIP/ix edit buffers
- TIP/ix dynamic files
- UNIX flat files

Note: To view a TIP/ix sequential file use [dfu](#).

The disk display and update (**ddu**) utility allows you to:

- page through the file a record a time (forward or backward)
- jump to a specific record (key value or record number)
- scroll through a particular record (if the record is larger than a screen full of data)
- jump to a specific offset in the record (useful for large records)
- change the file being viewed
- change the key of reference (for multi-keyed files).
- update/add a record
- print record(s)
- search record(s) for a string.

Syntax:

```
ddu[, [r][s]] filename
ddu[, [r][s]] edit buffer
ddu[, [r][s]] dynamic file
ddu[, [r][s]] [./|../]unix file [record-size]
ddu[, [s]] GDA
gda
```

Where:

- r** Treat the file as read-only (to protect yourself from accidentally updating it). The GDA is always displayed in read-only mode.
- s** Use a small-screen format to display the file. This speeds up screen painting on some (remote) terminals.

filename

The file you want to view or update. For example: myfile.

First, **ddu** attempts to access the file as a TIP/ix file.

If that fails, **ddu** attempts to access it as a UNIX flat file.

(A TIP/ix file is a file that has been defined to TIP/ix with [smfile](#) and [smsec](#).)

edit buffer

The edit buffer you want to view or update. Specify the group and buffer name. Separate the parts with a slash, comma, or space. For example:

group/buffer or

group,buffer or

group buffer (but not group, buffer)

Use the “status e” command to obtain a list of edit buffers.

dynamic file

The dynamic file you want to view or update. Specify the user id, catalog, and filename. Separate the parts with a slash, comma, or space. For example:

user id/catalog/filename

You must supply all 3 components of the dynamic file name. Use “status d” command to obtain a list of dynamic files.

unix file

Prefixing the file name with “/”, “./” or “../” will force the file name to be treated as a Unix file. For example:

/absolute/myfile

./myfile

../myfile

Otherwise, **ddu** will first attempt to open the file as one of the TIP/ix file types above.

The absence of a leading slash results in a relative path name (as opposed to absolute path name). For example:

subdir/yourfile

subdir/subsub/file

GDA Display the Global Data Area of TIP/ix. Can be requested by using GDA as either the first parameter or the transaction code.

record-size

This parameter only applies to UNIX files. This arbitrary value is used to allow record numbers to be used to

navigate through UNIX files. A request for the *n*th record causes **ddu** to display file data from offset (*n*-1) * record size) into the file.

The default record size used for navigating through a UNIX file is 1024.

To access a UNIX file “foo” when there is also a TIP/ix file “FOO” prefix the UNIX file name with “./”. For example:

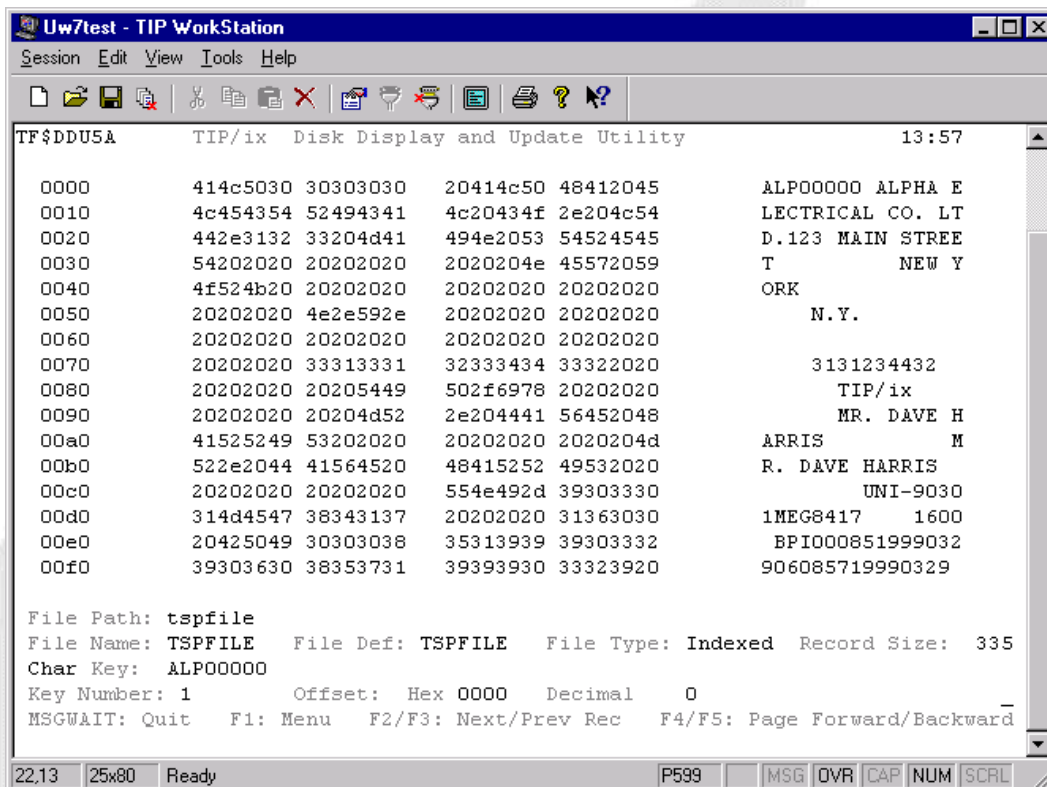
```
ddu ./foo
```

If you access a UNIX file to which you have read-only access, the “r” option is assumed, and **ddu** displays a message indicating that you are operating in read-only mode.

Upon entering **ddu**, the first record in the file is displayed.

Example:

If you enter “**ddu tspfile**”, the following screen is displayed:



```

Uw7test - TIP WorkStation
Session Edit View Tools Help
TF$DDUSA TIP/ix Disk Display and Update Utility 13:57
0000 414c5030 30303030 20414c50 48412045 ALP00000 ALPHA E
0010 4c454354 52494341 4c20434f 2e204c54 LECTRICAL CO. LT
0020 442e3132 33204d41 494e2053 54524545 D.123 MAIN STREE
0030 54202020 20202020 2020204e 45572059 T NEW Y
0040 4f524b20 20202020 20202020 20202020 ORK
0050 20202020 4e2e592e 20202020 20202020 N.Y.
0060 20202020 20202020 20202020 20202020
0070 20202020 33313331 32333434 33322020 3131234432
0080 20202020 20205449 502f6978 20202020 TIP/ix
0090 20202020 20204d52 2e204441 56452048 MR. DAVE H
00a0 41525249 53202020 20202020 2020204d ARRIS M
00b0 522e2044 41564520 48415252 49532020 R. DAVE HARRIS
00c0 20202020 20202020 554e492d 39303330 UNI-9030
00d0 314d4547 38343137 20202020 31363030 1MEG8417 1600
00e0 20425049 30303038 35313939 39303332 BPI000851999032
00f0 39303630 38353731 39393930 33323920 906085719990329

File Path: tspfile
File Name: TSPFILE File Def: TSPFILE File Type: Indexed Record Size: 335
Char Key: ALP00000
Key Number: 1 Offset: Hex 0000 Decimal 0
MSG$WAIT: Quit F1: Menu F2/F3: Next/Prev Rec F4/F5: Page Forward/Backward
22.13 25x80 Ready P599 MSG OVR CAP NUM SCRL

```

The column on the left represents the offset into the record, the hexadecimal representation of the data appears in the middle of the screen, and the character representation of the data appears on the right of the screen. Any non-displayable characters (such as a packed or binary data, line feeds, or LOW-VALUES) are represented in the character portion of the display as a period.

Some file attributes are displayed at the bottom of the screen. See the following table.

Attribute	Description												
File Path	<p>For TIPFCS data files this is the “Label/Path” field from the file definition (see smfile). For the file being displayed.</p> <p>For Edit buffers or dynamic files the parameters used to access the file are displayed (separated by a slash).</p> <p>For UNIX files this is the path supplied upon invoking ddu.</p>												
File Name	TIPFCS name used to access the file (see smsec).												
File Def	Name of the file definition referenced by the security entry used to resolve the request to access “File Name”.												
File Type	<p>Type of file being displayed as follows:</p> <table border="0"> <tr> <td>Indexed</td> <td>TIPFCS Indexed file</td> </tr> <tr> <td>Direct</td> <td>TIPFCS Direct file</td> </tr> <tr> <td>Edit</td> <td>TIPFCS Edit buffer</td> </tr> <tr> <td>Dynamic</td> <td>TIPFCS Dynamic file</td> </tr> <tr> <td>GDA</td> <td>TIP/ix Global Data area</td> </tr> <tr> <td>UNIX</td> <td>UNIX flat file</td> </tr> </table>	Indexed	TIPFCS Indexed file	Direct	TIPFCS Direct file	Edit	TIPFCS Edit buffer	Dynamic	TIPFCS Dynamic file	GDA	TIP/ix Global Data area	UNIX	UNIX flat file
Indexed	TIPFCS Indexed file												
Direct	TIPFCS Direct file												
Edit	TIPFCS Edit buffer												
Dynamic	TIPFCS Dynamic file												
GDA	TIP/ix Global Data area												
UNIX	UNIX flat file												
Record Size	<p>For TIPFCS files, this is the size of the records in the file.</p> <p>For GDA this is the size of the entire GDA. The value displayed will be truncated if the GDA size exceeds 5 digits.</p> <p>For UNIX files this is the size used to navigate the file. When the nth record is requested ddu displays the file from offset (n-1) * recsize into the file.</p>												
Char/Hex	<p>Format of the Key field. The key field will be displayed in character format unless it contains non-displayable characters.</p> <p>Changing the first character to an “H” and exiting this field causes ddu to display the hexadecimal representation of the record key (indexed files) or record number (direct files).</p> <p>Change the first character to an “H” to request</p>												

	a record by a hexadecimal key or record number.
Key	Key or record number of the record currently displayed. Change this value to request another record from the file.
Key Number	Fill in the number of the key of reference to use to access records in the file. Only applicable to Indexed files.
Offset	Fill in the desired Hex or Decimal offset within the record.

ddu Function Keys

The function keys used by ddu are:

Key	Description
F1	Display function menu bar at top of screen. Same as "Activate Menu Bar" key. See section in this manual on "Terminal Interface".
F2	Display next record in the file.
F3	Display previous record in the file.
F4	Display next screen of data for current record.
F5	Display previous screen of data for current record.
F6	Invoke Update mode. Pressing F6, toggles between Character and HEX update mode.
F7	Invoke Add mode. Pressing F7, toggles between Character and HEX add mode.
F8	Search current file for a string. Searches are case sensitive.
F9	Request to print record(s).
F10	Delete current record key.
MSG	Exit ddu.
WAIT	

Display a Record

For direct files fill in the key with the record number to be displayed. The "Char" preceding the Key field indicates that the record number is expected to be a positive integer. To enter a hexadecimal number, change the first character of "Char" (key type) to an "H". If the key type is changed to "Hex" prior to changing the Key field then the Hexadecimal

record number for the current record will be displayed upon exiting the key type field.

For an indexed file, the Current Index and the Key fields can be filled in to request a record to be displayed from the file. Fill in Key Number with the number of the key to use to access the file (see [smfile](#) for file configuration information) and fill in Key with the key value to use in retrieving the record from the file. The disk display and update (**ddu**) utility will display the next record in the file with a key value equal to or greater than the key value that was entered. This relieves you of the burden of having to enter a complete key value.

To use a character value for a key, set the first character of the field preceding the Key to "C" (Char).

To use a hexadecimal value for a key, set the first character of the field preceding the Key field to "H" and enter the hex value for the key of the desired record in the Key field. If the key type is change to "Hex" prior to changing the Key field then the Hexadecimal representation of the key of the current record is displayed upon exiting the key type field.

The key value entered is case sensitive when entered as character data. Therefore, if you want a key that is uppercase you should fill in the Key field in uppercase. This allows you to access records in files with mixed case keys.

Specify offset within record

The **ddu** record display shows 256 bytes of record data. Many records will have a record size larger than this. Use the Offset fields (Hex and Decimal) to set position in the record to start displaying data from. If both the Hex and Decimal fields are changed then the Hex offset will be used as the position to start displaying from.

Add a record

DDU provides a basic facility for adding records to a file. This function is only supported for indexed files, direct (relative) files, and edit buffers. When adding records you cannot add records when viewing a UNIX flat file, a TIP/ix dynamic file, or the TIP/ix GDA.

The ability to add records may be useful for entering test records. However, TQL provides a much more advanced and fully featured environment for adding records to a file. See the TIP/ix TQL reference for details on how to create and use TQL programs.

Press **F7** from the main record display to switch to "Add record mode". The cursor will move to the character display portion of the screen, which will now be unprotected to allow you to enter the data for the record to add. The data for the currently displayed record remains on the screen

and serves as a base from which to build the new record. After entering the data for the new record press **XMIT** to add the record.

Press **F7** to toggle between the character display and the hexadecimal display. Any changes made in one mode are preserved when switching to the other mode. This means that you can enter some data on the character display, press **F7**, and enter some data on the hexadecimal display. You can switch between modes as often as necessary but the changes only take affect (i.e. record is added) when you press **XMIT**.

This function can also be requested from the *Menu Bar*. Press **F1** (from the main record display) to display the menu bar. Press "**A**" or tab to "**Add**" and press **ENTER** or **XMIT** to request the add record function. The cursor will move to the character display portion of the screen, which will now be unprotected to allow you to enter the data for the record to add. After making the changes on the screen press **XMIT** to add the record.

For direct (relative) files, the record is added past the last record in the file. Therefore, if there are 16 records in the file then the record added will be record number 17.

For indexed files, the record is added by composing the key(s) from information in the data area for the record. The add will fail if a duplicate key results for any key configured to not allow duplicates. See [SMFILE](#) for details on configuring files in TIP/ix.

Only 256 bytes of the record can be set with **ddu** when a record is added. For large records, it may be necessary to first add the record and then update the record to alter the portion of the record beyond the first 256 bytes.

Update a Record

The disk display and update (**ddu**) utility provides a basic facility for updating records in a file. This function can be used with files (other than sequential files). You cannot update the GDA with **ddu**.

The ability to update records may be useful for fixing data in files. However, TQL provides a much more advanced and fully featured environment for updating records in a file. See the *TIP/ix TQL reference* for details on how to create and use TQL programs.

After displaying the record you wish to update, press **F6** to switch to "Update record mode". The cursor will move to the character display portion of the screen, which will now be unprotected to allow you to alter the data for the current record. After making the changes on the screen press **XMIT** to update the record.

Press **F6** to toggle between the character display and the hexadecimal display. Any changes made in one mode are preserved when switching to the other mode. This means that you can enter some data on the character display, press **F7**, and enter some data on the hexadecimal

display. You can switch between modes as often as necessary but the changes are only applied to the record in the file (i.e. record is updated) when you press **XMIT**.

This function can also be requested from the *Menu Bar*. Press **F1** (from the main record display) to display the menu bar. Press “**U**” or tab to “**Update**” and press **ENTER** or **XMIT** to request the update function. The cursor will move to the character display portion of the screen, which will now be unprotected to allow you to alter the data for the current record. After making the changes on the screen press **XMIT** to update the record.

For indexed files, the record is added by composing the key(s) from information in the data area for the record. The add will fail if a duplicate key results for any key configured to not allow duplicates or if the key value is changed for any key configured to not allow changes. See [SMFILE](#) for details on configuring files in TIP/ix.

Only 256 bytes of the record can be altered when a record is updated. This is the amount of data that is displayed on the screen. For large records, it may be necessary to do multiple updates to change all portions of the record. The update function allows changes to the portion of the record being displayed. To start displaying data beyond the beginning of the record fill in the *Offset* field on the screen (either Hex or Decimal value) and press **XMIT**.

The record data is case sensitive. Enter the data in uppercase if the record requires the data to be uppercase.

Delete a Record

The disk display and update (**ddu**) utility provides a basic facility for deleting records in a file. This function can be used with files (other than sequential files). You cannot delete the GDA with **ddu**.

After displaying the record you wish to delete press **F10**. A message will be displayed asking you to press **F2** to confirm that it is OK to delete this record. Press **F2** to delete the record. Press **MSG WAIT** or any function key **other** than **F2** to cancel the delete function.

This function can also be requested from the *Menu Bar*. Press **F1** (from the main record display) to display the menu bar. Press “**D**” or tab to “**Delete**” and press **ENTER** or **XMIT** to request the delete function. Press **F2** to confirm that it is OK to delete this record. Press **MSG WAIT** or a function key **other** than **F2** to cancel the delete function.

See the file configuration information in [smfile](#) to determine whether the record will be logically deleted by setting a delete byte in the record or physically removed from the file.

Switch to Another File

It is possible to stop displaying the current file and switch to another file. Currently **ddu** only allows for switching to a data file defined with [smfile](#) and [smsec](#). It is not possible to dynamically switch to another UNIX file, edit buffer, or dynamic file. For these other files you must restart **ddu** requesting the name of the UNIX file, edit buffer, or dynamic file you wish to view.

When viewing a data file (defined with [smfile](#)), edit buffer, or dynamic file you can switch to another **data file** by filling in the *File Name* field and pressing **XMIT**.

Alternatively, to view a different file:

- press F1 to display the Menu Bar
- select Switch (by pressing transmit)
- Enter the name of the data file you wish to view (maximum of 8 characters) and press transmit. The current file will be closed and then **ddu** will attempt to display the first record of the file name you have entered. The file must be defined with **smsec** and you must have adequate security and group membership to access the file.

Search for a String

This function can be used to find a record containing a string or find the location in a record that contains a string.

Press **F8** to obtain the String Search screen overlay. Enter the string to search for, select the search range (current record, current record to end of file, entire file), and press **XMIT** to initiate the search.

This function can also be requested from the **Menu Bar**. Press **F1** (from the main record display) to display the menu bar. Press “**S**” or tab to “**Search**” and press **ENTER** or **XMIT** to obtain the String Search screen overlay. Enter the string to search for, select the search range (current record, current record to end of file, entire file), and press **XMIT** to initiate the search.

The search function is case sensitive so the search string **must** be entered in the correct case.

Print Records from File

This function can be used to print records in a format similar to that used by the **ddu** display.

Press **F9** to obtain the Print Record(s) screen overlay. Enter the TIPPRINT destination (see [smprint](#) for valid print file names), select the

records to be printed (current record, from current record for n records, entire file), and press **XMIT** to start printing.

This function can also be requested from the **Menu Bar**. Press **F1** (from the main record display) to display the menu bar. Press "**P**" or tab to "**Print**" and press **ENTER** or **XMIT** to obtain the Print Record(s) screen overlay. Enter the TIPPRINT destination (see [smprint](#) for valid print file names), select the records to be printed (current record, from current record for n records, entire file), and press **XMIT** to start printing.

GDA Support

If the file name is specified as GDA or the transaction code (PIB-TRID) is GDA then **ddu** displays the TIP/ix Global Data Area.

Record size will refer to the size of the GDA. This will be truncated if the GDA size is too large to fit in the record size field. See the file **\$TIPROOT/conf/tipix.conf** to determine the size of the GDA on your TIP/ix system or type **tipinstall** at the UNIX prompt.

To display different areas of the GDA fill in the desired (Hex or decimal) offset or use F4 and F5 to scroll forward and backward through the GDA.

Using the Menu Bar

To use the menu bar press **F1** on the main **ddu** display screen. The menu bar will appear at the top of the screen overlaying the first two lines of the main **ddu** screen. The first selection on the menu bar will be highlighted. To highlight another item on the menu bar use the tab key or press the capital letter in the desired item name. For example, to highlight "Add", press the letter "A". You can select the highlighted item by pressing either of "**ENTER**" or "**XMIT**".

The following screen shows how the menu bar overlays the bottom left corner of the record display screen (using TIP/ws), using TIP/fe the top 2 lines of the record display screen will be overwritten. The second line is a description of the currently highlighted item on the menu bar (top line, only in TIP/fe).

defkey - Define Function keys

The **defkey** utility program allows you to specify a character sequence that will be "painted" on the screen whenever a function key is pressed as a response to the standard system prompt:

After the sequence is painted on the screen, **defkey** outputs an auto transmit sequence to the terminal (this automatic XMIT may be

suppressed) - the net effect of this is to simulate the keying of that character sequence.

The definition of function key contents is specified by user group. The search for the appropriate function key contents follows the same sequence as the standard order of search in the catalogue: the user's private group is searched first, then elective groups and finally the universal group TIP\$Y\$.

By utilizing the **defkey** program, the user may assign character strings to function keys and make it simple to enter those commands.

The **defkey** program stores the function key definitions in a TIP/ix dynamic file with the name: group/FUNCTION/KEYS, where "group" is name of the group to which these definitions apply. The function key definitions are retained in this dynamic file until the definitions are explicitly deleted.

Syntax:

```
defkey [grp]
```

Where:

grp the name of the desired group. Default is the user's private group.

The **defkey** utility displays the following screen format:

This screen format is presented (along with the current function key contents - if any) for the specified group name.

The user may change any of the function key contents AND/OR change the group name. If the group name is changed, the **defkey** program assumes that a NEW function key set is being created from an existing group's definitions.

If the last non-blank character in a function key string is the backslash character \ the **defkey** program interprets that as a signal that the automatic XMIT feature is NOT desired when that function key is selected. If that function key is pressed, the character string (without the \) is painted on the screen and the terminal operator is then able to make any desired alterations or additions before pressing **XMIT**.

To delete all function key definitions, press the **F4** key. The **defkey** program displays a message on the screen instructing you to press **XMIT** to confirm that you want to scratch the **defkey** function key file. Pressing any function key other than **XMIT** cancels the delete operation.

Example:

```
defkey edp
```

This command presents the function key definitions currently in effect for the group "edp" and allows the user to change the definitions.

dfu - Dump File Utility

The **dfu** utility displays the contents of a file on standard output (in both hex and display format). The contents of the file can be interpreted as either ASCII or EBCDIC data.

This utility is a programming aid for testing and debugging. For a more fully functional and interactive file display program, see the section on "[ddu](#)" in this manual.

One use of **dfu** is to display data in a file to determine if a program has altered it correctly.

Syntax:

```
dfu fname [[dflag] ... [dflag]]
```

Where:

fname

The file name.

dflag Utility flags, as follows:

- A** Displays characters in ASCII mode. This is the default setting.
- E** Displays characters in EBCDIC mode.
- nnH** Where "nn" is the starting/stopping hexadecimal location.
- nnD** Where "nn" is the starting/stopping decimal location.
- 16B** Dump 16 bytes per line. This is the default setting.
- 32B** Dump 32 bytes per line.

Example:

The following example displays "file.ext", interpreting readable characters as EBCDIC, and prints 32 bytes per line.

```
dfu file.ext E 32B
```

The following example displays "file.ext" from hexadecimal location 400 to hex location 800 in default ASCII and prints 16 bytes per line.

```
dfu file.ext 400H 800H
```

The following screen is an example of the information displayed by **dfu** after the command

```
"dfu.exrc" is given:
```

Additional considerations:

If you are not sure which exact file locations to display, and would like to page through the file, execute **dfu** from your Unix shell to take advantage of output piping (and the use of the Unix "pg" command) or output redirection. For example:

```
dfu test.cbl | pg
```

die - Abort a Program

The **die** program forces an abnormal termination of a **user program** running at a terminal. The **die** program cannot abort the TIP/ix shell. Only the program at the highest stack level of a TIP/ix session can be aborted by the **die** program. Use [whoson](#) to find the appropriate parameter for the **die** command.

Syntax:

```
die [-F|-T|-U] [*]value
die S=segnum
```

Where:

- F** Use the value parameter to match files only.
- T** Use the value parameter to match terminals only.
- U** Use the value parameter to match users only.

value

You can limit the action of the die program by specifying a value. The default is "*" (die all TIP/ix sessions).

If the meaning of value is not restricted (by specifying -U, -T or -F), die attempts to match the value four ways (user id, terminal, file, and program).

- * Match anything that starts with specified value.
- ! Match anything that does not start with specified value.

user id

Abort the program running at the highest stack level of each TIP/ix session being operated by this user id.

terminal

Abort the program running at the highest stack level of each TIP/ix session being operated at this terminal.

file

Abort the program running at the highest stack level of each TIP/ix session where this file has been assigned. (see aft or whoson).

program

Abort this program on any TIP/ix session where it is running at the highest stack level for that session.

S=segnum

Specify the seg number of a single specific session to be aborted. You can get the seg number with the whoson command.

Example:

Abort the program being executed by user JOHN.

```
die -U john
```

Abort all user ids, terminal names, file names, and program names that begin with the prefix "ABC". From the TIP/ix shell:

```
DIE *ABC
```

From a Unix shell, you have to protect the * from expansion by Unix:

```
die \*ABC
```

Error Conditions:

die cannot find a matching user, terminal, file, or program.

Additional considerations:

- If more than one program would be aborted, the die utility lists them, and prompts you for confirmation.
- You cannot use die to abort the TIP/ix shell, instead use tipctl -k or purge.
- The die utility ignores any sessions that are connected to another TIP system — die only acts on sessions and programs running locally (on the current TIP/ix system).
- If you try to match a program name, a match only occurs if the program you want is active at the highest stack level of a TIP/ix session.

For example, run ["tcm"](#) then select "user id definition". In this case, tcm is running at stack level 1 and ["smuser"](#) is running at stack level 2.

So "die smuser" would match this session but "die tcm" would not.

dsf - Display File Information

The **dsf** transaction displays (or prints) a report about TIP/ix files. (Using **dsf** is the same as using [status f](#)).

Syntax:

```
dsf [-p printdest] [[*]file 1] ... [[*]file 7]
```

Where:**-p printdest**

Print the report to the TIP/ix print destination specified by printdest. If this parameter is not specified, the report goes to stdout.

(To define a print destination, see smprint.)

[*]file 1...[*]file 7

Specify up to 7 TIP/ix file definition names or prefixes. If no files are supplied, the report lists all TIP/ix files.

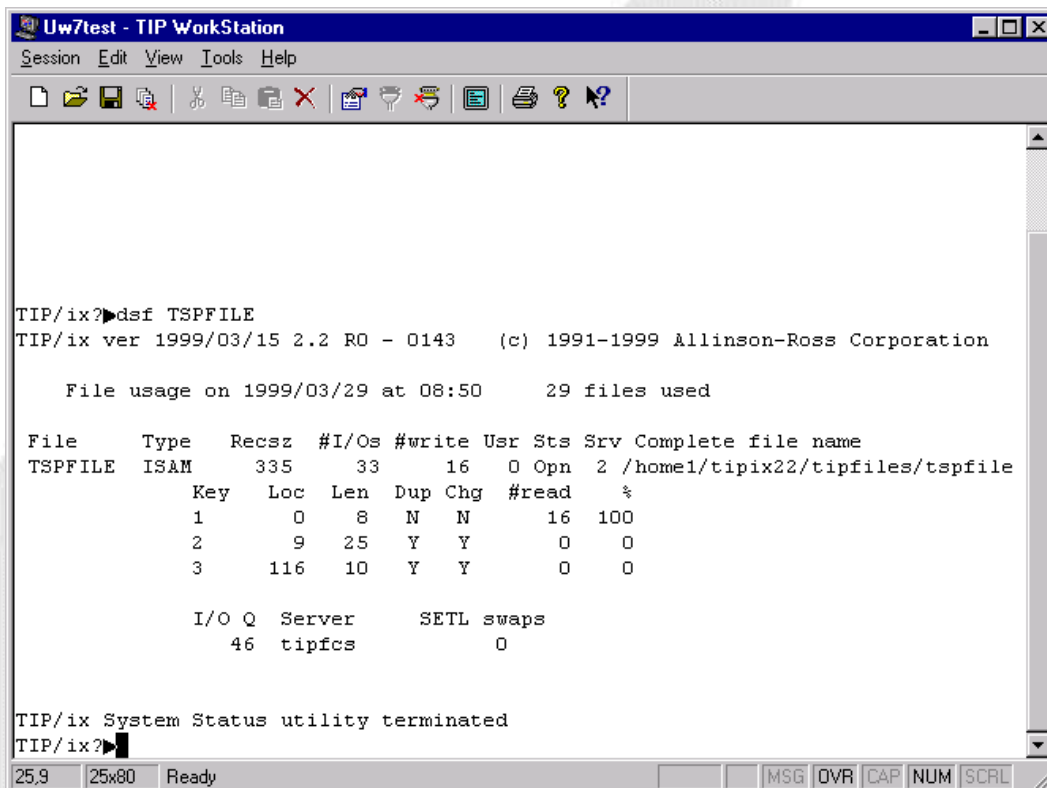
For example, "dsf *TIP" displays a report for all files with names beginning with "TIP".

Note: The file definition names specified here are defined with smfile. The TIPFCS names (defined by smsec) which applications use to access the files may be different.

Example:

```
TIP?>dsf tspfile
```

Gives a report like this:



```

Uw7test - TIP WorkStation
Session Edit View Tools Help
[Icons]

TIP/ix?>dsf TSPFILE
TIP/ix ver 1999/03/15 2.2 RO - 0143 (c) 1991-1999 Allinson-Ross Corporation

File usage on 1999/03/29 at 08:50 29 files used

File      Type  Recsz  #I/Os  #write  Usr  Sts  Srv  Complete file name
TSPFILE  ISAM   335    33      16     0  Opn  2  /home1/tipix22/tipfiles/tspfile
      Key  Loc  Len  Dup  Chg  #read  %
      1    0   8   N   N    16  100
      2    9  25   Y   Y    0   0
      3  116  10   Y   Y    0   0

      I/O Q  Server  SETL swaps
      46  tipics      0

TIP/ix System Status utility terminated
TIP/ix?>
25.9 25x80 Ready [MSG] [OVR] [CAP] [NUM] [SCRL]

```

Additional Consideration:

If the file has not been accessed by TIP/ix then the file and key information will be the same as in the [smfile](#) file definition. However, once

the file has been accessed, the information will correspond to the actual file information for the file (as reported by [dcheck](#) for D-ISAM files).

ebcasc - EBCDIC to ASCII Translation

In some countries, Unisys altered the OS/3 EBCDIC to ASCII translation to provide national language support.

The TIP/ix **ebcasc** program allows the Unix system administrator to specify how to perform this translation. The **ebcasc** program will create a file called **ebcasc.tbl** in the TIP/ix /tipfiles subdirectory. This file is an altered translation table that is used by *MCS* and the **Heritage Support Package** conversion programs when you use them to import your programs into TIP/ix.

Syntax:

```
ebcasc [-a aa ee] [-e ee aa] [-R] [-S] [-N nrc]
```

Where:

-a aa ee

Translate ASCII hexadecimal value AA to hexadecimal ee in EBCDIC.

-e ee aa

Translate EBCDIC hexadecimal value ee to hexadecimal AA in ASCII.

-R Reset the tables back to standard.

-S Set tables to the Swedish standard.

-N nrc

To select a standard National Character Set translation where nrc is:

BRI	English (United Kingdom).
NOR	Norwegian, Danish.
SWE	Swedish.
FRE	Belgian French.
GER	German.
ITA	Italian.
SPA	Spanish.
CFR	Canadian French.
SWI	Swiss French and German.

Example of ebcasc program use:

In the following example the EBCDIC character for hexadecimal C1 is translated to hexadecimal 41 in ASCII and hexadecimal C3 becomes hexadecimal 43 in ASCII.

```
ebcasc -e C141 -e C343
```

ej - Schedule a TIP/ix Shutdown

The **ej** program starts end-of-job processing for the TIP/ix system. TIP/ix does not allow any new logons to occur and waits for transactions that are executing to finish. The Eoj transaction also causes TIP/ix to set the status code **PIB-EOJ-PENDING** in the Program Information Block (PIB) of all TIP/ix programs that are executing. *Well-behaved TIP/ix programs should periodically check this flag to determine whether system shutdown has been requested.*

When there are no more users remaining on the system, the [SHUTDOWN](#) script (if one exists in the TIPROOT directory) will be executed. When the SHUTDOWN script has finished, all files are closed and TIP/ix terminates normally.

Syntax

```
ej [ timeout ] [ WAIT time-to-eoj ]
```

Where:

timeout The number of minutes to wait for transactions to complete. If a TIP/ix program is waiting (that is, via TIPTIMER), the wait time will be adjusted to be the lesser of the actual time remaining and this value.

WAIT This indicates that the ej program is to take effect on a delayed basis. That is, it will start after time-to-eoj minutes.

time-to-eoj The number of minutes to defer Eoj processing. This command runs under the UNIX prompt. If WAIT is specified, then Eoj will run in the background.

Examples

Eoj

Start End of Job processing. Once all users logoff, the SHUTDOWN script is run. The gentle approach.

Eoj 1

Reduce all TIPTIMERS to 1 minute. Any transactions that use TIPTIMER and that have not finished after one minute are terminated. Once all users logoff, the SHUTDOWN script is run.

Eoj WAIT 30

Force system shutdown in 30 minutes.

EOJ 2 WAIT 15

Reduce all TIPTIMERS to 2 minutes. Any transactions that use TIPTIMER and that have not finished after 2 minutes are terminated.

Wait another 13 minutes then force system shutdown.

fclose - Close Online Files

The **fclose** utility closes one or more online files and marks the files as "unavailable for online use". The files are not available to online programs until a subsequent "[FOPEN](#)" is issued (see the FOPEN utility documentation in this manual.)

This program does NOT operate interactively. Up to eight file names may be supplied on the command line. Each filename may be specified as a unique name or may use prefix notation.

This program may operate in an online or batch mode. Typically, a user may issue an **fclose** to close one or more files to the TIP online system and allow off-line "batch-style" processing to be performed on the files — for example, the running of backup scripts to create archive copies of files. In this case, the **fclose** program can be run off-line as a pre-backup function to ensure that the files are closed to the online system. Then, after the backups are complete, the [fopen](#) program can be run as a post-backup function to allow the files to be used online once more.

Syntax:

Style 1 - online (from the TIP/ix prompt)

```
fclose [,q|now] file 1 ... [,file 8]
```

Style 2 - batch (from the UNIX prompt)

```
fclose [-q][-f] file 1 ... [,file 8]
```

Where:

- q** This option (quiet) instructs fclose to suppress the usual informational messages that are displayed.
- now** This option allows you to close file(s) even if there are online programs (TIP or IMS) that have these file(s) open. Only way to reopen these files is to issue a fopen.
- q** This option (quiet) instructs fclose to suppress the usual informational messages that are displayed.

-f This option allows you to close file(s) even if there are online programs (TIP or IMS) that have these file(s) open. Only way to reopen these files is to issue a fopen.

file 1... The names of the files to be closed. At least one name must be specified. The file names represent file definitions (see smfile) and not TIPFCS file names (created and maintained by smsec).

Examples:

- Close the specified files from TIP/ix:
`fclose CUSTMAST,INVMASST,ORDENTRY`

- Close them from UNIX:
`fclose CUSTMAST INVMASST ENTRY`

- Close all files with a prefix of HR from TIP/ix:
`fclose *HR`

- Close them from UNIX:
`fclose *HR`

The backslash is to prevent UNIX from expanding the asterisk as a list of files.

- To close a file **even if it is being used**, use one of the following methods:

Method 1:

Fclose filename
Prevent new users from accessing file.

die filename
Abort the program at the current stack level for users that are accessing the file.

Purge filename
Abort any user sessions that are still accessing the file. These users will be taken out of the TIP/ix shell.

This forces an abort of any users of the file (while causing as little disruption as possible). The [die](#) and [purge](#) commands cause aborted transactions to be rolled back.

Method 2: (at the TIP/ix prompt)

Fclose now filename 1 .. filename 8
Closes file(s) even if there are online programs (TIP or IMS) that have these file(s) open. Only way to reopen these files is to issue a fopen.

Method 3: (at the Unix prompt)

Fclose -f filename 1 .. filename 8

Closes file(s) even if there are online programs (TIP or IMS) that have these file(s) open. Only way to reopen these files is to issue a fopen.

Return Codes

If you are using fclose from a UNIX shell script, fclose returns one of the following codes:

Code	Meaning
0	Successful.
1	Close is held pending. See Additional Considerations.
2	Item does not exist. The file name specified may not be defined to TIP/ix (there is no entry for that name entered via the smfile utility).
4	No response from TIPFCS (and TIP/ix is running).

Your Korn or Bourne scripts can use the shell parameter \$? to get the return code of the last program executed. For example:

```
#!/bin/ksh
export TIPROOT=/apps/tipix
export
PATH=$TIPROOT/bin:$TIPROOT/scripts:/apps/bin:.....
fclose tspfile
case $? in
0)
echo close was successful
;;
1)
echo close held pending
;;
2)
echo attempt to close file that is not defined
;;
4)
echo No response from TIPFCS (and TIP/ix is running)
;;
esac
```

Additional Considerations:

- The file close operation may be held pending until all users have relinquished control of the file. As soon as the fclose is issued however, the file is marked as "unavailable for online use" - new requests to use the file will not be honored.
- When running the fclose program in batch, the TIPROOT environment variable indicates which TIP/ix system is being referenced.
- Restarting TIP/ix does not cause files that have been marked as offline (with fclose) to be brought online. This is to prevent files from coming back online due to an unscheduled shutdown and startup of

TIP/ix. Once a file is closed with `fclose`, it remains offline until a subsequent `fopen`.

- This is different from TIP/30 where all files are marked as online when TIP/30 starts up. If you want to bring all TIP/ix files back online when TIP/ix starts up (TIP/30 style), do this:

```
fopen \*
tipctl b
```

- You can run `fclose` even when TIP/ix is not running. The next time TIP/ix starts up the file will be marked offline (unavailable).

The return code will be either 0 or 2.

fcsdebug - Support Utility

Control fcs debugging. Do not use this utility unless requested to do so by [Ingleneet](#) support personnel.

Syntax:

```
fcsdebug on
fcsdebug off
```

The `fcsdebug` command to just turn debugging on for an FCS server which handles a specific file.

Syntax:

```
fcsdebug on filename
fcsdebug off filename
```

Where:

filename
is the logical name of the data file.

fileview - browse, print, and delete files

The **fileview** transaction allows users to browse, print, and delete files (including UNIX spool files).

Syntax

```
fileview [directory] [,prefix] [,printer] [,hdr]
[,case] [,plen]
```

Where:

directory
Specify the directory to start with. The default is the current

directory or the directory given by the environment variable TIPSPLROOT.

prefix

Specifies the TIP/30-style filename prefix. The default is *.

printer

Specify any defined printer on TIP/ix, or AUX1. The default is PRNTR.

hdr

Specify if a header (separator) page is required. Default is "N" if the printer is an AUX printer, otherwise, default is "Y".

case

Is uppercase translation required? The default is no translation ("L").

plen

Specify page length for the printer. The default length is 66 lines per page.

Main Menu Screen

When **fileview** starts, the screen contains a list of files and subdirectories.

```

uw7test.tws - TIP WorkStation
Session Edit View Tools Help
TF$FV01A FILEVIEW - File Viewer Ver. 1999/03/31 2.3 RO - 0000
directory: /home/eduardov
prefix: * printer: PRNTR header: Y
case: L page length: 66
Size in bytes
Cmd Type File
D ..
F .sh_history 1514
F .profile 4482
D mfq
F .scoadmin.pref 111
D .dt
F .dtprofile 6065
F .mailcap 694
F .Xauthority 101
F .mime.types 1009
Change directory Change prefix Change printer:
<-- Enter 'D' to delete, 'S' to view file or to open directory,
'P' to print file.
F2=Next page F3=Previous page F5 or MSG WAIT=Exit F7=Help F10=Delete
7,15 24x80 Ready P256 MSG OVR CAP NUM SCRL
    
```

Where:

directory

the current working directory. If you change the directory field, the new directory becomes effective immediately.

prefix

the TIP/30-style prefix. * by default. If you change the prefix field, the new prefix becomes effective immediately.

printer

the current print destination. If you change the printer field, the new print destination becomes effective immediately.

cmd The action to take, as follows:

- D** Delete the selected file.
- S** View the selected file or open the selected directory.
- P** Print the selected file.
- X** Print the file on AUX1.

Type D for directory, or F for file.

The following functions are supported:

Key	Action Taken
F1	Refresh display.
F2	Display next page of list.
F3	Display previous page of list.
F5	Exit current step.
F6	Select parent directory.
F7	Display help screen.
F10	Delete the selected file.
MSG WAIT	Exit current step.

Change printer If page length is changed, it becomes effective immediately, which affects the viewing of files.

View Screen

This screen displays the content of files.

```

Uw7test - TIP WorkStation
Session Edit View Tools Help
TF$FV02A File Viewer: .sh_history Page: 1
+1-----10-----20-----30-----40-----50-----60-----70--+
1 : ls
2 :
3 :
4 :
5 :
6 :
7 :
8 :
9 :
10 :
11 :
12 :
13 :
14 :
15 :
16 :
17 :
+1-----10-----20-----30-----40-----50-----60-----70--+
Enter Cmd: Line: Page: Column:
Text:
EN=End Viewing FI=Find string GO=Go to '+'=Forward '>'=Shift right
F4=Next screen F5=Previous screen MSG WAIT=Exit F7=Help F10=Delete
21,13 25x80 Ready P599 MSG OVR CAP NUM SCRL
    
```

The default value for Line, Page, and Column is 1.

Where:

Cmd

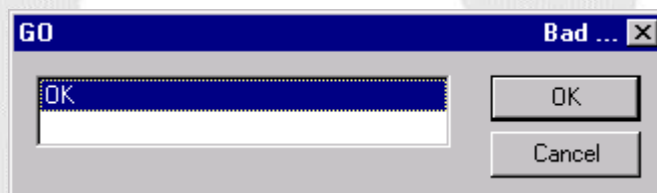
can be one of the following:

- BM** Add a bookmark. A bookmark is a user-defined location in a file. The bookmark persists until you close the file (or explicitly delete the bookmark).
- DB** Delete a bookmark.
- DE** Delete the file being viewed.
- ED** Invoke FSE editor.
- EN** End Viewing. Current file is closed. Next opened file, if available, becomes the new current file.
- FI** Find next occurrence of a string. Display string in highlighted line at top of screen.
- FM** Same as FI, but display string in middle of screen.
- F-** Same as FI, except search in backward direction.
- Go** Go to specified line, page, bookmark, or column. If bookmark, the bookmark is specified in the text field.
- He** Display HELP information
- L** Display lines from specified line and page.
- LB** List bookmarks. If selected, go to specified bookmark.
- LL** List last screen of last file page.
- N** Switch to next file.

- NX** Select and go to next file from opened file list. This is the interface to viewing multiple files.
- O** Current file is kept opened. Next file is opened if specified, or new files can be specified and opened.
- Pr** Same as L command.
- PE** Peek at a specified line.
- P+** Peek at line after last peeked at line.
- P-** Peek at line before last peeked at line.
- Q** Quit. The same as command O.
- TS** Change the Tab Size. When displaying a file on screen, tabs are converted to spaces. (The file itself is not modified.) The default tab size is 4.
- VI** Invoke vi editor.
- +** Forward specified number of lines.
- Backward specified number of lines.
- <** Shift line data to the left.
- >** Shift line data to the right.
- F1** Refresh display.
- F2** Next file page.
- F3** Previous file page.
- F4** Display next screen.
- F5** Display previous screen.
- F7** Display help screen.
- F9** Repeat last find (F1, FM, or F-).
- F10** Same as command DE.
- MSG WAIT**
End viewing. Current file is closed.

List of bookmarks:

When the command Go is entered and it is go to a bookmark, the following popup window (TIP/ws) appears when the specified bookmark is not found:



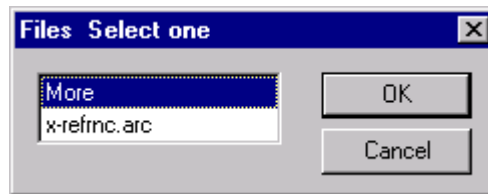
The user selects one entry from this list for command Go. If the entry for more is selected and there are more bookmarks, the next screen of bookmarks will be displayed for selection.

Where:

- MSG WAIT**
Cancel Go command.

List of opened files:

When the command NX is entered, the following popup window appears:



If the entry for more is selected, more opened files will be displayed.

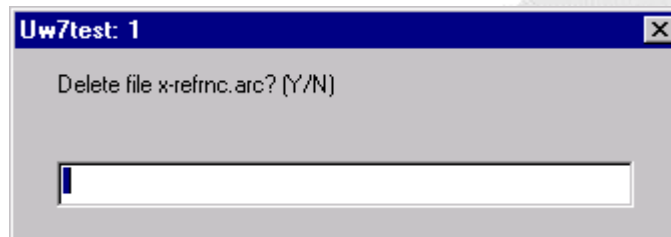
Where:

MSG WAIT

Cancel the operation.

Screen of Deletion:

If delete is selected in TIP/ws, you are given the option (Y/N) if Y is selected, then the file is erased from UNIX spooler. Otherwise, the operation is cancelled.



Where:

MSG WAIT

Cancel the operation of deleting file.

fin - Exit TIP/ix shell

Use this command to exit the tipix end-user interface (shell).

Syntax:

fin

There are no parameters for this command.

fixperms - Adjust Permissions

Configure your system's permissions. This script was originally provided to upgrade permissions from TIP/ix 1.7 to 2.0.

Syntax

```
fixperms [-0|-1] [ -bcdinsv ]
```

Where:

- 0** Minimal security
- 1** Maximum security
- b** Set permissions for batch programs
- c** Set permissions for all catalogued files
- d** Set permissions for dynamic files
- i** Interactive mode
- n** Show but do not execute the operations.
- s** Set permissions for system files
- v** Verbose output

fixperms - Minimal Security

If security is not an important consideration for your TIP/ix system, you may choose to configure your system with generally open permissions. This will permit batch programs to access TIP/ix data files with no further changes to your system.

To set the ownership and permissions for the existing data files in your TIP/ix system, run the following command as *root*:

```
fixperms -0scd
```

This changes the ownership of all TIP/ix system files, all catalogued files, and all dynamic files and edit buffers to the user *tipixusr* that you defined during the installation, and sets the permissions of those files so that batch programs may still access them.

fixperms - Secure Systems

To ensure that your data files cannot be read or modified other than through authorized interfaces, you may wish to configure your system with secure permissions. The cost of this added security is that it will require any batch programs that access TIP/ix data files to have authorization.

To set the ownership and permissions for the existing data files in your TIP/ix system, run the following command as *root*:

fixperms -lscd

This changes the ownership of all TIP/ix system files, all catalogued files, and all dynamic files and edit buffers to the user tipixusr that you defined during the installation, and sets the permissions of those files so that only authorized programs and users may access them.

Once this is done, only programs, which have the permissions of the TIP/ix administrative user id, will be permitted to access TIP/ix files. This means that you must either log in as the TIP/ix administrative user to run all batch programs, or else you must make all batch programs owned by the TIP/ix administrative user and turn on their setuid bit. The fixperms program can be used to so configure your batch programs

If "mybatch" is a batch program, which accesses TIP/ix files, you can type the command

```
fixperms -lb mybatch
```

and mybatch will be made "setuid tipixusr." Likewise, if the directory "/tipix/batch/bin" contains all of your batch program executables (and nothing else), you may type

```
fixperms -lb /tipix/batch/bin/*
```

to configure all such programs at once.

Additional considerations:

If you currently use the LD_LIBRARY_PATH environment variable when you link your batch programs, you will need to instead use the -L option with the linker. This is because UNIX does not honor the LD_LIBRARY_PATH environment variable for setuid programs.

fopen - Open Online Files

The **fopen** utility transaction opens one or more files and makes those files available for online program use. The files to be opened may have previously been closed by the [FCLOSE](#) transaction (see the [FCLOSE](#) utility documentation in this manual.) This program does NOT operate interactively.

You may specify up to eight filenames on the command line. Each filename may be specified as a unique name or may use prefix notation.

Some fopen options change the access (read-only or update) and file share (shared or exclusive) attributes of the specified files. The access or file share changes, if successful, are immediate and lasting. You can verify the changes with [smfile](#).

Syntax 1: - online (from the TIP/ix prompt)

```
fopen[, [q]init] file 1 ... [,file 8]
fopen[, [q]srd] file 1 ... [,file 8]
fopen[, [q]sup] file 1 ... [,file 8]
fopen[, [q]exrd] file 1 ... [,file 8]
fopen[, [q]exup] file 1 ... [,file 8]
```

Where:

- ,q** This option (quiet) instructs **fopen** to suppress the usual informational messages that are displayed. If this option is combined with another option, it must precede that option. For example: “fopen,qinit tstfile”
- ,init** This option instructs **fopen** to open and initialize the file. If TIP/ix is not available, or if the file is already open, this option is ignored.

If the file is *successfully opened* with this option, it contains *no records*.
- ,srd** Set the file attributes to *shared, read-only*, then open the file to TIP/ix accordingly.

If the file is already open, this option is ignored.
- ,sup** Set the file attributes to *shared, update*, then open the file to TIP/ix accordingly.

If the file is already open, this option is ignored.
- ,exrd** Set the file attributes to *exclusive, read-only*, then open the file to TIP/ix accordingly.

If the file is already open, this option is ignored.
- ,exup** Set the file attributes to *exclusive, update*, and then open the file to TIP/ix accordingly.

If the file is already open, this option is ignored.

Syntax 2: - batch (from the UNIX prompt)

```
fopen [-q] [-i | -s | -t | -e | -f] file 1
... [file 8]
```

Where:

- q** This option (quiet) instructs **fopen** to suppress the usual informational messages that are displayed.
- i** Open and initialize the file. If TIP/ix is not available, or if the file is already open, this option is ignored.

If the file is *successfully opened* with this option, it contains *no records*.

- s** Set the file attributes to *shared, read-only*, then open the file to TIP/ix accordingly.

If the file is already open, this option is ignored.

- t** Set the file attributes to *shared, update*, then open the file to TIP/ix accordingly.

If the file is already open, this option is ignored.

- e** Set the file attributes to *exclusive, read-only*, then open the file to TIP/ix accordingly.

If the file is already open, this option is ignored.

- f** Set the file attributes to *exclusive, update*, and then open the file to TIP/ix accordingly.

If the file is already open, this option is ignored.

file 1... [file 8]

The names of the files to be opened. At least one name must be specified. The file names represent file definitions (see [smfile](#)) and not TIPFCS file names (created and maintained by [smsec](#)).

Examples:

- Open all the files that start with “CUST” and the two specified files (INVMAS^T and ORDENT). From TIP/ix:

```
fopen *CUST, INVMAST, ORDENT
```

From UNIX:

```
fopen \*CUST INVMAST ORDENT
```

“*CUST” is TIP prefix notation (not a UNIX shell regular expression).

- This opens the file associated with the file definition TSTFILE (see [smfile](#)) with the options “q” and “init”. From TIP/ix:

```
fopen, qinit TSTFILE
```

From UNIX:

```
fopen -q -i TSTFILE
```

The “q” option suppresses informational messages and the “init” option initializes the file (so that the file is empty). If the file is already open then no action occurs and the “init” option is ignored.

Return Codes

If you are using `fopen` from a UNIX shell script, `open` returns one of the following codes:

Code	Meaning
0	Successful.
2	Item does not exist. The file name specified may not be defined to TIP/ix (there is no entry for that name entered via the <code>smfile</code> utility).
4	No response from TIPFCS (and TIP/ix is running).

Your Korn or Bourne scripts can use the shell parameter `$?` to get the return code of the last program executed. See **`fclose`** for an example.

Additional Considerations:

- When running the `fopen` program in batch, the `TIPROOT` environment variable indicates which TIP/ix system is being referenced.
- If you request file initialization, and the file does not have the update attribute, the file is opened but not initialized.
- Restarting TIP/ix does not cause files that have been marked as offline (with `fclose`) to be brought online. This is to prevent files from coming back online due to an unscheduled shutdown and startup of TIP/ix. Once a file is closed with `fclose`, it remains offline until a subsequent `fopen`.

This is different from TIP/30 where all files are marked as online when TIP/30 starts up. If you want to bring all TIP/ix files back online when TIP/ix starts up (TIP/30 style), do this:

```
fopen \*
tipctl b
```

- You can run `fopen` even when TIP/ix is not running. The next time TIP/ix starts up the file will be marked online (available). The return code will be either 0 or 2.

If you encounter the error message “error opening file. Check UNIX permissions” this may be due to the size of the file trying to be opened is zero **k**, or another file in the same FCS server may be of size zero **k**.

free - Release Active Files

The **`free`** program is used to release files from the active file table (AFT). Files are placed in the active file table by TIP/ix when a transaction program accesses the file. If a transaction program aborts, it is possible that the files that the transaction was using may remain in the active file table.

When a file is removed from the AFT, any associated record locks currently maintained for that file are also released.

Regardless of the syntax used, the **free** program also unconditionally flushes the user's screen format pool

Syntax:

```
free file 1 [,file 2] ... [,file 8]
free,a
free,x
free,f file 1 [,file 2] ... [,file 8]
```

Where:

- free** If the transaction free is invoked without command line options, up to 8 filenames can be specified to be removed from the active file table. (The filename * is a popular value with the same meaning as free,a.)
- free,a** If the transaction free is invoked with command line option "a", all files currently in the AFT are released (no filenames are required or expected on the command line!)
- free,x** If the transaction free is invoked with command line option "x", all record locks for all files in the AFT are relinquished, but the files remain in the AFT.
- free,f** If the transaction free is invoked with command line option "f", all record locks for all files specified on the command line are relinquished, but the files remain in the AFT.

freset - Close and Re-open Online Files

The **freset** utility transaction closes and re-opens one or more files. This is the same as running [fclose](#) immediately followed by [fopen](#) on the same file(s). If the close does not complete (because the file is assigned to some user(s)) then the open will report that file is already open and the file will not be initialized if the "init" option was specified. This program does *not* operate interactively.

You may specify up to eight filenames on the command line. Each filename may be specified as a unique name or may use prefix notation.

Some **freset** options change the access (read-only or update) and file share (shared or exclusive) attributes of the specified files. The access or file share changes, if successful, are immediate and lasting. You can verify the changes with [smfile](#).

Syntax 1: - online

```
freset[,[q]init] file 1 [,file 2] ... [,file 8]
```

```
freset[,[q]srd] file 1 [,file 2] ... [,file 8]
freset[,[q]sup] file 1 [,file 2] ... [,file 8]
freset[,[q]exrd] file 1 [,file 2] ... [,file 8]
freset[,[q]exup] file 1 [,file 2] ... [,file 8]
```

Where:

,q This option (quiet) instructs **freset** to suppress the usual informational messages that are displayed. If this option is combined with another option, it must precede that option. For example: “freset,qinit tstfile”

,init This option instructs **freset** to initialize the file when it is re-opened.

The initialization will not take place if the close does not complete. This would happen if the file were currently assigned to some users.

If TIP/ix is not available this option is ignored, otherwise if the file is successfully opened with this option, it contains no records.

,srd Close the files, set the file attributes to *shared, read-only*, then open the files to TIP/ix.

The file access will only be altered if the close completes successfully (the file is not assigned to any users).

,sup Close the files, set the file attributes to *shared, update*, then open the files to TIP/ix.

The file access will only be altered if the close completes successfully (the file is not assigned to any users).

,exrd Close the files, set the file attributes to *exclusive, read-only*, and then open the files to TIP/ix. The file access will only be altered if the close completes successfully (the file is not assigned to any users).

,exup Close the files, set the file attributes to *exclusive, update*, then re-open the files to TIP/ix.

The file access will only be altered if the close completes successfully (the file is not assigned to any users).

Syntax 2: - batch

```
freset [-q] [-i | -s | -t | -e | -f] file 1 ...[,file 8]
```

Where:

-q This option (quiet) instructs **freset** to suppress the usual informational messages that are displayed.

-s Close the files, set the file attributes to *shared, read-only*, and then re-open the files to TIP/ix. The file access will

only be altered if the close completes successfully (the file is not assigned to any users).

- t** Close the files, set the file attributes to *shared*, *update*, and then re-open the files to TIP/ix.

The file access will only be altered if the close completes successfully (the file is not assigned to any users).

- e** Close the files, set the file attributes to *exclusive*, *read-only*, and then re-open the files to TIP/ix. The file access will only be altered if the close completes successfully (the file is not assigned to any users).

- f** Close the files, set the file attributes to *exclusive*, *update*, and then re-open the files to TIP/ix. The file access will only be altered if the close completes successfully (the file is not assigned to any users).

file 1 ... [,file 8]

The names of the files to be opened. At least one name must be specified. The file names represent file definitions (see [smfile](#)) and not TIPFCS file names (created and maintained by [smsec](#)).

Examples:

- The following closes and re-opens all the files that start with "CUST" and the two specified files (INVMAS and ORDENT). Note that "*CUST" is TIP prefix notation (not a UNIX shell regular expression).

```
freset *CUST INVMAS ORDENT
```

- The following closes and then re-opens the file associated with the file definition TSTFILE (see [smfile](#)) with the options "q" and "init". The "q" option suppresses informational messages and the "init" option initializes the file when it is re-opened (so that the file is empty). The file cannot be closed (because it is assigned to some user(s)) then the file will not be initialized.

```
freset,qinit TSTFILE
```

- This is the batch version of the previous example.

```
freset -q -i TSTFILE
```

Error Conditions:

The file name specified may not be defined to TIP/ix (there is no entry for that name entered via the [smfile](#) utility).

Additional Considerations:

When running the **freset** program in batch, the TIPROOT environment variable indicates which TIP/ix system is being referenced.

FSE - Full Screen Editor

The Full Screen Editor (**fse**) is a screen format oriented editor that is designed to be both powerful and easy to use. It uses TIP/ix services so it can only be run from within the TIP/ix shell. This means that **fse** is not available from the UNIX shell.

fse operates by displaying a full screen of text (17 lines) using a TIP/ix screen format. The screen format provides a command area in addition to the display area.

The user may directly alter the text that is displayed or may enter commands to display, find, move, copy, add, delete or modify text in the workspace.

Searching and substitution commands are provided and can act on specific column ranges.

Scrolling is accomplished by using the "Forward Page" and the "Backward Page" function keys (or commands).

fse performs all of its work in a working copy of the data. This copy is held in a TIP/ix Edit Buffer (often called a *workspace* in this documentation).

Lines in the workspace are displayed with consecutive line numbers that are used as reference points by the various commands.

Syntax:

```
fse [-R] [file] [[group] buffer] [reclen]  
fse [-R] [library] [module] [[group] buffer]  
[reclen]  
fse [-R] [buffer]
```

Where:

-R Option which causes **fse** to operate in Read Only mode. Writes will only be honored if a file name is supplied. This option is assumed if the user does not have write permission to the file. **fse** displays a message with the initial screen indicating Read Only mode is active.

-W[80 | 124]

Open a file in wide display format. There are three cases:

-W or -W124 provides 124 columns

-W80 provides 80

If -W not supplied, the default width is **72**.

file Unix file to edit.

If the first parameter contains a slash or period or is longer than 8 characters, then it will be treated as a Unix file and the first syntax form will apply.

If the file name does not begin with a slash then **fse** will access it in the current working directory.

To force **fse** to use the Unix file syntax when 2 or more parameters are supplied, simply prefix the file name with period slash .

Example: **fse ./abc xyz**

The ./ ahead of abc forces it to be treated as a Unix file name instead of a library. The buffer name will be xyz in the user's first elective group.

library

The logical name of a TIP/ix library as defined by [SMSEC](#). The [SMFILE](#) definition determines what directory will store the library modules. Cannot exceed 8 characters.

module

The name of a module in the TIP/ix library. If it exceeds 8 characters in length, the module is treated as a Unix file, and its full path name is determined from the library name provided.

group

The name of the group that will contain the edit buffer.

The default group name is the user's first elective group as defined using [SMUSER](#).

buffer

The name of the edit buffer to be created or accessed.

When accessing a TIP/ix library module, the module name is used as the default buffer name

When accessing a Unix file, the default buffer name is set to the portion of the Unix file from the last slash up until the last period after the slash.

Example: **fse src/abc.cbl**

This would result in a buffer name of *abc* in the user's first elective group.

reclen

If the last parameter is numeric, it is treated as the record length to use for the edit buffer. The default length is 80. Must be between 72 and 200.

If the file being edited contains lines longer than *reclen* then lines exceeding *reclen* will be split into multiple lines. *fse* will display a message with the initial screen if this situation occurs.

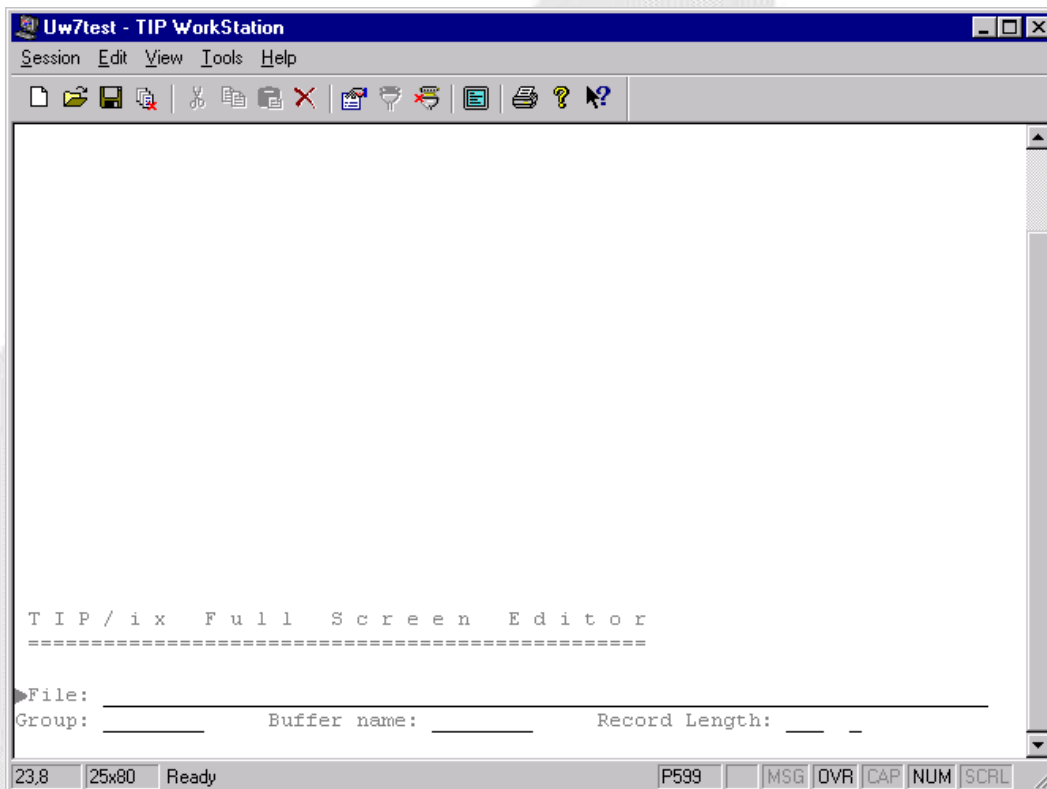
It is not possible to use a slash to delimit a library name from a module name as this will result in *fse* treating the first parameter as a Unix path.

If a library and module are supplied on the command line **fse** attempts to copy the specified module from the library to an edit buffer. The edit session will continue with an empty edit buffer if the copy fails. If the edit buffer already exists the user is prompted to confirm that the copy should be attempted. Respond No to bypass the copy and edit the existing edit buffer

If a file name is supplied on the command line **fse** attempts to copy the specified file to an edit buffer. The edit session will continue with an empty edit buffer if the copy fails. If the edit buffer already exists the user is prompted to confirm that the copy should be attempted. Respond No to bypass the copy and edit the existing edit buffer.

To access an existing edit buffer supply the buffer name as the initial file name and then respond with No to the prompt "Overwrite edit buffer".

If no command line parameters are supplied, **fse** displays the following screen format to allow easy entry of the required information.

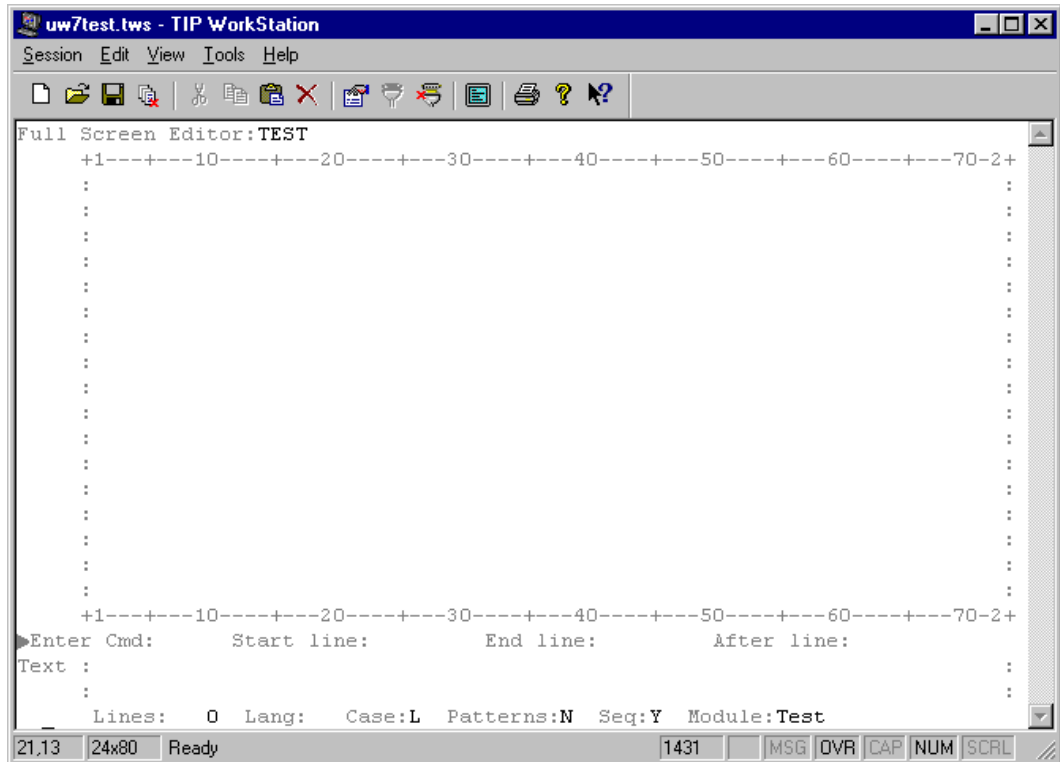


For editing **fse** uses a screen format that serves two purposes

- display the current screen of text (if any)
- provide (at the bottom of the CRT) a command and status area.

The screen format displayed varies slightly depending on the type of text in the edit buffer; for simplicity, this documentation assumes that the text has been declared to be COBOL source code.

The screen format appears as follows :



The screen format is actually two distinct areas. The upper area (lines 1 through 20) is used to display the current screen of text from the workspace. Line 1 of the screen format contains the current name of the edit buffer and is used for error or informational messages.

You may also use the upper area for direct modification of the text that is displayed in that area. The lower area (lines 21 through 24) is used to enter commands to **fse**. The last line of the screen format contains protected fields that display current status information about the edit buffer:

Lines

This field always indicates the total number of lines in the edit buffer.

Lang

This field indicates the declared language type of the data in the edit buffer (see description of the SE command in SE Set Options for a valid language codes).

Case

This field indicates the current input case.

- U** input (alphabetic) translated to upper case.
- L** input taken literally (no translation).

Patterns

A "Y" or "N" indicating whether or not pattern mode is in effect.

In pattern mode, fse interprets search strings in a non-literal manner.

- Seq** A "Y" or "N" indicating whether or not fse is to automatically sequence (according to the declared language) on a write command.

Module

The file that is currently associated with the workspace.

There are *two* cursor resting positions (the first is near the start of line 20; the second is near the start of line 24). Take care where you place the cursor before you press the **XMIT** key.

In this documentation, the command fields are referenced by the following names:

{startline}

Field named "Start line:" (row 21)

{endline}

Field named "End line:" (row 21)

{afterline}

Field named "After line:" (row 21)

{text1}

Field named "Text | ... |" (row 22)

{text2}

Unnamed field "| ... |" (row 23)

Line Numbers

Line numbers are normally specified to the Full Screen Editor as a positive *whole number* in the (inclusive) range of 1 to the current maximum line number in the workspace. The current maximum line number is always displayed in the lower left corner of the screen.

One exception to this rule is the use of line numbers -9 through -1 (inclusive). A negative line number implies using the line number that was

previously "stored" in the line number register 1 through 9 (see description of the **fse** "#d" command).

Line 9999

Another exception is the use of line number 9999. That particular value is interpreted as "the last line in the buffer".

Whenever **fse** inserts or deletes lines in the workspace, the entire workspace is renumbered. The workspace lines are always whole numbers - fractional line numbers are not used or recognized by **fse**.

fse recognizes line numbers that are a maximum of four digits (the fields on the screen are defined with a sign - this explains why 5 screen positions exist for each field).

Column Ranges

A column range is normally specified to the Full Screen Editor in one of two formats:

- n A specific column number as a numeric value.
- m:n A range of columns from column "m" to "n" inclusive.

The nature of each particular **fse** command dictates whether a single column number or a column range is appropriate for the command.

Column numbers in the first format (a single column) are accepted (where appropriate for some commands) in {startline}, {endline}, {afterline}, {text1}, or {text2}.

Column numbers of the second format (a range) are only accepted in the fields {text1} and {text2}.

Strings

When the Full Screen Editor command syntax requires the specification of a text string (for example the "find string" command) the string is normally entered in the {text1} or {text2} field provided in the lower area of the screen format.

The maximum string length that is recognized by **fse** is 72 characters.

If the string to be entered includes significant trailing spaces, it *must* be entered within single or double quotes (either type of quote character is acceptable provided the same one is used at both ends of the string). Leading spaces in a string are always significant.

If the first non-blank character of the string is a digit (0 through 9 inclusive), the string must be entered within quotes (otherwise the digit or

digits are assumed to represent a column number or column number range!).

If the string contains a single quote or double quote character, it is suggested that the **other** quote character be used to delimit the string itself.

When a string is specified as part of a search operation ([FInd](#), [DElete](#), [Substitute](#)) the string may be prefixed by an exclamation mark to indicate that a line is desired that does *not* contain the string (for example: FI !ABC means find the next line not containing the string "ABC").

When the user has set on "pattern matching" mode, some characters in strings take on special meaning see [Pattern Matching](#) for the documentation of effects of pattern matching mode.

Command Summary

The following table is a summary of the commands that **fse** recognizes. **fse** commands are limited to one or two characters. Upper case letters in the command syntax are required; lower case characters are optional.

Command	Description
AB	Add lines before.
Ad	Add lines.
BX	Create comment box.
CB	Copy lines before target line.
CC	Copy column range to another column.
CO	Copy lines.
DE	Delete lines.
DU	Duplicate a set of lines.
En	End editing (retain work space).
EX	Execute TIP/ix command line(s).
FA	Find all lines containing a string.
FI	Find next occurrence of a string.
FM	Same as FI, except place found line in middle of screen.
F#	Equate a command to a user function key (F10-F22).
F-	Same as FI, except search in backward direction.
He	Display HELP Information
In	Insert empty lines into screen (after a line).

IB	Insert empty lines into screen (before a line).
KN	Keep ON
KF	Keep OFF
Li	List (display) lines on screen.
LL	List (display) last screen of text of edit buffer.
MA	Establish left and right margins.
MB	Move lines before target line.
MC	Move constant or column range to a column range.
MF	Turn status marker off. When writing the module fse will not write a status line prior to the module text.
MN	Turn status marker on. If the module has a language code of C (COBOL) or R (RPG) then when writing the module fse will write a status line as a comment line prior to the module text.
MO	Move lines.
O	Set language to "space".
OA	Set language "A".
OC	Set language "C".
OD	Set language "D".
OL	Option Literal case (input text unaltered).
OP	Set language "P".
OR	Set language "R".
OT	Set language "T".
OU	Option Upper case (input text mapped to upper case).
OX	Set language to "X".
Pr	Print (display) lines - same as L command.
PE	Peek at a specified line.
P+	Peek at line after last peeked at line.
P-	Peek at line before last peeked at line.
Qu	Quit editing (discard the work space).
Re	Read lines from a file.
RC	Recall (redisplay) last command entered from keyboard.
Su	Substitute old string with new string.

SA	Sort ascending.
SD	Sort descending.
SE	Set editor defaults.
SH	Execute a UNIX shell.
SP	Substitute old string with new string; show changes.
SW	Switch (interchange) two lines.
UX	Execute a UNIX command.
Up	Update a range of lines.
Wr	Write lines to a library module, UNIX file, or edit buffer.
WE	Write lines and then END fse session.
WN	Write lines (NO OVERWRITE PROMPT!).
WQ	Write lines and then QUIT fse session.
WX	Write Without Overwrite Check and END fse session.
WZ	Write Without Overwrite Check and QUIT fse session.
+	Forward n lines.
-	Backward n lines.
=	Same as SE command.
<	Shift line data to the left.
>	Shift line data to the right.
^	Call fse recursively.
%	Call fse out of a current edit session and do not return to fse.
#d	Save line number in fse register number (1-9).
ld	Clear fse register number (1-9).
XMIT	In Command area (with {Cmd} field empty) - display next screen. (Forward Page) In Display area - update lines from display area
MSG WAIT	END editing (save work space if not empty).
F1	Refresh screen display.
F2	Display next "page" (Forward Page). Default page is 16 lines.
F3	Display previous "page" (Backward Page). Default

page is 16 lines.

F4	QUIT editor; discard work..
F5	Insert blank line ahead of line that cursor is on. - or restore line just deleted with F6
F6	Delete line that cursor is on.
F7	"Split" line at cursor location.
F8	"Join" line to following line at cursor location.
F9	Reissue last Find command.
F10...F22	Available for user definition (via F# command).

AB - Add Lines Before

The Add command allows the user to add new lines of text *before* a specific line number. The user should enter "AB" as the command and may specify a {startline} (an entry in {afterline} is synonymous).

If a {startline} or {afterline} is not specified, **fse** assumes that the lines are to be added at the beginning of the workspace.

If the text to be added is two lines or less the user may enter the data directly in the {text1} and {text2} fields of the screen format and press transmit from the second cursor resting location.

If more than two lines are to be added, the user should leave {text1} and {text2} blank and simply specify the line number that immediately precedes the lines to be added (this line number may be entered in {startline} or {afterline}).

fse re-displays the screen with the contents of the specified line in protected format on the first line of the display, and leaves the remaining 16 lines left unprotected and blank.

The user may then enter any desired text below the (protected) first line and press **XMIT** at the first cursor resting location.

Trailing lines that are entirely blank are not added.

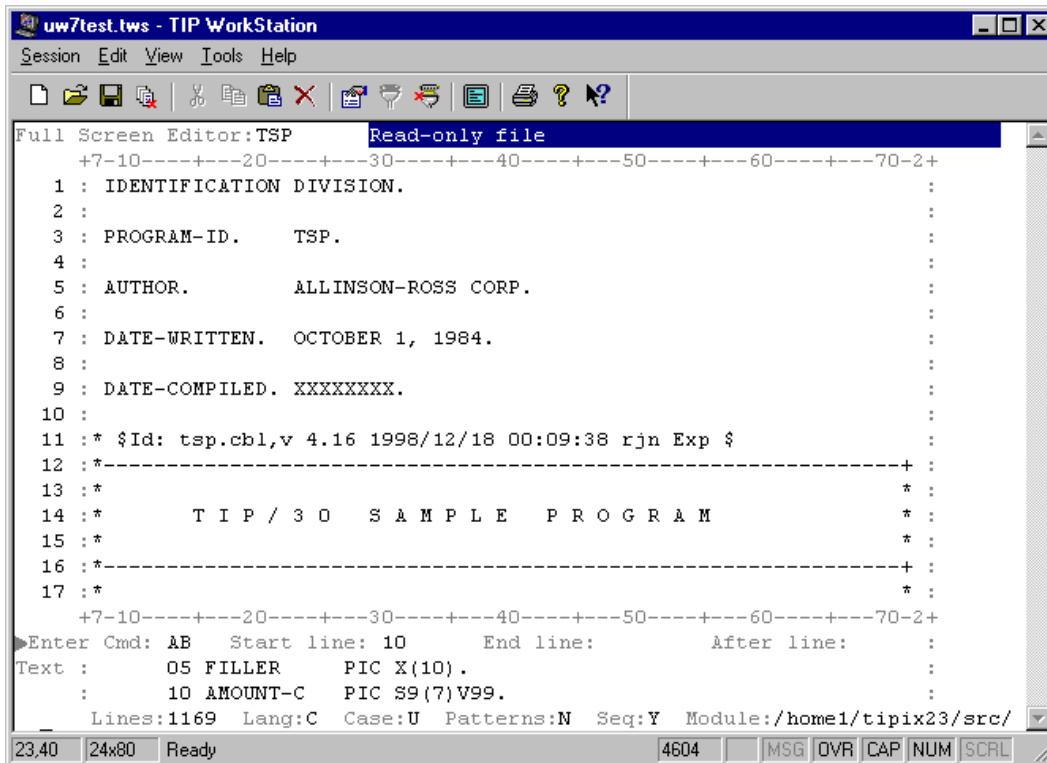
The {endline} is ignored by the ADD command.

If the last line added (line 17) is NOT blank, a fresh screen is displayed to allow entry of more lines.

Limitation: Pressing **MSG WAIT** while in ADD mode exits **fse**.

Pressing **XMIT** without entering any data cancels the ADD command.

Example:



```

Full Screen Editor:TSP      Read-only file
+7-10-----+--20-----+--30-----+--40-----+--50-----+--60-----+--70-2+
 1 : IDENTIFICATION DIVISION.                :
 2 :                                          :
 3 : PROGRAM-ID.      TSP.                    :
 4 :                                          :
 5 : AUTHOR.          ALLINSON-ROSS CORP.     :
 6 :                                          :
 7 : DATE-WRITTEN.    OCTOBER 1, 1984.        :
 8 :                                          :
 9 : DATE-COMPILED.   XXXXXXXXX.             :
10 :                                          :
11 : * $Id: tsp.cb1,v 4.16 1998/12/18 00:09:38 rjn Exp $ :
12 : *-----+-----+-----+-----+-----+-----+ :
13 : *                                          * :
14 : *      T I P / 3 0   S A M P L E   P R O G R A M      * :
15 : *                                          * :
16 : *-----+-----+-----+-----+-----+-----+ :
17 : *                                          * :
+7-10-----+--20-----+--30-----+--40-----+--50-----+--60-----+--70-2+
Enter Cmd: AB      Start line: 10      End line:          After line:      :
Text :      05 FILLER      PIC X(10).      :
      :      10 AMOUNT-C   PIC S9(7)V99.    :
      :                                          :
      : Lines:1169 Lang:C Case:U Patterns:N Seq:Y Module:/home1/tipix23/src/
23.40 24x80 Ready 4604 MSG OVR CAP NUM SCRL

```

This example illustrates adding (only two) lines of text directly before line 10. Since the text to be added was only one or two lines it is more convenient to code them in the text area provided rather than issue a naked AB(10) command and then entering the text in the upper area of the screen format.

Ad - Add Lines

The ADD command allows the user to add new lines of text *after* a specific line number. The user should enter "AD" as the command and may specify a {startline} (an entry in {afterline} is synonymous).

If a {startline} or {afterline} is not specified, **fse** assumes that the lines are to be added at the end of the workspace.

If the text to be added is two lines or less the user may enter the data directly in the {text1} and {text2} fields of the screen format and press transmit from the second cursor resting location.

If more than two lines are to be added, the user should leave {text1} and {text2} blank and simply specify the line number that immediately precedes the lines to be added (this line number may be entered in {startline} or {afterline}).

BX - Create Comment Box

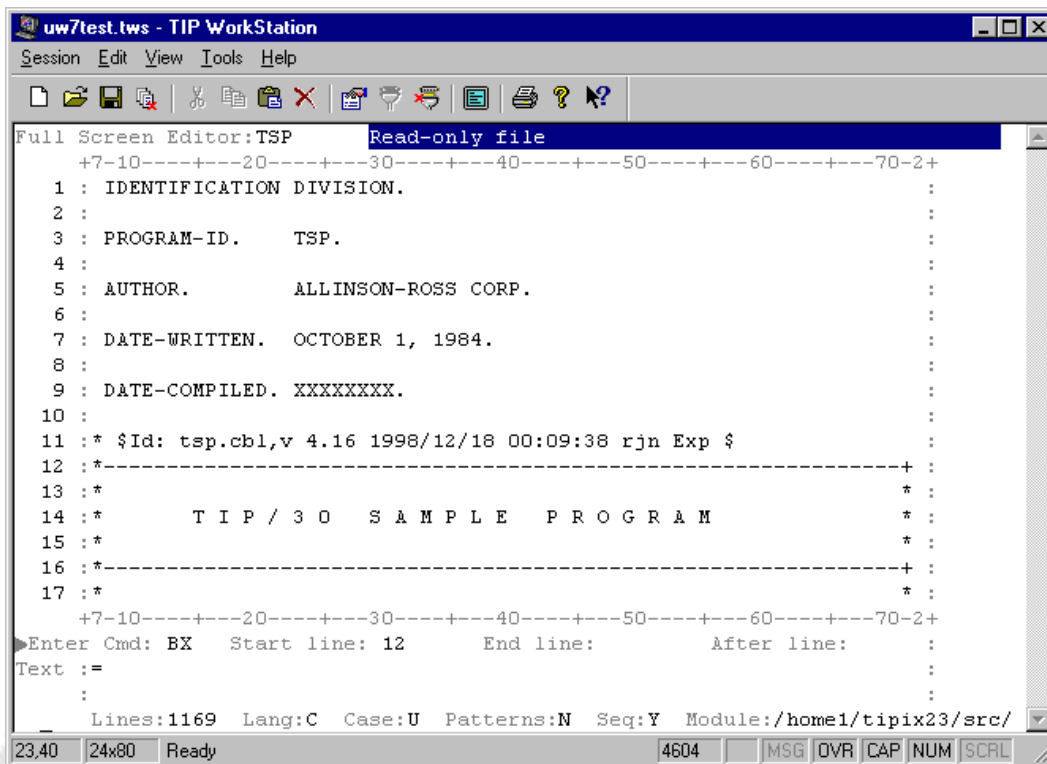
The BX command may be used to generate a comment box. This command is available for language COBOL or ASSEMBLER.

The {startline} must be set to the line immediately preceding the desired start of the box.

The {afterline} is set to the number of interior lines in the box (3 is the default for this value).

The first character of {text1} may be used to override the character used to draw the box ("- " is the default).

Example:



```

uw7test.tws - TIP WorkStation
Session Edit View Tools Help
Full Screen Editor:TSP Read-only file
+7-10-----20-----30-----40-----50-----60-----70-2+
1 : IDENTIFICATION DIVISION.
2 :
3 : PROGRAM-ID. TSP.
4 :
5 : AUTHOR. ALLINSON-ROSS CORP.
6 :
7 : DATE-WRITTEN. OCTOBER 1, 1984.
8 :
9 : DATE-COMPILED. XXXXXXXX.
10 :
11 : * $Id: tsp.cbl,v 4.16 1998/12/18 00:09:38 rjn Exp $
12 : *-----+
13 : *
14 : * T I P / 3 0 S A M P L E P R O G R A M
15 : *
16 : *-----+
17 : *
+7-10-----20-----30-----40-----50-----60-----70-2+
Enter Cmd: BX Start line: 12 End line: After line:
Text :=
:
:
Lines:1169 Lang:C Case:U Patterns:N Seq:Y Module:/home1/tipix23/src/
23.40 24x80 Ready 4604 MSG OVR CAP NUM SCRL

```

Generates:

- Before Box Insert

```

Uw7test - TIP WorkStation
Session Edit View Tools Help
Full Screen Editor:P9551
+7-10-----+---20-----+---30-----+---40-----+---50-----+---60-----+---70-2+
33 : ENVIRONMENT DIVISION.
34 :
35 : CONFIGURATION SECTION.
36 :
37 : DATA DIVISION.
38 :
39 : WORKING-STORAGE SECTION.
40 :
41 : 01 FUNCTION-CODES. COPY TC-FCS.
42 :
43 : 01 UPDATE-MENU-INFO.
44 : 05 UPDATE-MENU-NAME VALUE "TF$TSPO_" PICTURE X(8).
45 :
46 : 01 MENU-INFO.
47 : 05 MENU-NAME VALUE "TF$TSP1_" PICTURE X(8).
48 :
49 : 01 FULL-INFO.
+7-10-----+---20-----+---30-----+---40-----+---50-----+---60-----+---70-2+
Enter Cmd: BX Start line: 33 End line: After line:
Text :=
:
:
Lines:1337 Lang:C Case:U Patterns:N Seq:Y Module:p9551.cb1
22.8 24x80 Ready P900 MSG DVR CAP NUM SCRL
    
```

- After Box Insert (BX Command)

```

Uw7test - TIP WorkStation
Session Edit View Tools Help
Full Screen Editor:P9551
+7-10-----+---20-----+---30-----+---40-----+---50-----+---60-----+---70-2+
34 :*-----*
35 :*
36 :*
37 :*
38 :*-----*
39 :
40 : CONFIGURATION SECTION.
41 :
42 : DATA DIVISION.
43 :
44 : WORKING-STORAGE SECTION.
45 :
46 : 01 FUNCTION-CODES. COPY TC-FCS.
47 :
48 : 01 UPDATE-MENU-INFO.
49 : 05 UPDATE-MENU-NAME VALUE "TF$TSPO_" PICTURE X(8).
50 :
+7-10-----+---20-----+---30-----+---40-----+---50-----+---60-----+---70-2+
Enter Cmd: Start line: End line: After line:
Text :
:
:
Lines:1342 Lang:C Case:U Patterns:N Seq:Y Module:p9551.cb1
4.8 24x80 Ready P900 MSG DVR CAP NUM SCRL
    
```

After creating the comment box, **fse** re-displays the upper screen starting with the first line of the box and positions the cursor in the first blank on line 2 of the box (to facilitate entering the comments).

CB - Copy Lines Before

The CB command allows the user to copy a range of lines from one part of the edit workspace to a point that is "before" another line.

Enter "CB" as the command and provide the starting line and ending line to be copied. Specify in {afterline} the line number of the line which, is to follow the copied lines.

For example, to copy lines 10 through 80 before line 1, specify the command as "CB", the {startline} as "10", the {endline} as "80" and the before line (in the field called {afterline}) as "1". **fse** copies the specified lines before line 1. The lines originally at lines 10 through 80 remain unchanged.

If a string is entered in {text1}, then only lines containing that string are copied. If a column range precedes the string then only lines that contain the string in that column range are copied.

Of course, such a qualification string may be preceded by an exclamation mark ("!") to indicate that the absence of the string is the desired qualification.

For example, to copy lines 10 through 80 before line 1 only if they contain the string "05" in columns 12 through 24, enter "CB" as the command, 10 as the {startline}, 80 as the {endline}, 1 as the {afterline}, and 12:24'05' in {text1} as in the following example:

Example:

```

Uw7test - TIP WorkStation
Session Edit View Tools Help
Full Screen Editor:P9551
+7-10-----20-----30-----40-----50-----60-----70-2+
 2 :
 3 : PROGRAM-ID.    TSP.
 4 :
 5 : AUTHOR.       ALLINSON-ROSS CORP.
 6 :
 7 : DATE-WRITTEN.  OCTOBER 1, 1984.
 8 :
 9 : DATE-COMPILED. XXXXXXXX.
10 :
11 : * $Id: tsp.cbl,v 4.16 1998/12/18 00:09:38 rjn Exp $
12 : *-----+
13 : *
14 : *      T I P / 3 0   S A M P L E   P R O G R A M
15 : *
16 : *-----+
17 : *
18 : *      THIS IS A SAMPLE PROGRAM TO ILLUSTRATE HOW A
+7-10-----20-----30-----40-----50-----60-----70-2+
Enter Cmd: CB  Start line: 10      End line:      After line: 1
Text :12:24'05'
:
:
Lines:1335 Lang:C Case:U Patterns:N Seq:Y Module:p9551.cbl
22,16 24x80 Ready P900 MSG OVR CAP NUM SCRL
    
```

CC - Copy Column Range

The CC command allows the user to copy columns of existing text to a target column within a range of lines. The {startline} defines the first line of the range and the {endline} defines the last line of the range.

The {text1} line must contain the column range to be copied. This value must be entered as a column **range** (that is: 1:5 or 9, etc).

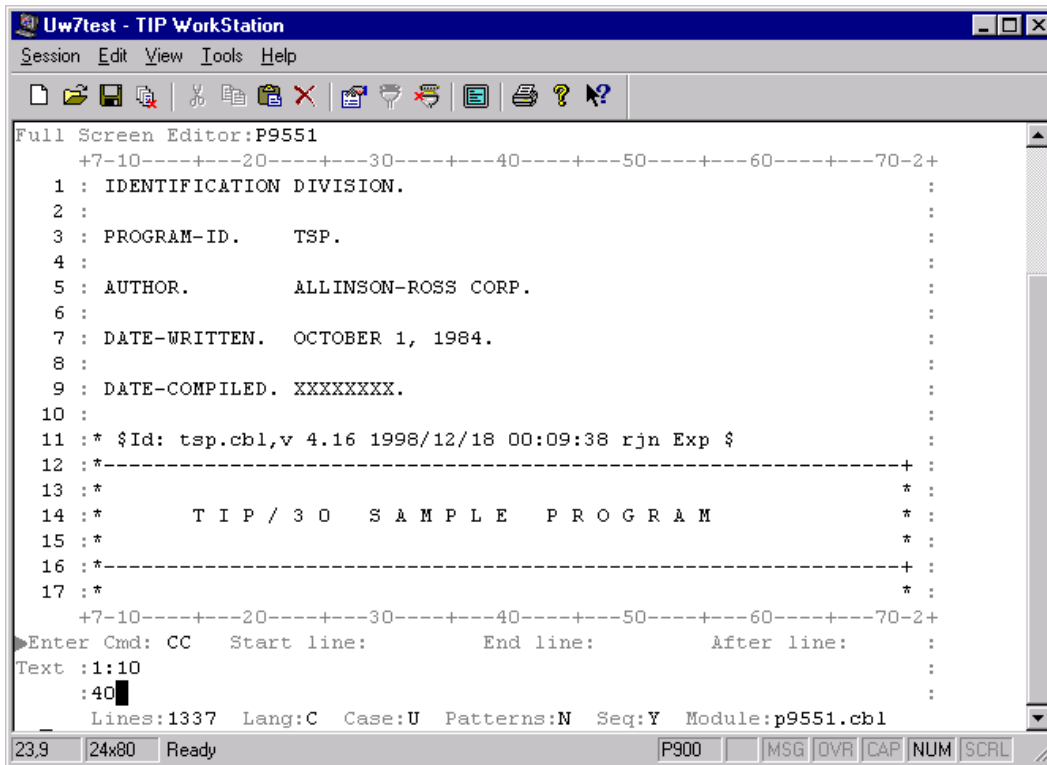
The {text2} line must contain a single column number indicating the target column.

The columns of text specified by the first specification are copied to the column specified (pushing existing text to the right as the copied text is inserted).

The default line range of this command is the 17 lines that are currently displayed in the text area of the screen.

Text is not pushed past the right margin (extra text "falls off" the right edge of a line and disappears into the bit bucket).

Example:



```

Uw7test - TIP WorkStation
Session Edit View Tools Help
Full Screen Editor:P9551
+7-10-----+---20-----+---30-----+---40-----+---50-----+---60-----+---70-2+
1 : IDENTIFICATION DIVISION.                :
2 :                                           :
3 : PROGRAM-ID.      TSP.                    :
4 :                                           :
5 : AUTHOR.          ALLINSON-ROSS CORP.     :
6 :                                           :
7 : DATE-WRITTEN.    OCTOBER 1, 1984.        :
8 :                                           :
9 : DATE-COMPILED.  XXXXXXXXX.              :
10 :                                           :
11 :* $Id: tsp.cb1,v 4.16 1998/12/18 00:09:38 rjn Exp $ :
12 :-----+-----+-----+-----+-----+-----+-----+ :
13 :*                                           * :
14 :*          T I P / 3 0   S A M P L E   P R O G R A M          * :
15 :*                                           * :
16 :-----+-----+-----+-----+-----+-----+-----+ :
17 :*                                           * :
+7-10-----+---20-----+---30-----+---40-----+---50-----+---60-----+---70-2+
Enter Cmd: CC   Start line:           End line:           After line:           :
Text : 1:10                                         :
      :40                                         :
      Lines:1337 Lang:C Case:U Patterns:N Seq:Y Module:p9551.cb1
23.9 24x80 Ready P900 MSG DVR CAP NUM SCRL

```

This example illustrates copying columns 1 through 10 to column 40 (effective on the lines that are currently displayed on the screen).

CO - Copy Lines After

The copy command allows the user to copy a range of lines from one part of the edit workspace to a point that is "after" another line.

Enter "CO" as the command and provide the starting line and ending line to be copied as well as the number of the line that is just ahead of the desired location of the copied text. The last line in the edit buffer is the default value of {afterline}.

For example, to copy lines 1 through 8 after line 17, the user would specify the command as "CO", the {startline} as "1", the {endline} as "8" and the {afterline} as "17". **Fse** copies the lines after line 17 and ahead of the line, that **was** line 18. The lines originally at lines 1 through 8 remain unchanged.

If a string is entered in {text1}, only lines containing that string are copied. If a column range precedes the string then only lines that contain the string in that column range are copied.

Of course, such a qualification string may be preceded by an exclamation mark ("!") to indicate that the absence of the string is the desired qualification.

For example, to copy lines 1 through 8 after line 17 only if they contain the string "05" in columns 12 through 24, enter "CO" as the command, 1 as the {startline}, 8 as the {endline}, 17 as the {afterline}, and 12:24'05' in {text1} as in the following example:

Example:

```

Uw7test - TIP WorkStation
Session Edit View Tools Help
Full Screen Editor:P9551
+7-10-----+---20-----+---30-----+---40-----+---50-----+---60-----+---70-2+
 1 : IDENTIFICATION DIVISION.                                     :
 2 :                                                             :
 3 : PROGRAM-ID.        TSP.                                     :
 4 :                                                             :
 5 : AUTHOR.            ALLINSON-ROSS CORP.                     :
 6 :                                                             :
 7 : DATE-WRITTEN.      OCTOBER 1, 1984.                         :
 8 :                                                             :
 9 : DATE-COMPILED.    XXXXXXXX.                                 :
10 :                                                             :
11 : * $Id: tsp.cb1,v 4.16 1998/12/18 00:09:38 rjn Exp $         :
12 : *-----+-----+-----+-----+-----+-----+-----+ :
13 : *                                                             * :
14 : *          T I P / 3 0   S A M P L E   P R O G R A M       * :
15 : *                                                             * :
16 : *-----+-----+-----+-----+-----+-----+-----+ :
17 : *                                                             * :
+7-10-----+---20-----+---30-----+---40-----+---50-----+---60-----+---70-2+
Enter Cmd: CO   Start line: 1           End line: 8           After line: 17   :
Text : 12:24'05' :
:
Lines:1337 Lang:C Case:U Patterns:N Seq:Y Module:p9551.cb1
22,16 24x80 Ready P900 MSG OVR CAP NUM SCRL
    
```

DE - Delete Lines

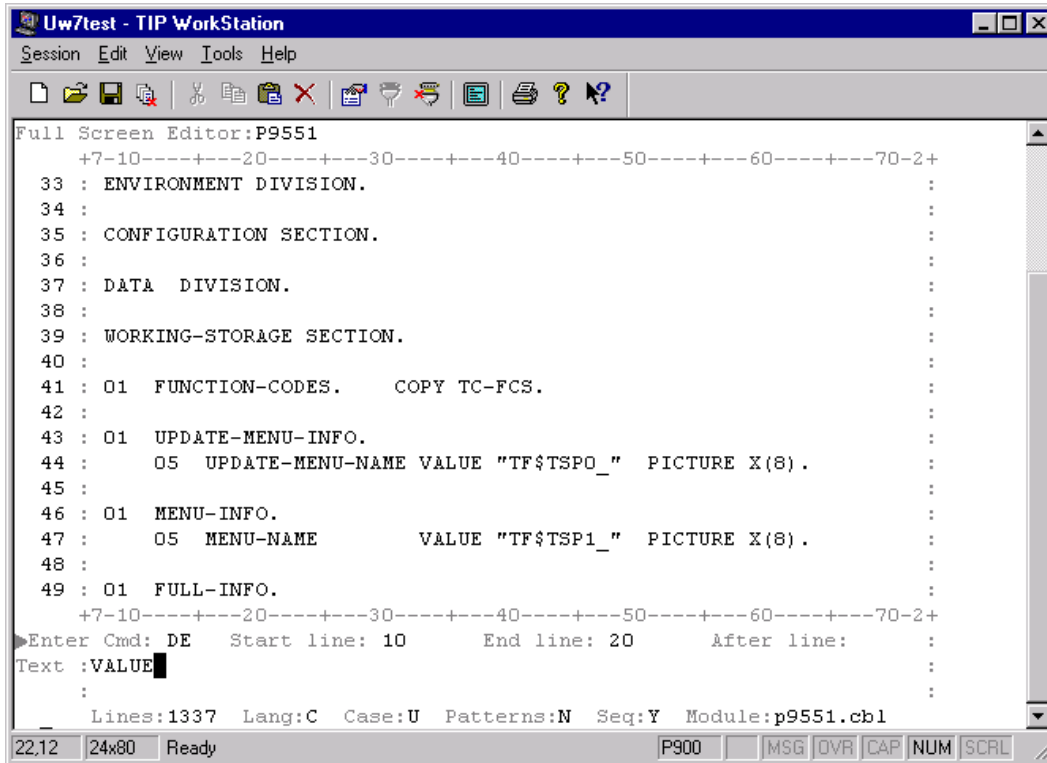
The delete command allows the user to delete a range of lines. As a precaution, **fse** does not allow the user to delete lines **UNLESS** the first line (of the range) is currently being displayed in the upper area of the display **OR** has been revealed by a PEEK command.

The user must enter "DE" as the command, the {startline} and an optional {endline}. The Full Screen Editor deletes the lines from the {startline} to the {endline} *inclusive*.

If an {endline} is not specified, the DE command deletes only one line (the {startline}).

The delete command may be limited to lines that contain a certain string by entering a string in {text1}. Lines may be deleted if they do **NOT** contain a certain string by prefixing the string with an exclamation mark.

For example, to delete lines 10 through 20 only if they contain the string "VALUE", enter DE as the command, 10 as the {startline}, 20 as the {endline}, and **VALUE** in {text1}.

Example:


```

Full Screen Editor:P9551
+7-10-----+---20-----+---30-----+---40-----+---50-----+---60-----+---70-2+
33 : ENVIRONMENT DIVISION.                :
34 :                                       :
35 : CONFIGURATION SECTION.              :
36 :                                       :
37 : DATA DIVISION.                      :
38 :                                       :
39 : WORKING-STORAGE SECTION.             :
40 :                                       :
41 : 01 FUNCTION-CODES.      COPY TC-FCS.  :
42 :                                       :
43 : 01 UPDATE-MENU-INFO.                :
44 :    05 UPDATE-MENU-NAME VALUE "TF$TSPO_" PICTURE X(8). :
45 :                                       :
46 : 01 MENU-INFO.                        :
47 :    05 MENU-NAME          VALUE "TF$TSP1_" PICTURE X(8). :
48 :                                       :
49 : 01 FULL-INFO.                        :
+7-10-----+---20-----+---30-----+---40-----+---50-----+---60-----+---70-2+
Enter Cmd: DE  Start line: 10      End line: 20      After line:  :
Text :VALUE█                                         :
:                                                     :
_  Lines:1337 Lang:C Case:U Patterns:N Seq:Y Module:p9551.cb1
22.12 24x80 Ready P900 MSG DVR CAP NUM SCRL

```

This example illustrates deleting lines that contain the string VALUE (5 characters) within lines 10 through 20 inclusive.

Lines not containing the string VALUE could be deleted by specifying the string as **!VALUE**.

The qualification string may also make use of a column specification to select lines containing (or not containing) the string within a specified range of columns.

DU - Duplicate Lines

The duplicate command allows one or more lines to be duplicated a specified number of times. The duplicated line(s) always immediately follow the original set of lines.

For example, to duplicate lines 5 through 8, 4 times, enter DU as the command, 5 as the {startline}, 8 as the {endline}, and 4 in the {afterline} field:

Example:

```

Uw7test - TIP WorkStation
Session Edit View Tools Help
Full Screen Editor:P9551
+7-10-----+20-----+30-----+40-----+50-----+60-----+70-2+
33 : ENVIRONMENT DIVISION.
34 :
35 : CONFIGURATION SECTION.
36 :
37 : DATA DIVISION.
38 :
39 : WORKING-STORAGE SECTION.
40 :
41 : 01 FUNCTION-CODES. COPY TC-FCS.
42 :
43 : 01 UPDATE-MENU-INFO.
44 : 05 UPDATE-MENU-NAME VALUE "TF$TSPO_" PICTURE X(8).
45 :
46 : 01 MENU-INFO.
47 : 05 MENU-NAME VALUE "TF$TSP1_" PICTURE X(8).
48 :
49 : 01 FULL-INFO.
+7-10-----+20-----+30-----+40-----+50-----+60-----+70-2+
Enter Cmd: DU Start line: 5 End line: 8 After line: 4
Text :
Lines:1337 Lang:C Case:U Patterns:N Seq:Y Module:p9551.cb1
22.7 24x80 Ready P900 MSG DVR CAP NUM SCRL
    
```

As a safety feature, **fse** does not allow the creation of more than 500 new lines with a DU command.

En - End Full Screen Editor

The end command signals **fse** that the user has completed all desired editing. **Fse** terminates normally and retains the workspace for potential future editing.

Fse clears the screen and issues a message indicating that the workspace is retained. The text of this message includes the group name and buffer name of the retained workspace (edit buffer) as a reminder to the user.

MSG WAIT is interpreted as the "E" command.

EX - Execute TIP/ix Command Lines

The EX command "executes" one or more lines of text as if the text is a TIP/ix command line entry. **fse** uses the TIPSUB facility of TIP/ix to execute either a range of lines or one or two lines provided in {text1} and {text2}.

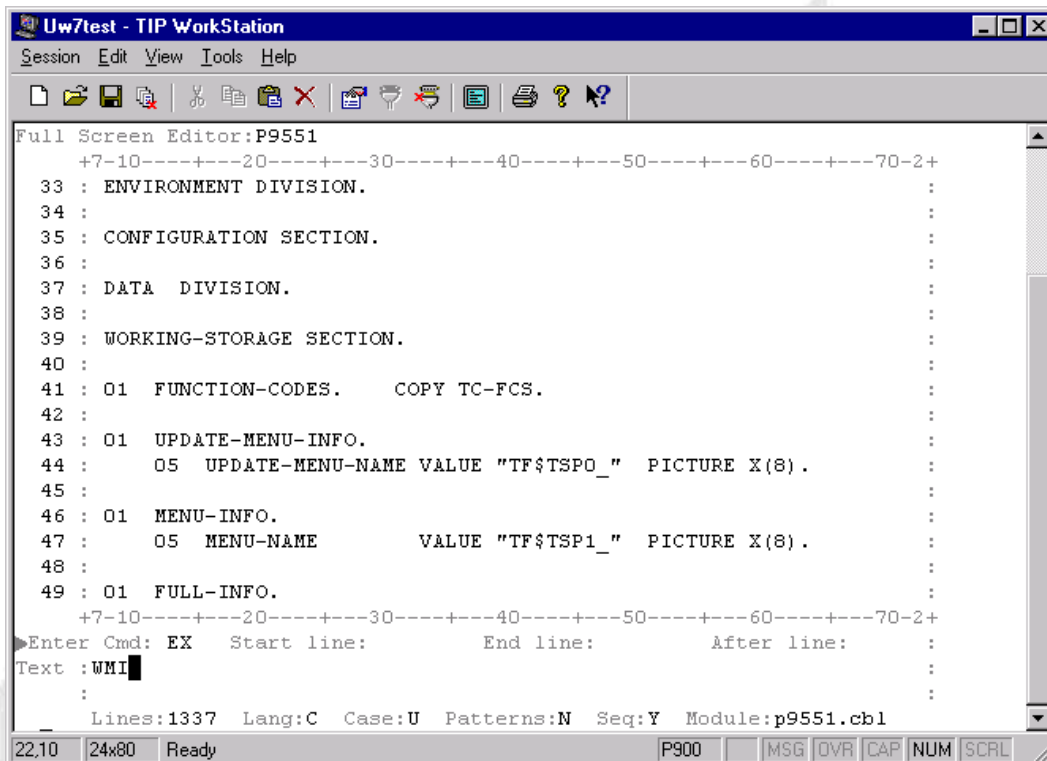
This command eliminates the need to exit the editor (with the "End" command) to be able to run other transactions.

The command line (or lines) to be executed must conform to standard TIP/ix command line structure (no leading spaces; a valid transaction code, etc).

Comment lines (these vary according to the language of the text being edited) **are** executed - note the transaction name must immediately follow the comment character (for example: "*WMI" starting in column 1 for Assembler or column 7 for COBOL).

Use the commands "[SH](#)" or "[UX](#)" to execute UNIX commands from **fse**. "EX" only executes TIP/ix programs defined with [SMPROG](#) (and [SMSEC](#)).

Example:



```

Uw7test - TIP WorkStation
Session Edit View Tools Help
Full Screen Editor:P9551
+7-10-----20-----30-----40-----50-----60-----70-2+
33 : ENVIRONMENT DIVISION.
34 :
35 : CONFIGURATION SECTION.
36 :
37 : DATA DIVISION.
38 :
39 : WORKING-STORAGE SECTION.
40 :
41 : 01 FUNCTION-CODES. COPY TC-FCS.
42 :
43 : 01 UPDATE-MENU-INFO.
44 : 05 UPDATE-MENU-NAME VALUE "TF$TSP0_" PICTURE X(8).
45 :
46 : 01 MENU-INFO.
47 : 05 MENU-NAME VALUE "TF$TSP1_" PICTURE X(8).
48 :
49 : 01 FULL-INFO.
+7-10-----20-----30-----40-----50-----60-----70-2+
Enter Cmd: EX Start line: End line: After line:
Text :WMI
:
Lines:1337 Lang:C Case:U Patterns:N Seq:Y Module:p9551.cb1
22,10 24x80 Ready P900 MSG OVR CAP NUM SCRL

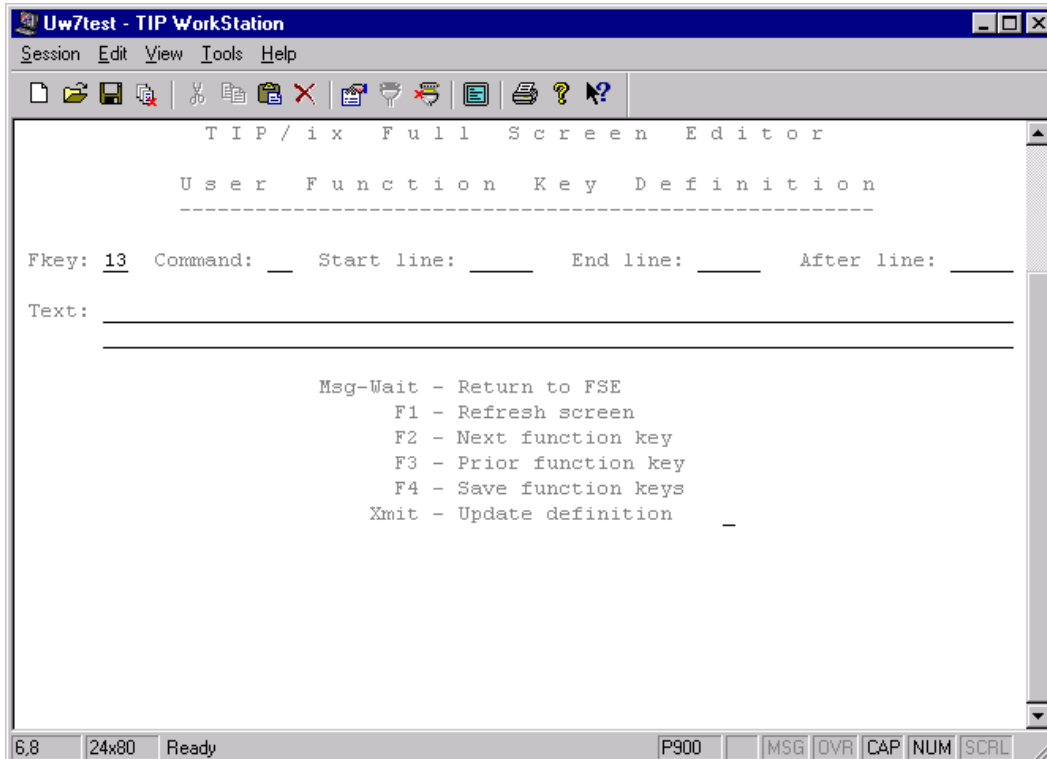
```

OR

For example, assume that you often wish to advance the display by just 8 lines (a sort of half page advance). The hard way is to enter "+" as the command and "8" in the {startline} field.

To avoid repeatedly keying this command might be to equate (say) function key #13 (a handy lower case function key on a UTS400) to this command.

To accomplish this, enter "F#" as the command and 13 in the {startline} field. **fse** responds by displaying the following screen format:



Notice that the function key that you specified in the main **fse** screen (13 in this example) has been carried forward to this screen.

You may now proceed to fill in {startline}, {endline}, {afterline}, {text1} and {text2} as appropriate for the command that you are assigning to the function key you have selected.

The command that is equated to the function key can be quite complex and might include text information (a search string for example).

To continue our example of half-screen forward paging, we enter:



and press **XMIT** to update the definition of F#13 (in this case).

At this point, the definition of F#13 is updated, **but** the definition of F#13 is purely local to this session. Once you return to the **fse** editing screen (more on that in a moment), you find that pressing **F13** results in the automatic execution of the "+" command with "8" in {startline}. The command is executed directly without being displayed, but it may be "recalled" using the RC command.

To define other keys you may use **F2** or **F3** as advertised in the screen format shown above.

To save the function keys in your personal [DEFKEY](#) file, press **F4** whenever you have finished defining keys; such saved keys remain permanently in effect for YOU until such time as you discard the DEFKEY file or re-save the **fse** function definitions. Whenever you use **fse**, those saved function keys are automatically "live" and in effect.

To return to your **fse** session (whether or not you have elected to SAVE your function key definitions), press **MSG WAIT**.

FA - Find All Lines Containing a String

The find all command is used to search for all lines that contain a given string. The lines are displayed one page at a time and may be updated by altering them on the screen and pressing **XMIT** with the cursor in the first resting location.

FI - Find Lines Containing a String

The FI command is used to search the workspace (in a forward direction - toward the **end** of the workspace) for the next line matching a specified string. A {startline} may be given to start searching from that point. The FI command does not "wrap around" when it reaches the last line of the workspace.

If a line is found that matches the string, **fse** re-displays the text - beginning with the line that matched the search string.

The default {startline} for the FI command is the *second* line currently displayed in the upper area of the screen.

The user may take advantage of this fact by issuing FI repeatedly to "step" through occurrences of a string (since the second and subsequent FI commands begin looking from the line *after* the first line on the screen, one automatically avoids finding the same line over and over).

To find, for example, the beginning of the PROCEDURE DIVISION in a COBOL program, enter FI as the command, "1" as {startline}, and "PROCEDURE" in {text1}:

Example:

The screenshot shows a window titled "Uw7test - TIP WorkStation" with a menu bar (Session, Edit, View, Tools, Help) and a toolbar. The main area is a "Full Screen Editor:NEW" with a grid. The grid shows a search command being entered: "Enter Cmd: FI Start line: 1 End line: After line:" followed by "Text :PROCEDURE". The status bar at the bottom shows "22,16 24x80 Ready" and "P900 MSG OVR CAP NUM SCRL".

A more efficient approach is to take advantage of the fact that COBOL division names must begin in column 8:

Additional Considerations:

The string to be found may be prefixed by a column specification to limit the search for the string to the column range specified. The {text2} field is not used by this command (the search string, with or without a column range, must be supplied in {text1}).

Example:

Search for the string "LBL" starting in column 25.

```
25LBL
```

FM - Find Lines Containing a String

The FM command functions in the same manner as the FI command, with one difference: when the desired line is found, the matching line is displayed in the middle of the upper area of the screen rather than in line 1 (hence the command mnemonic **F**ind **M**iddle).

If {startline} is not specified, the starting line for the search defaults to the second line that is currently displayed in the upper area of the screen format.

He - Help for Full Screen Editor

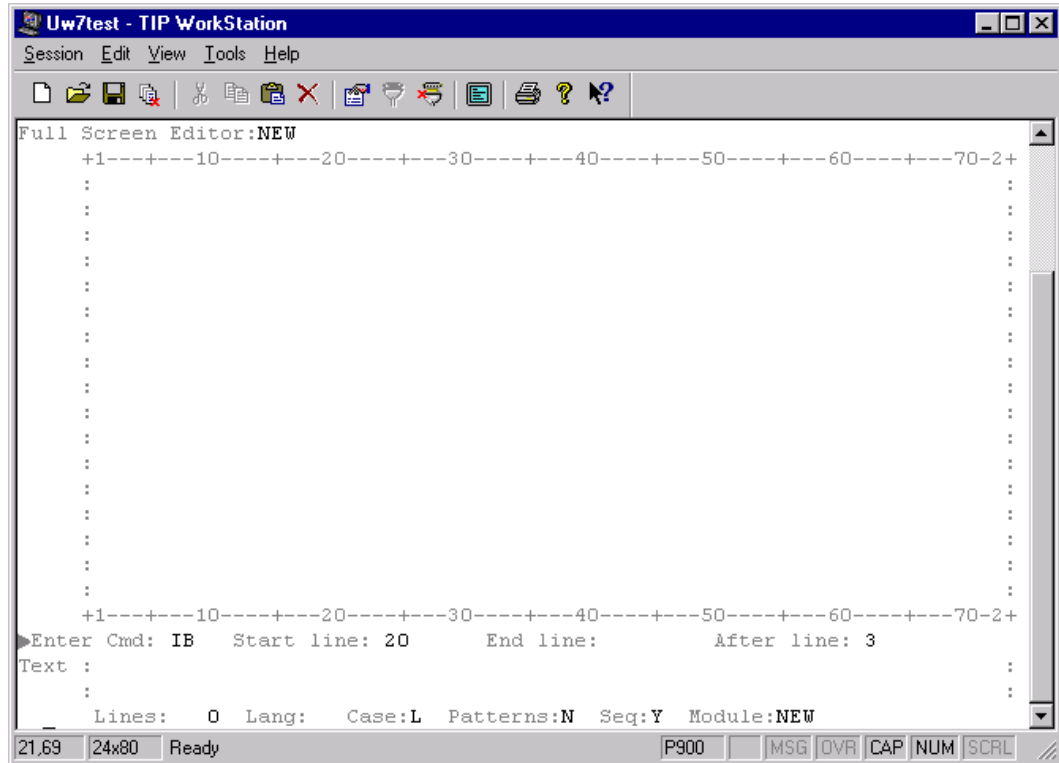
The H command displays help information about fse commands. This information is not shown here since it may change from time to time. The same HELP information may be solicited by using the TIP/ix help processor.

IB - Insert Empty Lines Before

The IB command inserts a specified number of (empty) lines ahead of the specified {startline}. These lines can then be filled in and updated by pressing **XMIT** with the cursor in the first resting location.

The {afterline} field is used to specify the number of lines to insert. If {afterline} is not specified, one blank line is inserted.

To insert 3 blank lines ahead of line 20: enter "IB" as the command, 20 as {startline}, and 3 as {afterline}.

Example:


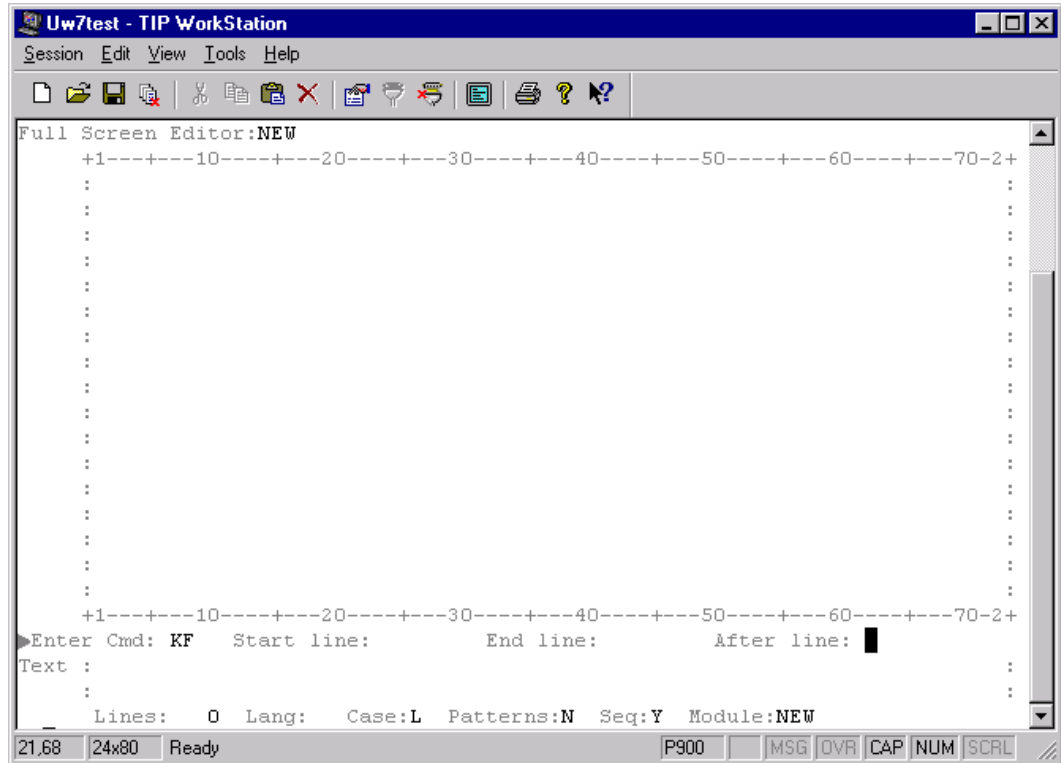
In - Insert Empty Lines After

The I command inserts the specified number of blank (empty) lines following the {startline}. These lines can then be filled in and updated by pressing **XMIT** with the cursor in the first resting location.

The {afterline} field is used to specify the number of lines to insert. If {afterline} is not specified, one blank line is inserted.

For example, to insert 5 blank lines after line 24, enter "I" as the command, 24 as {startline}, and 5 as {afterline}.

Example:



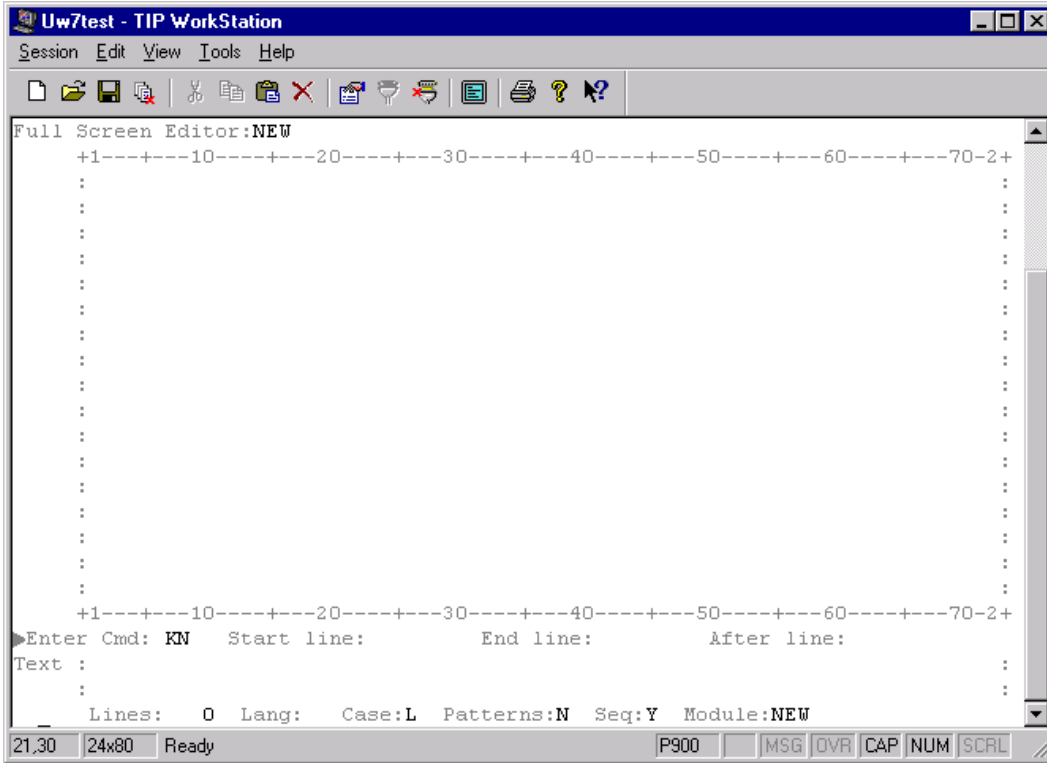
KN - Keep Commands On

If you intend to use a command repeatedly, perhaps altering it with each new execution, enter the KN command *first*.

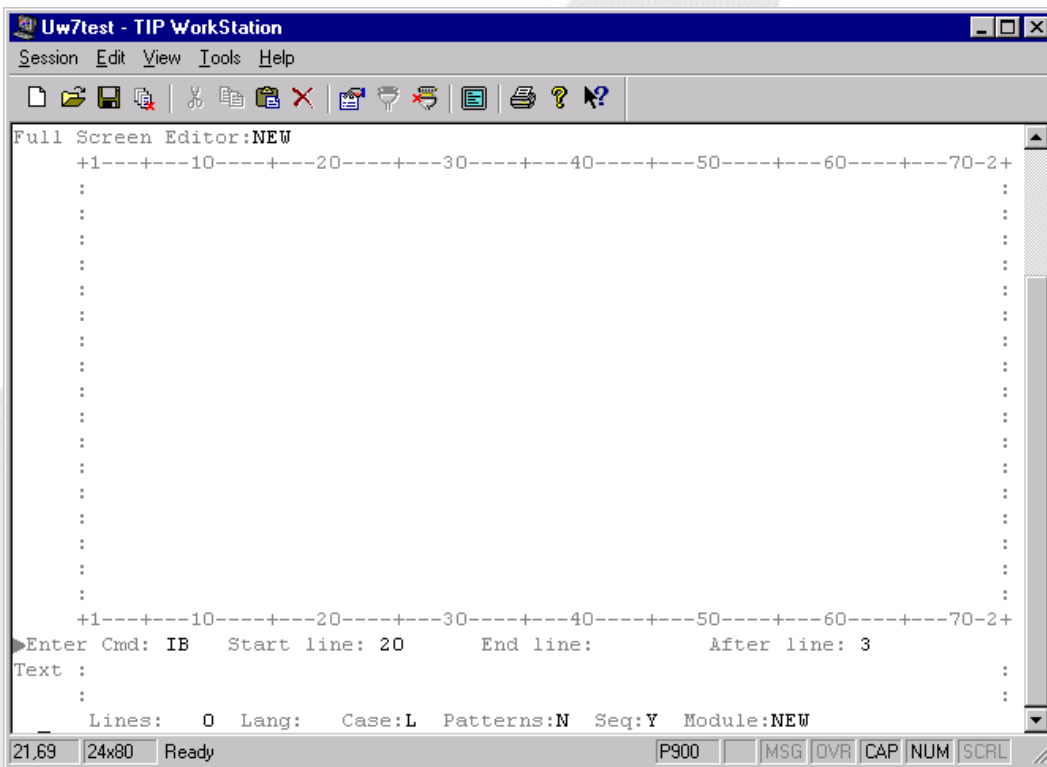
As you execute each command, **fse** will re-display it for further use.

Example:

First enter:

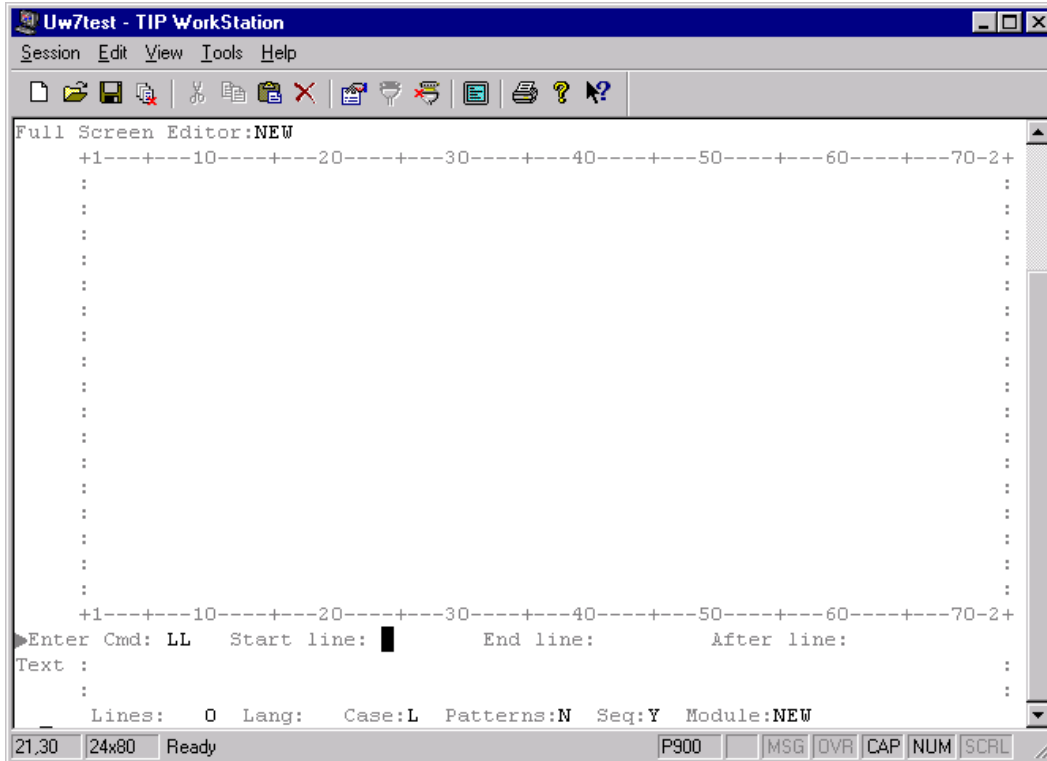


and press **XMIT**. Next enter the command you wish to use:



Example:

- If you specify a {startline}, fse displays lines finishing with the specified {startline}.
- If you do not specify a {startline}, fse displays lines finishing with the last line of the workspace.

**MA - Margin Set**

The margin command establishes new left and right margins for fse. The new left margin column is specified in {startline} and the new right margin column is specified in {endline}.

If either field is left empty, the default margin (left or right) is used in its place.

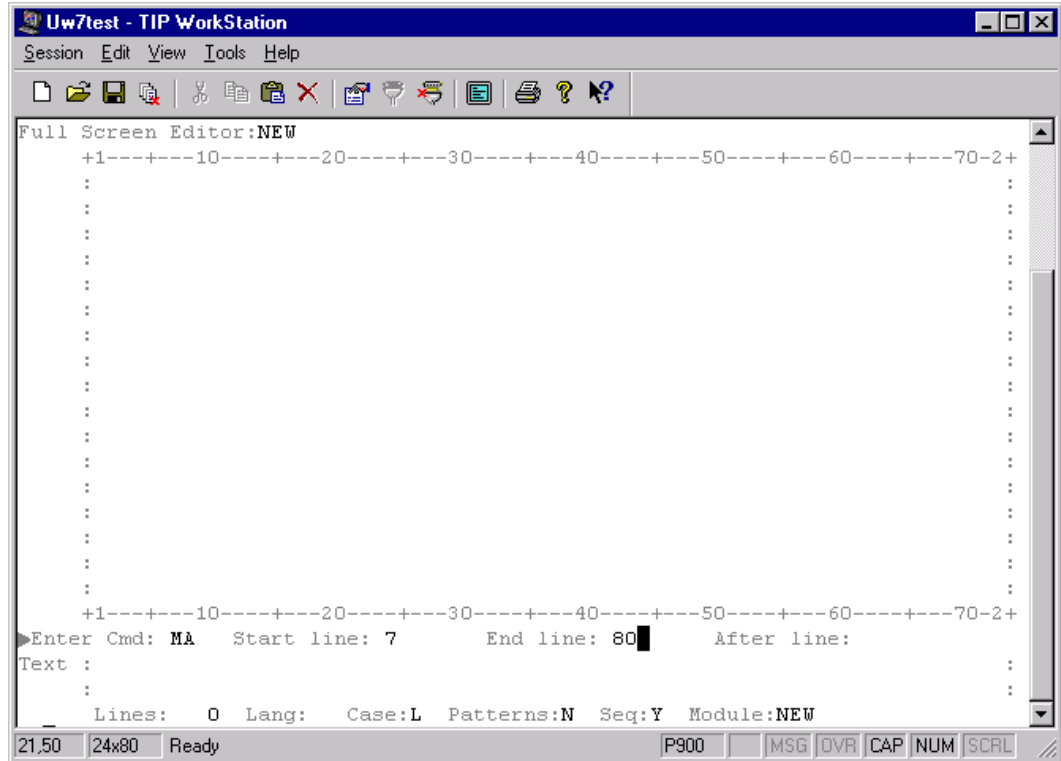
The error message: "Margin number exceeds record length" is displayed if the right margin is beyond the declared record length (established when **fse** is invoked).

The default margins used by **fse** are:

```
Language "COBOL" : 7 through 72 inclusive
Language "RPG"   : 6 through 74 inclusive
none of the above : 1 through 72 inclusive
```

The margin command may be used to gain access to columns that are normally not accessible. For example, for COBOL text, one could set the margins to 7 and 80 and then modify data in columns 73-80.

Example:



MB - Move Lines Before

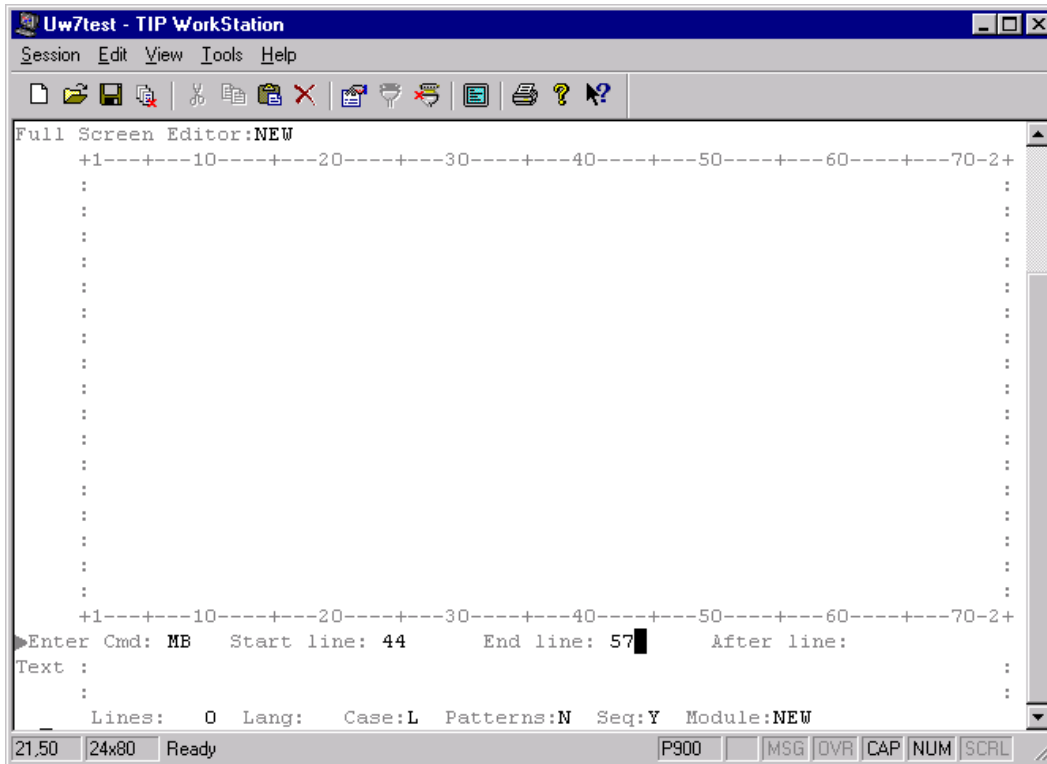
The MB command is identical to the CB (copy before) command with the exception that the moved lines are NOT left in their previous location.

The MB command requires the {startline}, {endline}, and {afterline} numbers be specified.

An optional string may be specified in the first text line to move only lines that match the specified string.

Example:

This example illustrates moving lines 44 through 57 (inclusive) before line 208.



MC - Move Constant or Columns

The move columns (move constant) command allows the user to move a range of columns OR a string constant to a range of columns.

The {startline} and {endline} may be specified to limit the scope of the MC command. Default is all lines that are currently displayed in the upper area of the screen.

The {afterline} field is not used by the MC command.

{text1} must contain the character string OR the column range that is to be moved. {text1} may contain:

- a string representing a constant (for example: ABC or '12')
- a single column number (for example: 13)
- a column range (for example: 10:20)

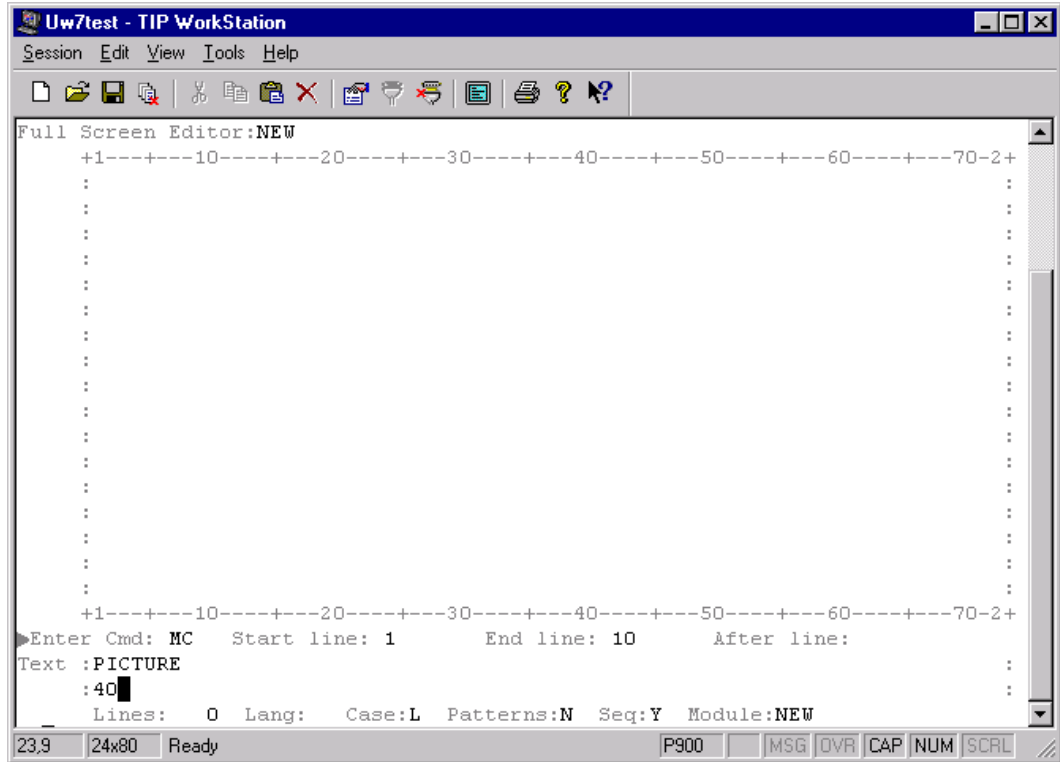
{text2} must contain a single column or a column range representing the column(s) to be altered.

If the text to be moved is not the same length as implied by the receiving column(s), the text is padded with spaces or truncated as appropriate.

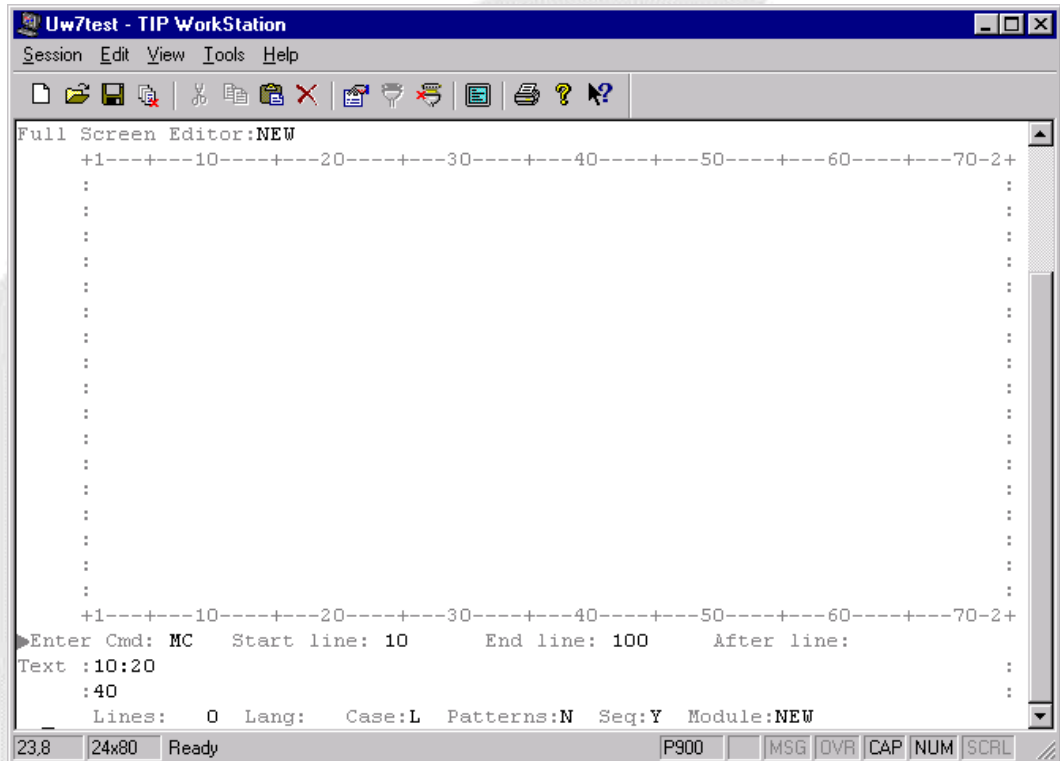
Examples:

- This example illustrates moving the string "PICTURE" (7 characters) to column 40 of line 1 through 10 inclusive. The text that was in

columns 41 through the right margin is shifted right 7 positions to accommodate the new text (any overflow falls off the end into the bit bucket). The text in column 40 is overwritten.



- This example illustrates moving the contents of columns 10 through 20 (inclusive) to column 40 in lines 10 through 100 (inclusive).



MF - Marker Off

The MF command prevents **fse** from writing a marker line prior to the edit buffer text when a write ([WR](#), [WE](#), [WN](#), [WQ](#), [WX](#), or [WZ](#)) command is issued.

Marker lines only apply to language type of C (COBOL). They are written as a comment line so as not to affect the module. Fse uses information on this line to determine the optional settings for the module when the edit session begins. The [SE](#) (set) command can be used to change the options. However, if the marker line is disabled (default mode for **fse**) then the options will have to be reset for each edit session.

Use the [MN](#) (marker on) command to have **fse** write a marker line ahead of the module text when a write command is issued. Once a module is written with a marker line **fse** will retain it for all future edit sessions (until MF is run prior to a write command).

The MF command does not require (or acknowledge) any entry in the {startline}, {endline}, {afterline}, {text1} or {text2} fields.

MN - Marker On

The MN command causes **fse** to write a marker line prior to the edit buffer text when a write (WR, WE, WN, WQ, WX, or WZ) command is issued. Marker lines are only ever written for language types of C (COBOL) or R (RPG). The MN command has no affect for other language types.

The marker line is written as a comment line so as not to affect the module. **fse** uses information on this line to determine what options are to be for the module when the edit session begins. The SE (set) command can be used to change the options. However, if the marker line is disabled (default mode for **fse**) then the options will have to be reset for each edit session.

Once a module is written with a marker line **fse** will retain it for all future edit sessions (until the MF command is run prior to a write command).

The MN command does not require (or acknowledge) any entry in the {startline}, {endline}, {afterline}, {text1} or {text2} fields.

An example of an **fse** marker line for a language type of COBOL is:

```
* FSE$-003$C$U$N$Y$DATE $Y$16$04$
```

The asterisk (in column 7 for COBOL) is used to indicate to COBOL that this is a comment line. The rest of the information represents the **fse** option settings for this module.

Item	Description
\$FSE\$-	Indicates that this is an fse marker line.
003	Represents the fse version number for this module. fse increments the version on the marker line when the module is read into the edit buffer. Therefore, the updated version number will appear in the file when the module is written.
C	Indicates that this is a COBOL program
U	Indicates lines added or changed are forced to upper case.
N	Indicates that pattern matching is not active for this module. See the description of pattern matching
Y	Indicates that fse will automatically sequence the lines on a write command. (For COBOL this is columns 1-6).
DATE	Indicates that fse will place the current date in YY/MM/DD format in columns 73:80. of updated lines when the edit buffer is written to a file or library module.
Y	Display line register number instead of actual line number.
16	Number of lines in a page. Used by F2 and F3 commands.
04	Number of spaces to replace tab characters with.

MO - Move Lines After

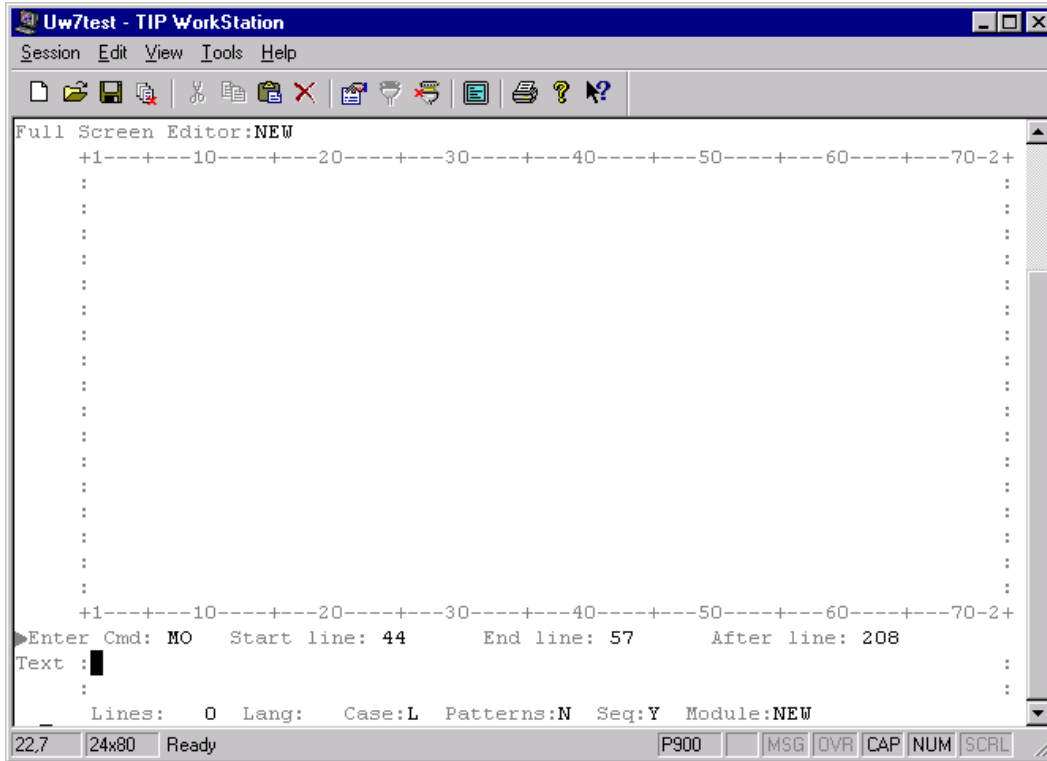
The MO command is identical to the copy command "[CO](#)" with the exception that the moved lines are NOT left in their previous location.

The move command requires the {startline}, {endline}, and {afterline} numbers be specified.

An optional string may be specified in the first text line to move only lines that match the specified string.

Example:

This example illustrates moving lines 44 through 57 (inclusive) after line 208.



O - Set Language " "

The O command sets the language code of the text being edited to space (unspecified). The O command has exactly the same effect as using the SE command (see [SE Set Options](#)) and modifying the language field to a space.

The O command does not require (or acknowledge) any entry in the {startline}, {endline}, {afterline}, {text1} or {text2} fields.

OA - Set Language "A"

The OA command sets the language code of the text being edited to "A" (Assembler). The OA command has exactly the same effect as using the SE command (see [SE Set Options](#)) and modifying the language field to "A".

The OA command does not require (or acknowledge) any entry in the {startline}, {endline}, {afterline}, {text1} or {text2} fields.

OC - Set Language "C"

The OC command sets the language code of the text being edited to "C" (COBOL). The OC command has exactly the same effect as using the SE command (see [SE Set Options](#)) and modifying the language field to "C".

The OC command does not require (or acknowledge) any entry in the {startline}, {endline}, {afterline}, {text1} or {text2} fields.

OD - Set Language "D"

The OD command sets the language code of the text being edited to "D" (Document). The OD command has exactly the same effect as using the SE command (see [SE Set Options](#)) and modifying the language field to "D".

The OD command does not require (or acknowledge) any entry in the {startline}, {endline}, {afterline}, {text1} or {text2} fields.

OL - Option Literal

The option literal command provides a fast and simple method to declare that subsequent input text (including any search strings!) is to be interpreted in the case it was entered at the keyboard.

The SE command (see [SE Set Options](#)) can also be used to accomplish this change.

The OL command does not require (or acknowledge) any entry in the {startline}, {endline}, {afterline}, {text1} or {text2} fields.

This setting is retained by **fse** with the edit buffer. However, to keep this information with the library module or UNIX file it is necessary to issue the [MN](#) (marker on) command prior to writing the module. This option is only available for COBOL and RPG.

OP - Set Language "P"

This is for Cobol source. However, if *case* is set to "U", **fse** translates the current screen as follows:

- Comments are not translated.
- Literal strings are not translated.
- Everything else is translated.

The OP command sets the language code of the text being edited to "P". The OP command has exactly the same effect as using the SE command (see [SE Set Options](#)) and modifying the language field to "P".

The OP command does not require (or acknowledge) any entry in the {startline}, {endline}, {afterline}, {text1} or {text2} fields.

OR - Set Language "R"

The OR command sets the language code of the text being edited to "R" (RPG). The OR command has exactly the same effect as using the SE command (see [SE Set Options](#)) and modifying the language field to "R".

The OR command does not require (or acknowledge) any entry in the {startline}, {endline}, {afterline}, {text1} or {text2} fields.

OT - Set Language "T"

The OT command sets the language code of the text being edited to "T" (Text). The OT command has exactly the same effect as using the SE command (see [SE Set Options](#)) and modifying the language field to "T".

The OT command does not require (or acknowledge) any entry in the {startline}, {endline}, {afterline}, {text1} or {text2} fields.

OU - Option Upper

The option upper command provides a fast and simple method to declare that subsequent input text (including any search strings!) is to be translated into upper case. That is, any alphabetic characters in the text are automatically translated into upper case.

The SE command (see [SE Set Options](#)) can also be used to accomplish this change.

The OU command does not require (or acknowledge) any entry in the {startline}, {endline}, {afterline}, {text1} or {text2} fields.

This setting is retained by **fse** with the edit buffer. However, to keep this information with the library module or UNIX file it is necessary to issue the [MN](#) (marker on) command prior to writing the module. This option is only available for COBOL and RPG.

OX - Set Language "X"

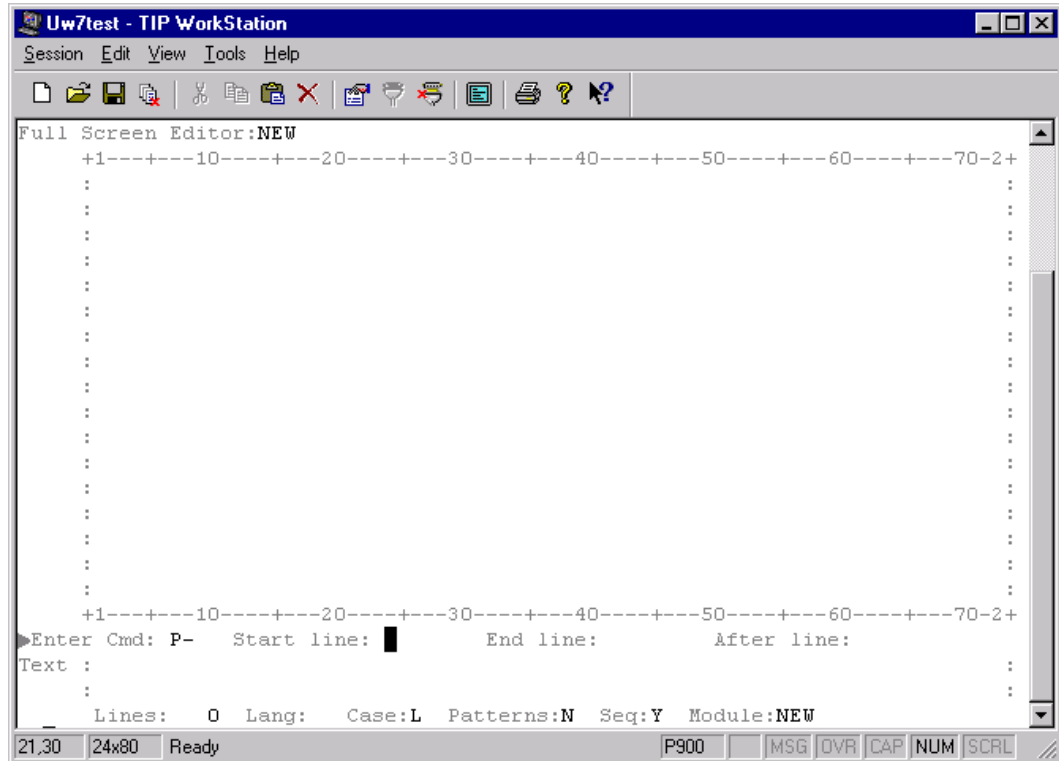
The OX command sets the language code of the text being edited to "X". The OX command has exactly the same effect as using the SE command (see [SE Set Options](#)) and modifying the language field to "X".

The OX command does not require (or acknowledge) any entry in the {startline}, {endline}, {afterline}, {text1} or {text2} fields.

P- - Peek Scroll Down

The P- command is a simple method to scroll (in a descending sense) the lines that have been displayed with a previous PEEK command. For example, if the user issues a P- command after issuing a PE(13) command, he is shown lines (12,13). A subsequent P- command displays lines (11,12) and so on.

This command is useful for line fishing (but gets a little tedious if you aren't close the first time).

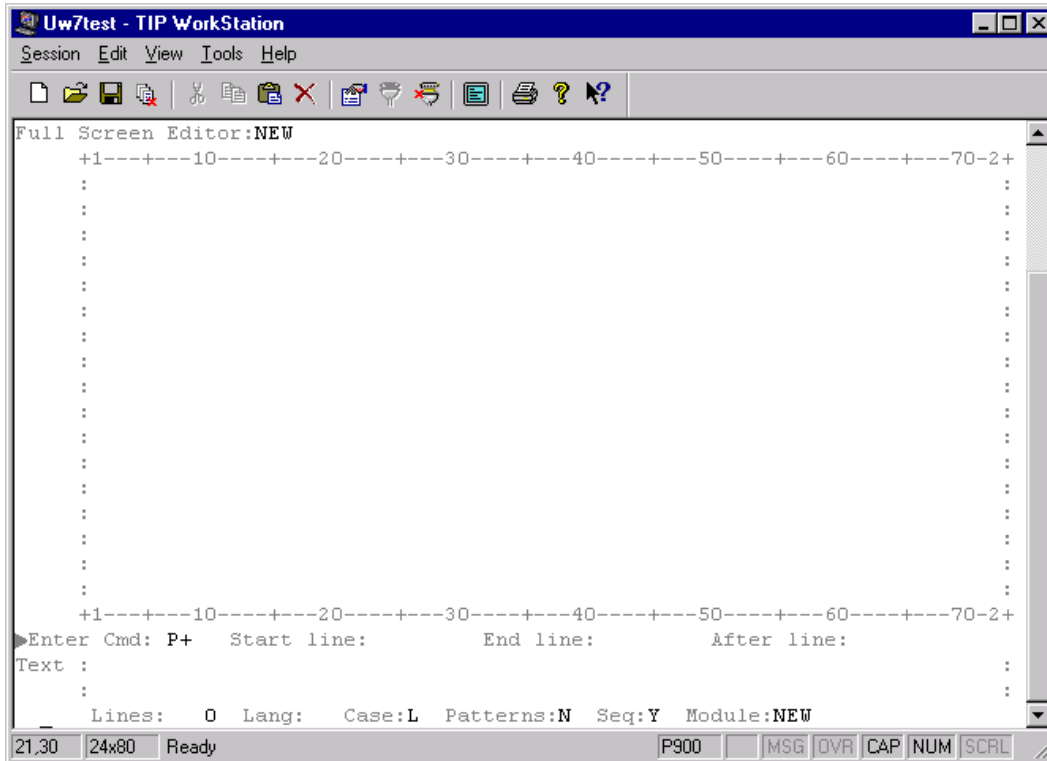
Example:


P+ - Peek Scroll Up

The P+ command is a simple method to scroll (in an ascending sense) the lines that have been displayed with a previous PEEK command. For example, if the user issues a P+ command after issuing a PE(13) command, he is shown lines (14,15). A subsequent P+ command displays lines (15,16) and so on.

This command is useful for line fishing (but gets a little tedious if you aren't close the first time).

Example:



PE - Peek at Line

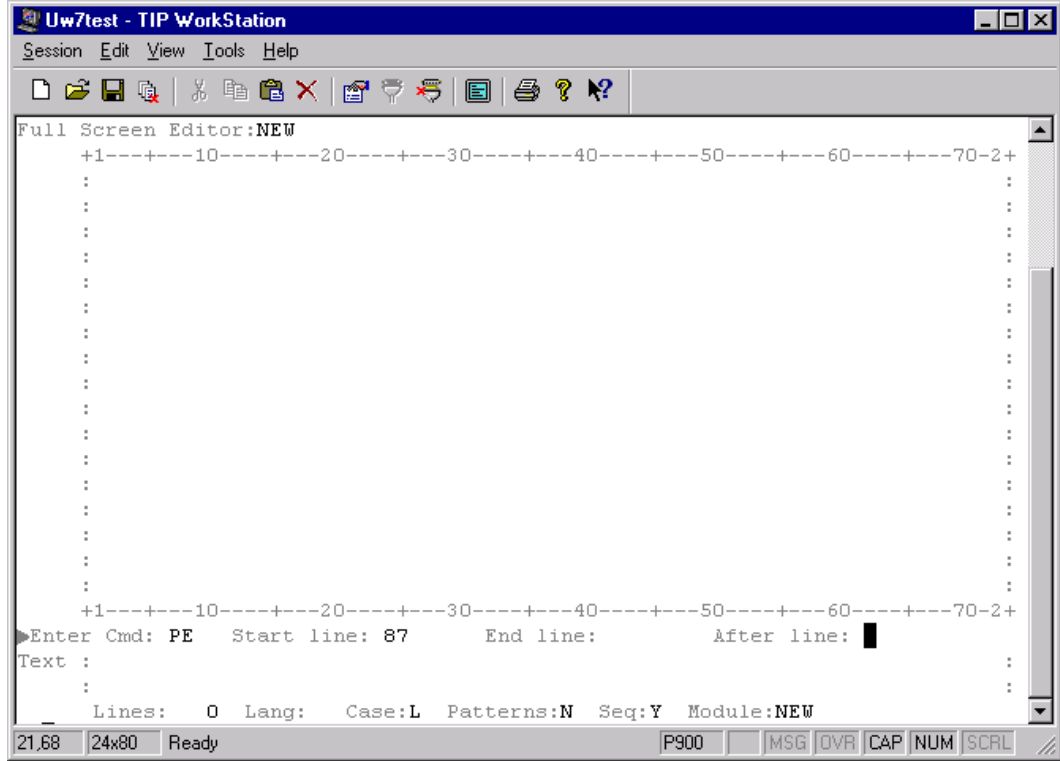
The PE command allows the user to view one or two lines without disturbing the lines that are currently displayed in the upper area of the screen format. The command expects a value to be entered in {startline}. This line (and the line that follows it) are displayed in {text1} and {text2}.

This command is useful when you think you remember a line number (how many move or copy commands have been "off by just one line?").

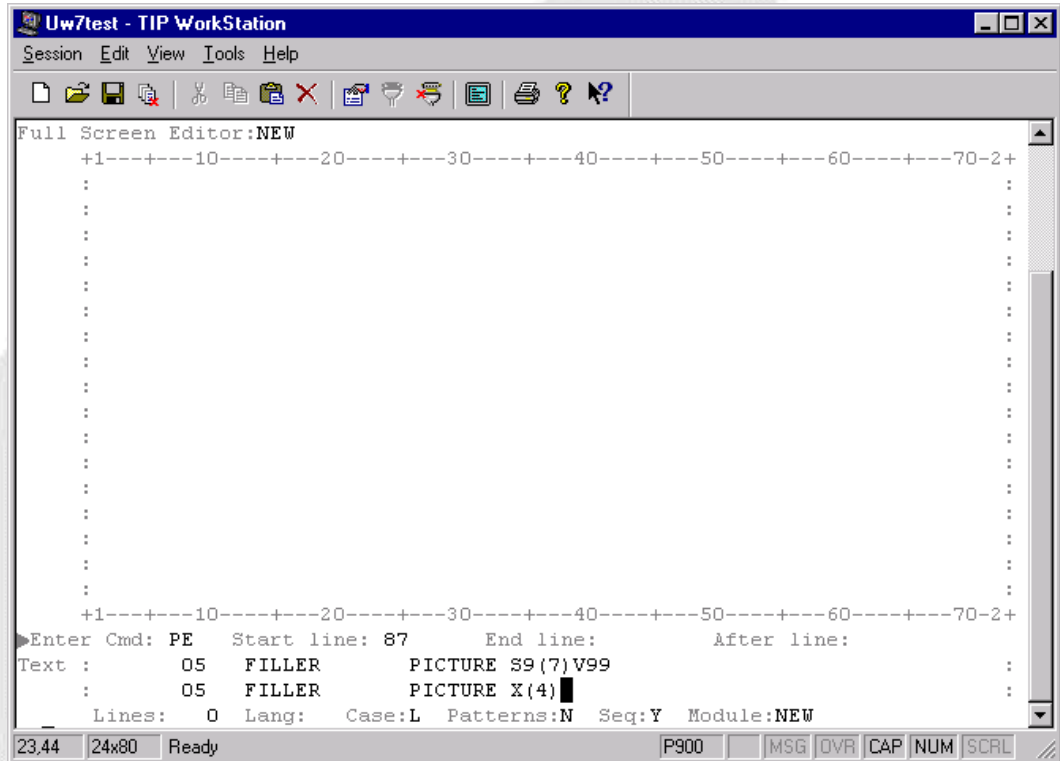
The (two) lines of text that are revealed in this manner may then be added by entering an Add command (the text is already in the appropriate area for the Add command!).

Example:

This example illustrates peeking at line 87 (and 88 — if it exists).



After pressing **XMIT** (to enter the PE[87] command), the display might look like this (with lines 87 and 88 displayed):



Pr - Print Lines

The Print command is identical to the List command described in a previous section. It does not actually print the lines, but displays them on the terminal. This command is provided because some editors interpret a "print" command as a "display on the screen" function.

Qu - Quit Full Screen Editor

The quit command causes the Full Screen Editor to end the editing session AND discard the edit workspace.

If changes have been made to the contents of the workspace and the changes have not been written, **fse** warns the user and asks for confirmation of the QUIT command.

The QUIT command is normally used after a write command has copied the edit buffer to a TIP/ix library or a UNIX file (to save the contents of the edit buffer).

Pressing **F4** is equivalent to this command.

RC - Recall Last Command

The RC command re-displays the last command that was entered in the command area (lower portion) of the **fse** screen format. Some **fse** commands clear the lower area of the screen format; the RC command allows the user to recall the last command (and therefore resubmit the command with or without modification).

Additional Considerations:

- **fse** will not allow you to save an invalid command for recall.

Re - Read from TIP/ix Library, UNIX file, or Edit Buffer

The read command allows lines to be read from a TIP/ix library module, UNIX file, or Edit buffer into the **fse** workspace. The name of the module to be read is expected to be in the {text1} field in the lower area of the screen format. A default value can only be used if the workspace is empty.

A {startline} and/or {endline} may be specified to indicate that only a subset of the file is to be read.

The {startline} defaults to 1 if it is not specified; {endline} defaults to 9999 (meaning the last line of the input).

The lines that are read are copied immediately after the line number specified in the {afterline} field of the command (or at the end of the current workspace if no {afterline} is specified).

The name of the module to read is expected to be in {text1}. The module name is expected to comply with one of the following formats:

```
<library> <module>
<group> <buffer> <e|E>
<buffer> <e|E>
<unixfile>
```

To force "abc/def" to be interpreted as a UNIX file it is necessary to add the prefix "./". So instead enter "./abc/def". Otherwise fse will attempt to read the module "def" in the library "ABC". (Library names are converted to upper case.)

Alternatively, the SE command (see [SE Set Options](#)) may be used before using the Read command to specify the file name. This method can only be used if edit buffer is empty.

Example 1:

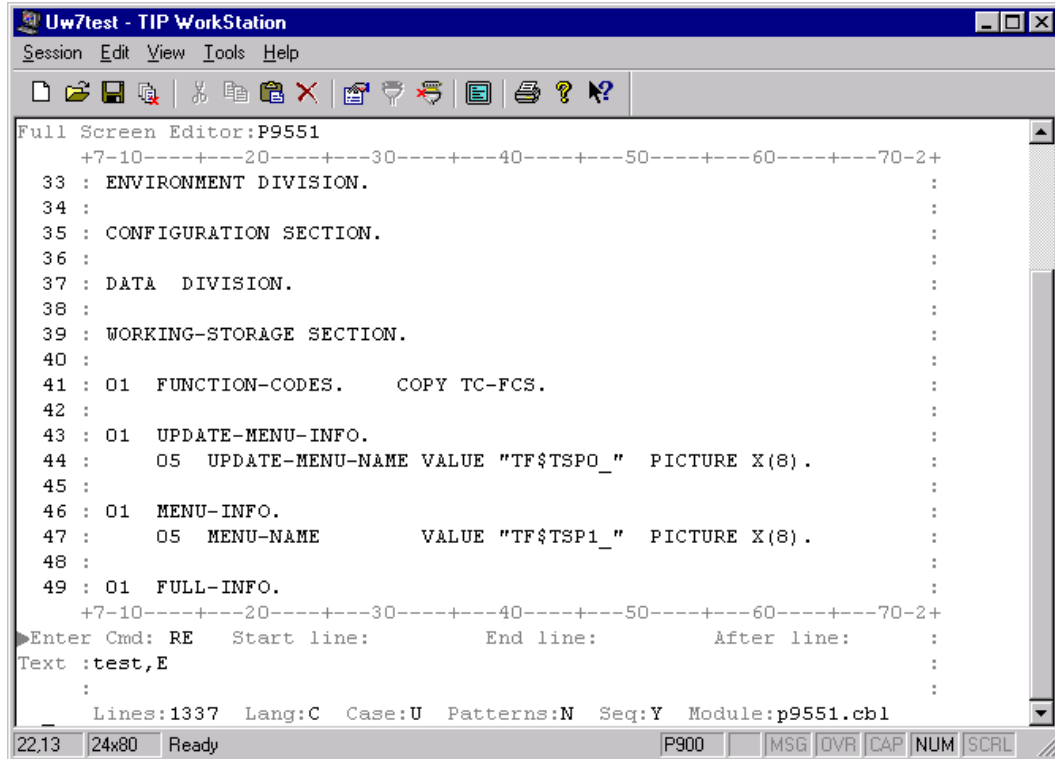
This example illustrates reading (all lines by default) of the source element "pay020" from the current UNIX directory. The lines read in are placed immediately following line 100 of the workspace.

```
Uw7test - TIP WorkStation
Session Edit View Tools Help
Full Screen Editor:P9551
+7-10-----20-----30-----40-----50-----60-----70-2+
33 : ENVIRONMENT DIVISION.
34 :
35 : CONFIGURATION SECTION.
36 :
37 : DATA DIVISION.
38 :
39 : WORKING-STORAGE SECTION.
40 :
41 : 01 FUNCTION-CODES. COPY TC-FCS.
42 :
43 : 01 UPDATE-MENU-INFO.
44 : 05 UPDATE-MENU-NAME VALUE "TF$TSPO_" PICTURE X(8).
45 :
46 : 01 MENU-INFO.
47 : 05 MENU-NAME VALUE "TF$TSP1_" PICTURE X(8).
48 :
49 : 01 FULL-INFO.
+7-10-----20-----30-----40-----50-----60-----70-2+
Enter Cmd: RE Start line: End line: After line: 100 :
Text :pay020
:
Lines:1337 Lang:C Case:U Patterns:N Seq:Y Module:p9551.cb1
22,13 24x80 Ready P900 MSG OVR CAP NUM SCRL
```

This is the same as entering a module name of "./pay020".

Example 2:

This example illustrates reading (all lines by default) an edit buffer "TEST" in the user's 1st elective group. If the user has no elective groups then fse will attempt to access the edit buffer in the user's user id group.



```

Uw7test - TIP WorkStation
Session Edit View Tools Help
Full Screen Editor:P9551
+7-10----+---20----+---30----+---40----+---50----+---60----+---70-2+
33 : ENVIRONMENT DIVISION.
34 :
35 : CONFIGURATION SECTION.
36 :
37 : DATA DIVISION.
38 :
39 : WORKING-STORAGE SECTION.
40 :
41 : 01 FUNCTION-CODES. COPY TC-FCS.
42 :
43 : 01 UPDATE-MENU-INFO.
44 : 05 UPDATE-MENU-NAME VALUE "TF$TSPO_" PICTURE X(8) .
45 :
46 : 01 MENU-INFO.
47 : 05 MENU-NAME VALUE "TF$TSP1_" PICTURE X(8) .
48 :
49 : 01 FULL-INFO.
+7-10----+---20----+---30----+---40----+---50----+---60----+---70-2+
Enter Cmd: RE Start line: End line: After line:
Text :test,E
Lines:1337 Lang:C Case:U Patterns:N Seq:Y Module:p9551.cb1
22,13 24x80 Ready P900 MSG OVR CAP NUM SCRL

```

Since no "After line" was specified the edit buffer contents will be added to the end of the workspace.

Example 3:

This example illustrates reading a TIP/ix library module. The library name is "TSTLIB" and the module (or file) in that library is "pay020". Note that only the module name is case sensitive. The library name will be forced to upper case.

```

Uw7test - TIP WorkStation
Session Edit View Tools Help
Full Screen Editor:P9551
+7-10-----20-----30-----40-----50-----60-----70-2+
33 : ENVIRONMENT DIVISION.
34 :
35 : CONFIGURATION SECTION.
36 :
37 : DATA DIVISION.
38 :
39 : WORKING-STORAGE SECTION.
40 :
41 : 01 FUNCTION-CODES. COPY TC-FCS.
42 :
43 : 01 UPDATE-MENU-INFO.
44 : 05 UPDATE-MENU-NAME VALUE "TF$TSPO_" PICTURE X(8).
45 :
46 : 01 MENU-INFO.
47 : 05 MENU-NAME VALUE "TF$TSP1_" PICTURE X(8).
48 :
49 : 01 FULL-INFO.
+7-10-----20-----30-----40-----50-----60-----70-2+
Enter Cmd: RE Start line: End line: After line:
Text :tstlib,pay020
Lines:1337 Lang:C Case:U Patterns:N Seq:Y Module:p9551.cb1
22,20 24x80 Ready P900 MSG OVR CAP NUM SCRL
    
```

Since no "After line" was specified the edit buffer contents will be added to the end of the workspace.

Additional Considerations:

- A read command is considered to be a "change" of the contents of the workspace and therefore causes a subsequent Quit command to think that changes have been made.
- fse does not accept a READ command without parameters if the edit buffer contains data. This avoids reading the module again if you inadvertently enter R as a command.

SA - Sort Ascending

The sort ascending command sorts lines into ascending order using a specified column as the start of the "sort key". The sort uses the standard ASCII character set as the collating sequence.

The sort is NOT a "stable" sort — lines that have identical "sort keys" do not necessarily remain in their original sequence (with respect to each other).

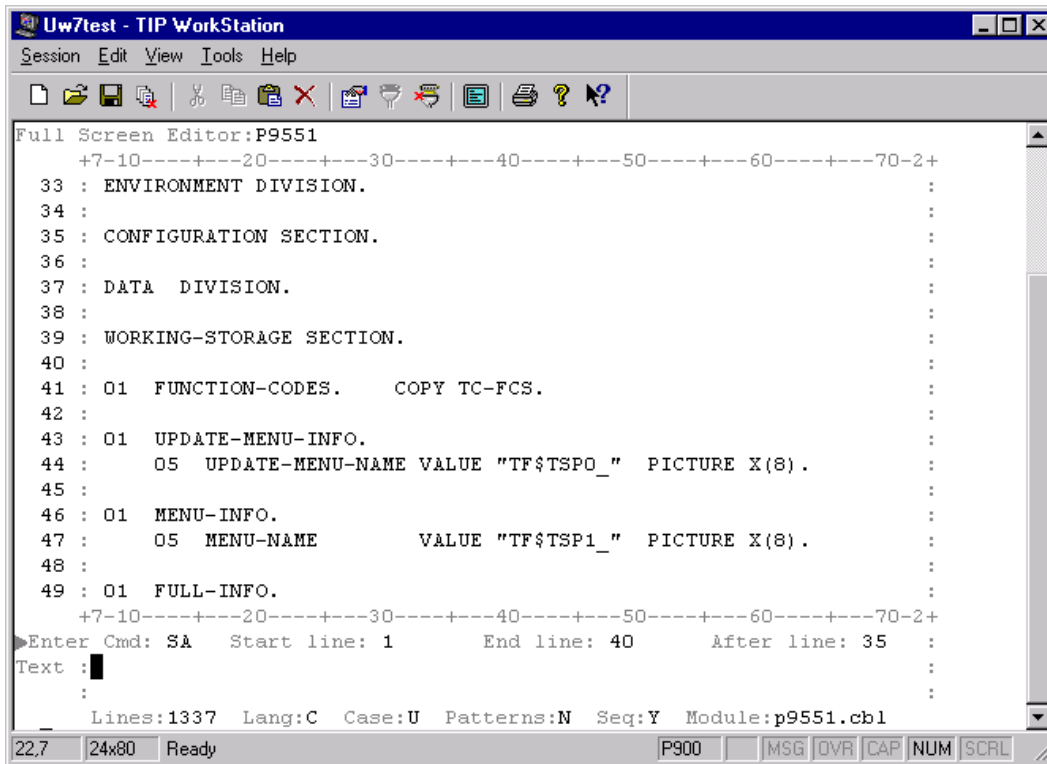
The user must specify a {startline} and {endline} and may specify in the {afterline} field the column that is to be considered the first column of the "sort key".

If {afterline} is omitted, the sort uses the entire line as the sort key.

A sort command is considered to be a "change" of the contents of the workspace (whether or not any lines actually are moved!) and therefore causes a subsequent Quit command to think that changes had been made.

Example:

This example sorts lines 1 through 40 inclusive into ascending sequence, using a key that starts in column 35.



```

Uw7test - TIP WorkStation
Session Edit View Tools Help
Full Screen Editor:P9551
+7-10-----+---20-----+---30-----+---40-----+---50-----+---60-----+---70-2+
33 : ENVIRONMENT DIVISION.                :
34 :                                       :
35 : CONFIGURATION SECTION.              :
36 :                                       :
37 : DATA DIVISION.                     :
38 :                                       :
39 : WORKING-STORAGE SECTION.            :
40 :                                       :
41 : 01 FUNCTION-CODES.      COPY TC-FCS.  :
42 :                                       :
43 : 01 UPDATE-MENU-INFO.              :
44 :   05 UPDATE-MENU-NAME VALUE "TF$TSPO_" PICTURE X(8).  :
45 :                                       :
46 : 01 MENU-INFO.                      :
47 :   05 MENU-NAME          VALUE "TF$TSP1_" PICTURE X(8).  :
48 :                                       :
49 : 01 FULL-INFO.                      :
+7-10-----+---20-----+---30-----+---40-----+---50-----+---60-----+---70-2+
Enter Cmd: SA   Start line: 1      End line: 40      After line: 35  :
Text :         :
:                                       :
:                                       :
Lines:1337 Lang:C Case:U Patterns:N Seq:Y Module:p9551.cb1
22.7 24x80 Ready P900 MSG OVR CAP NUM SCRL

```

SD - Sort Descending

The sort descending command sorts lines into descending order using a specified column as the start of the "sort key". The sort uses the standard ASCII character set as the collating sequence.

This command is identical to the [SA](#) command (described in the previous section) except that the lines are ordered in descending sequence by the specified key.

SE - Set Options

The set command allows the user to specify information about the edit workspace or to alter certain session parameters for **fse**.

The following screen format is displayed when the SE command is entered. The user may change any of the values in the unprotected fields; protected fields (suffixed by an "*" in the discussion that follows) are shown for information only.

```

TIP / ix Full Screen Editor
=====
File Type: UNIX
File: /home/ScottC/p9551.cb1

      Version: 1      Language: C      Case: U      Changed: N

Pattern Matching: N  Auto Sequence: Y  Update Stamp: STANDARD

Buffer;      Group: SCOTTC      Name: P9551

Default string:

      Page Size: 16  Page Width: 66  Left Margin: 7  Right Margin: 72

Line registers:

Line register Option: Y      Tab Size: 4

Press MSG-WAIT to return to FSE or Press XMIT to alter information
    
```

Where:

File Type*

This field indicates the type of file represented by the File name in the following field. Possible values are:

UNIX File is a UNIX file name

LIBRARY

File is a TIP/ix library and module name.

Protected field — shown for information only.

File This file is used by default by the **fse** read and write commands.

If file type is UNIX then this is a UNIX file name. This will be shown as an absolute path even though you may have specified a relative path when invoking fse.

If file type is LIBRARY then this is a TIP/ix library and module name.

When this field is changed the value of file type will be set to the file type that fse determines the new value to be. Therefore, it is possible to edit a UNIX file and then change

the default destination to be a LIBRARY. The reverse is also true. By changing the value of File the file type could also change from LIBRARY to UNIX.

Version

The current version number of the file.

Language

The language code for the contents of the edit workspace.

Specifying the language of the text in the workspace establishes default margins and dictates the number of columns that **fse** displays on the terminal.

- A** Assembler format (margins 1-72), text automatically tabbed so that the label field starts in column 1, the opcode field starts in 10, the operand field starts in column 20 and comments start in column 39.
- C** Cobol format (margins 7-72)
- D** Documentation (margins 1-72)
- P** Cobol format (margins 7-72) except that comment lines (* in column 7) are NOT translated to upper case.
- R** RPG format (margins 6-72)
- T** Assembler format (margins 1-72), text automatically tabbed so that the label field starts in column 1, the opcode field starts in 10, the operand field starts in column 16 and comments start in column 40.
- X** Upper case text (margins 1-72).
- " "** Upper case text (margins 1-72).

If the language type is changed during an **fse** session by altering the information shown by the SE command, the margins *are* reset at the same time.

Case The case processing of input text (default is Upper case unless Type="D")

- U** input forced to upper case
- L** input may be either upper or lower case

Changed*

This field contains a "Y" if the edit buffer has been changed and those changes have not yet been written to a library element.

Protected field — shown for information only.

Pattern Matching

"Y" or "N" indicating whether or not pattern matching mode is in effect.

When pattern matching mode is in effect, certain characters in strings take on special meaning. See the description of [Pattern Matching](#) .

Auto Sequence

"Y" or "N" indicating whether or not **fse** is to automatically sequence each line (according to its language code) on a write to a library.

Update Stamp

The desired type of "stamping" of updated lines. **fse** marks each line that is updated with a specific type of "update stamp".

Update stamping applies only to edit buffers that are marked as language code A, C, P or D.

All other language codes cannot make use of update stamping.

STANDARD

implies that **fse** places the current version number (3 digits) in columns 73-75 of any line that is updated and places an asterisk in column 76 of updated lines (the asterisk reveals lines that were updated during the last editing session).

spaces

same as STANDARD.

NO do not stamp updated lines in any manner

DATE

place current date in YYMMDD format in columns 73:78 of updated lines

user id

place current user id in columns 73:80 of updated lines

none of the above

fse takes the 8 characters in this field (a secret code??) and uses those characters to stamp columns 73-80.

The "stamping" process takes place as a side effect of a write command (therefore, updates that occur across more than one edit session receive the same version number).

Since the "stamping" occurs as a side effect of a write command, it follows that the stamping cannot be previewed by looking at columns 73-80 of the edit buffer.

Group*

Name of the user group that "owns" this edit buffer (workspace).

This normally is the user's first elective group.

Protected field — shown for information only.

Name*

The name of the edit workspace.

Protected field — shown for information only.

Default String*

The last string used in a search expression. This is also the default string used if one is not provided on a Find command ([FA](#), [FI](#), [FM](#), **F9**).

Note: A column specification that was part of a search string (for example: 1:20'FOO') is also retained by **fse**.

Protected field — shown for information only.

Page Size

The number of lines that constitutes a screen of text.

fse moves ahead or back this number of lines when the user uses the Page Forward/Backward commands.

Left margin*

The current column number that is the left margin.

Protected field — shown for information only.

Right margin*

The current column number that is the right margin.

Protected field — shown for information only.

Line registers*

Line numbers that have been "saved" in **fse** registers 1 through 9 (see the [#d](#) command).

Protected field — shown for information only.

Display option

Line register display option — whether or not the lines that have been "saved" in a line register (via a [#d](#) command) are to be highlighted. If this field contains a "Y", **fse** displays the register number of a saved line number rather than the actual line number (see examples in the discussion of the [#d](#) command).

The register number is also embellished with a trailing negative sign (to catch your eye) and the line number field is set to reverse video (although many terminals cannot respond to that attribute!).

Default: "Y" for all languages except RPG. The RPG screen does not have room for a trailing minus decoration;

reverse video is the best one can expect when the language is set to "R".

Tab size

Number of spaces that a tab character in a Unix file is expanded to. Default is four spaces.

SH Execute UNIX Shell

The SH command executes a UNIX shell using the value of the user's \$SHELL environment variable. If the user does not have a \$SHELL environment variable then **fse** will execute /bin/sh.

The user can execute UNIX commands from the shell.

Enter the command "exit" to terminate the shell and return to **fse**.

This command eliminates the need to exit the editor (with the "End" command) to be able to run UNIX commands.

Use the command "UX" to execute a single UNIX command from **fse** without running a shell.

Example:

```

Uw7test - TIP WorkStation
Session Edit View Tools Help
Full Screen Editor:P9551
+7-10----+---20----+---30----+---40----+---50----+---60----+---70-2+
33 : ENVIRONMENT DIVISION.           :
34 :                                 :
35 : CONFIGURATION SECTION.         :
36 :                                 :
37 : DATA DIVISION.                :
38 :                                 :
39 : WORKING-STORAGE SECTION.       :
40 :                                 :
41 : 01 FUNCTION-CODES.             COPY TC-FCS.      :
42 :                                 :
43 : 01 UPDATE-MENU-INFO.           :
44 :    05 UPDATE-MENU-NAME VALUE "TF$TSPO_" PICTURE X(8). :
45 :                                 :
46 : 01 MENU-INFO.                  :
47 :    05 MENU-NAME                VALUE "TF$TSP1_" PICTURE X(8). :
48 :                                 :
49 : 01 FULL-INFO.                  :
+7-10----+---20----+---30----+---40----+---50----+---60----+---70-2+
Enter Cmd: SH      Start line:           End line:           After line:           :
Text :                                                     :
:                                                     :
_      Lines:1337 Lang:C Case:U Patterns:N Seq:Y Module:p9551.cb1
21.30 24x80 Ready P900 MSG OVR CAP NUM SCRL
    
```

SP - Substitute and Display Changes

The SP command has the same parameters and options as the regular substitute command. The SP command however, displays the changes that it makes (in the upper area of the screen).

When the screen gets full (after each 17 lines of changes):

- pressing key F2 continues the SP command;
- entering another command terminates the SP command and proceeds to the new command;
- the user may make additional changes (manually) in the upper area of the screen format and press the XMIT key to make those additional changes — and then press the F2 key to continue the original SP command.

Additional Considerations:

Since the SP command operates interactively, it does not report the number of lines that were modified as a result of substitution.

Su - Substitute Text

The substitute command allows the user to replace one string with a different string. The starting and ending lines may be specified to limit the substitution to just that range of lines. The number of lines modified is reported when the substitute command is completed.

If an optional occurrence number is NOT specified (in the {afterline} field), the substitute command changes all occurrences that are encountered within the range specified. If neither a {startline} nor {endline} is given then substitution takes place only within the 17 lines currently displayed in the upper area of the screen.

The {afterline} field may contain the "occurrence" number (all occurrences on each inspected line is the default). To substitute the word 'RED' with 'GREEN' in lines 30 through 80, enter "S" as the command, 30 as {startline}, 80 as {endline}, "RED" in {text1}, and "GREEN" in {text2}.

The substitute command processes an individual occurrence of the specified string by specifying an occurrence number in the {afterline} field. In the above example, if only the third occurrence of RED is to be changed to GREEN, enter 3 as the {afterline}. The substitute command may be further restricted to look only in certain columns for the string. This is done by preceding the old string (in {text1}) by a column range.

To change RED into GREEN if the string RED started in columns 15 through 21, enter 15:21RED in {text1}.

Example:

This example illustrates performing a substitution on lines 1 through 100 (inclusive). If the string "RED" is found in columns 15 through 21 (inclusive) of a line it is replaced with "GREEN".

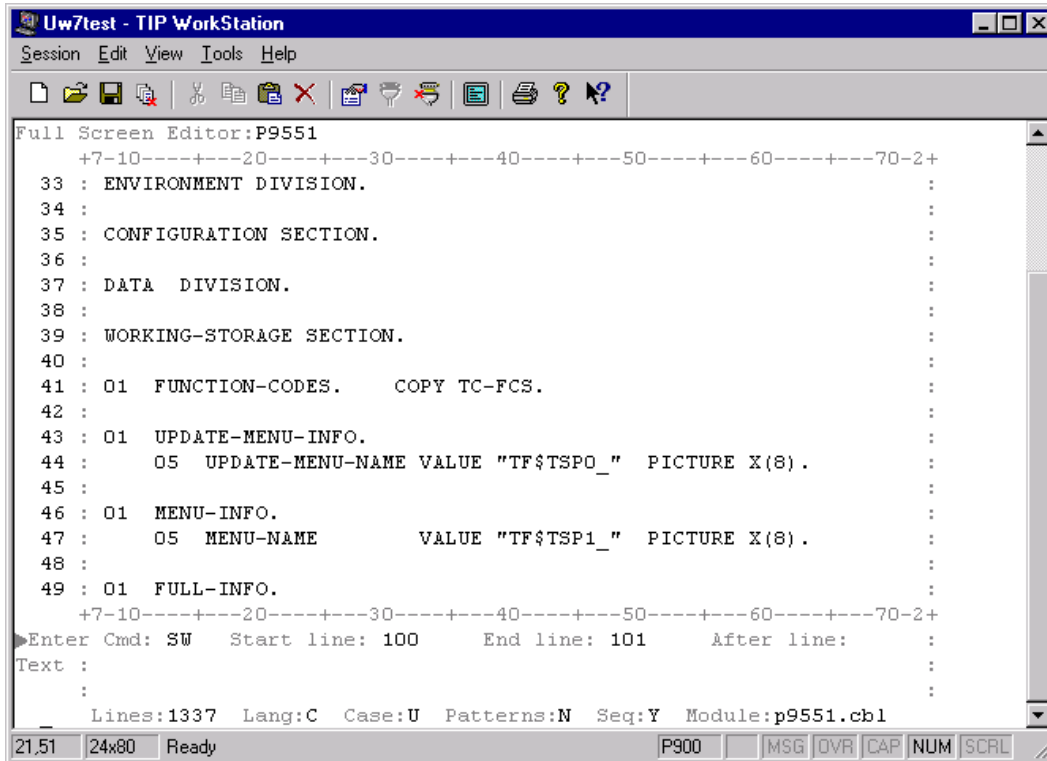
```

Uw7test - TIP WorkStation
Session Edit View Tools Help
Full Screen Editor:P9551
+7-10-----20-----30-----40-----50-----60-----70-2+
33 : ENVIRONMENT DIVISION.           :
34 :                                 :
35 : CONFIGURATION SECTION.         :
36 :                                 :
37 : DATA DIVISION.                 :
38 :                                 :
39 : WORKING-STORAGE SECTION.       :
40 :                                 :
41 : 01 FUNCTION-CODES.      COPY TC-FCS. :
42 :                                 :
43 : 01 UPDATE-MENU-INFO.           :
44 :   05 UPDATE-MENU-NAME VALUE "TF$TSPO_" PICTURE X(8). :
45 :                                 :
46 : 01 MENU-INFO.                   :
47 :   05 MENU-NAME           VALUE "TF$TSP1_" PICTURE X(8). :
48 :                                 :
49 : 01 FULL-INFO.                   :
+7-10-----20-----30-----40-----50-----60-----70-2+
Enter Cmd: SU   Start line: 1      End line: 100   After line:  :
Text :15:21RED   :
      :GREEN    :
      Lines:1337 Lang:C Case:U Patterns:N Seq:Y Module:p9551.cb1
23.12 24x80 Ready P900 MSG DVR CAP NUM SCRL
    
```

SW - Switch exchange Two Lines

The SW command exchanges two lines. The line numbers of the two lines to exchange are supplied as the {startline} and {endline}. The {startline} must be less than {endline}.

Example:



```

Uw7test - TIP WorkStation
Session Edit View Tools Help
Full Screen Editor:P9551
+7-10-----+---20-----+---30-----+---40-----+---50-----+---60-----+---70-2+
33 : ENVIRONMENT DIVISION.                                :
34 :                                                       :
35 : CONFIGURATION SECTION.                              :
36 :                                                       :
37 : DATA DIVISION.                                     :
38 :                                                       :
39 : WORKING-STORAGE SECTION.                            :
40 :                                                       :
41 : 01 FUNCTION-CODES.      COPY TC-FCS.                 :
42 :                                                       :
43 : 01 UPDATE-MENU-INFO.                                  :
44 :     05 UPDATE-MENU-NAME VALUE "TF$TSP0_" PICTURE X(8). :
45 :                                                       :
46 : 01 MENU-INFO.                                         :
47 :     05 MENU-NAME       VALUE "TF$TSP1_" PICTURE X(8). :
48 :                                                       :
49 : 01 FULL-INFO.                                         :
+7-10-----+---20-----+---30-----+---40-----+---50-----+---60-----+---70-2+
Enter Cmd: SW Start line: 100 End line: 101 After line:
Text :
:
:
Lines:1337 Lang:C Case:U Patterns:N Seq:Y Module:p9551.cb1
21.51 24x80 Ready P900 MSG OVR CAP NUM SCRL

```

Up - Update Line Range

The update command displays a range of lines in the upper area of the screen format and then allows the user to update the lines in place and (possibly) add additional lines.

If a line range is not specified (a naked U command) only line one is displayed for update.

If a line range is specified that represents more than 17 lines (the maximum number of lines that can be displayed in the upper area) **fse** displays only the first 17 lines of the range.

The user may now alter the text in the upper area of the screen (including typing additional lines following the ones shown) and press **XMIT** to replace the original lines 70-74 with the new text from the upper area of the screen.

UX - Execute a UNIX Command

The UX command "executes" the UNIX command entered in {text1}.

This command eliminates the need to exit the editor (with the "End" command) to be able to run a UNIX command.

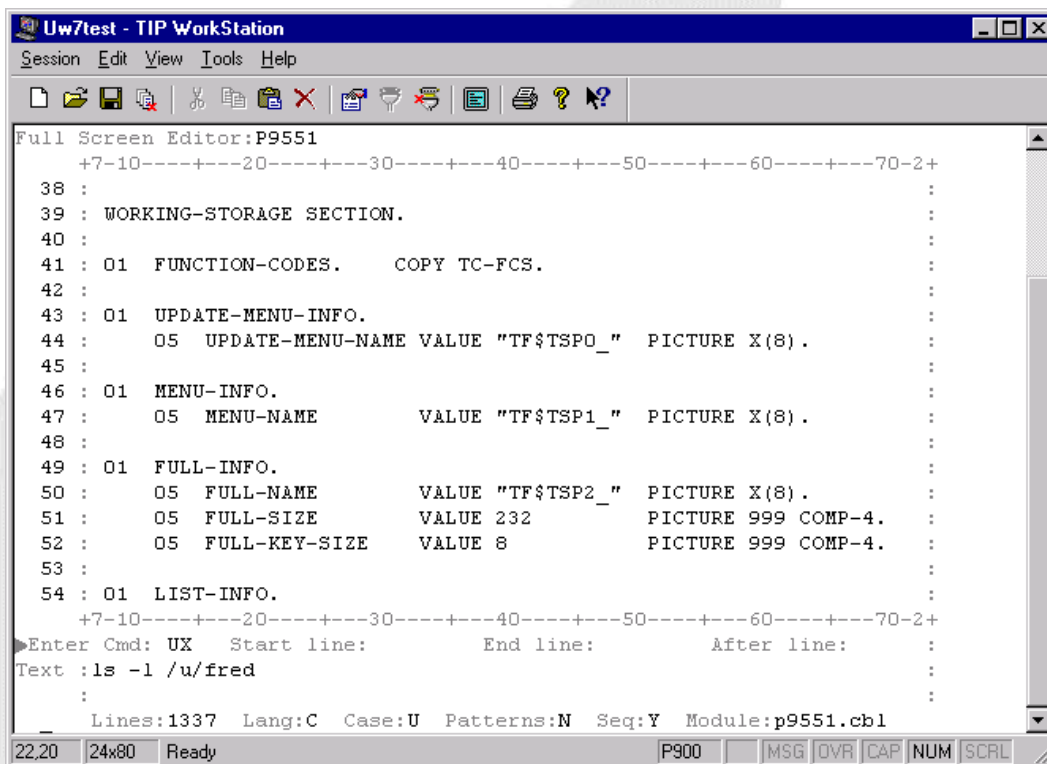
The UNIX command entered will be "piped" to the UNIX program "pg" to prevent "stdout" from interfering with MCS in managing the screen.

When interacting with "pg" press enter for the next screen of output. Press "q" and enter to exit "pg" and return to **fse**. See the UNIX *man* pages on "pg" for more information on how "pg" works.

Use the command "**SH**" to execute a UNIX shell when you want to run multiple UNIX commands.

Note: vi will not work in fse using this command.

Example:



```

Uw7test - TIP WorkStation
Session Edit View Tools Help
Full Screen Editor:P9551
+7-10----+---20----+---30----+---40----+---50----+---60----+---70-2+
38 :                                     :
39 : WORKING-STORAGE SECTION.          :
40 :                                     :
41 : 01 FUNCTION-CODES.      COPY TC-FCS. :
42 :                                     :
43 : 01 UPDATE-MENU-INFO.          :
44 :   05 UPDATE-MENU-NAME VALUE "TF$TSP0_" PICTURE X(8). :
45 :                                     :
46 : 01 MENU-INFO.                 :
47 :   05 MENU-NAME           VALUE "TF$TSP1_" PICTURE X(8). :
48 :                                     :
49 : 01 FULL-INFO.                 :
50 :   05 FULL-NAME           VALUE "TF$TSP2_" PICTURE X(8). :
51 :   05 FULL-SIZE           VALUE 232           PICTURE 999 COMP-4. :
52 :   05 FULL-KEY-SIZE       VALUE 8             PICTURE 999 COMP-4. :
53 :                                     :
54 : 01 LIST-INFO.                 :
+7-10----+---20----+---30----+---40----+---50----+---60----+---70-2+
Enter Cmd: UX   Start line:           End line:           After line:           :
Text :ls -l /u/fred                                     :
:
Lines:1337 Lang:C Case:U Patterns:N Seq:Y Module:p9551.cb1
22.20 24x80 Ready P900 MSG OVR CAP NUM SCRL

```


WE - Write Module and End

The "write with automatic end" command performs the same function as the W command and then performs an END command after the write has completed. This is equivalent to issuing a W command and then separately issuing an E command.

If the specified destination already exists then **fse** displays an informational message indicating that the file exists and requires the user to press the **F2** key to confirm that the file is to be overwritten.

If **F2** is NOT pressed (or a 60-second timeout occurs), the WE command is canceled.

See the [WR](#) command for a description of how to specify the write destination.

WN - Write No Overwrite Prompt

The "write with no overwrite prompt" command performs the same function the W command (see previous section) with the exception that **NO** prompt is issued if the specified destination already exists.

See the [WR](#) command for a description of how to specify the write destination.

WQ - Write Module and Quit

The "write with automatic quit" command performs the same function as the W command (see earlier section) and then performs a [QUit](#) command after the write has been finished. This is equivalent to issuing a W command and then separately issuing a [QUit](#) command.

If the specified destination already exists then **fse** displays an informational message indicating that the file exists and requires the user to press the **F2** key to confirm that the file is to be overwritten.

If **F2** is NOT pressed (or a 60-second timeout occurs), the WQ command is canceled.

See the [WR](#) command for a description of how to specify the write destination.

Additional Considerations:

- **fse** will not allow you to write an empty module.

WR - Write Module to TIP/ix Library, UNIX file, or Edit Buffer

The write command allows text to be written out to a library module, UNIX file, or Edit buffer from the **fse** workspace. The write destination can be supplied with the write command or entered via a prior “Set” command. If supplied with the Write command the destination is expected to be in the {text1} field in the lower area of the screen format.

A range of lines may be specified to be written; the default is to write the entire contents of the workspace to the indicated destination.

If the file name is omitted in the {text1} field **fse** uses whatever values are currently in effect (see [SE Set Options](#)) — as echoed in the lower right corner of the screen format.

If the write destination already exists then **fse** displays an informational message indicating that the file exists and requires the user to press the **F2** key to confirm that the file is to be overwritten.

If the user DOES NOT press **F2** (or spends more than 60 seconds thinking about it), the Write command is canceled.

The file information in the {text1} field is expected in one of the following formats:

```
<library> <module>  
<group> <buffer> <e|E>  
<buffer> <e|E>  
<unixfile>
```

To force “abc/def” to be interpreted as a UNIX file it is necessary to add the prefix “./”. So instead enter “./abc/def”. Otherwise **fse** will attempt to write to the module “def” in the library “ABC”. (Library names are converted to upper case.)

Example 1:

This example illustrates writing (the entire workspace) to the UNIX file “pay020” in the current directory. This is the same as specifying a destination of “./pay020”.

```

Uw7test - TIP WorkStation
Session Edit View Tools Help
Full Screen Editor:P9551
+7-10-----20-----30-----40-----50-----60-----70-2+
38 :
39 : WORKING-STORAGE SECTION.
40 :
41 : 01 FUNCTION-CODES. COPY TC-FCS.
42 :
43 : 01 UPDATE-MENU-INFO.
44 : 05 UPDATE-MENU-NAME VALUE "TF$TSP0_" PICTURE X(8).
45 :
46 : 01 MENU-INFO.
47 : 05 MENU-NAME VALUE "TF$TSP1_" PICTURE X(8).
48 :
49 : 01 FULL-INFO.
50 : 05 FULL-NAME VALUE "TF$TSP2_" PICTURE X(8).
51 : 05 FULL-SIZE VALUE 232 PICTURE 999 COMP-4.
52 : 05 FULL-KEY-SIZE VALUE 8 PICTURE 999 COMP-4.
53 :
54 : 01 LIST-INFO.
+7-10-----20-----30-----40-----50-----60-----70-2+
Enter Cmd: WR Start line: End line: After line:
Text :pay020
:
:
Lines:1337 Lang:C Case:U Patterns:N Seq:Y Module:p9551.cb1
22.13 24x80 Ready P900 MSG DVR CAP NUM SCRL
    
```

Example 2:

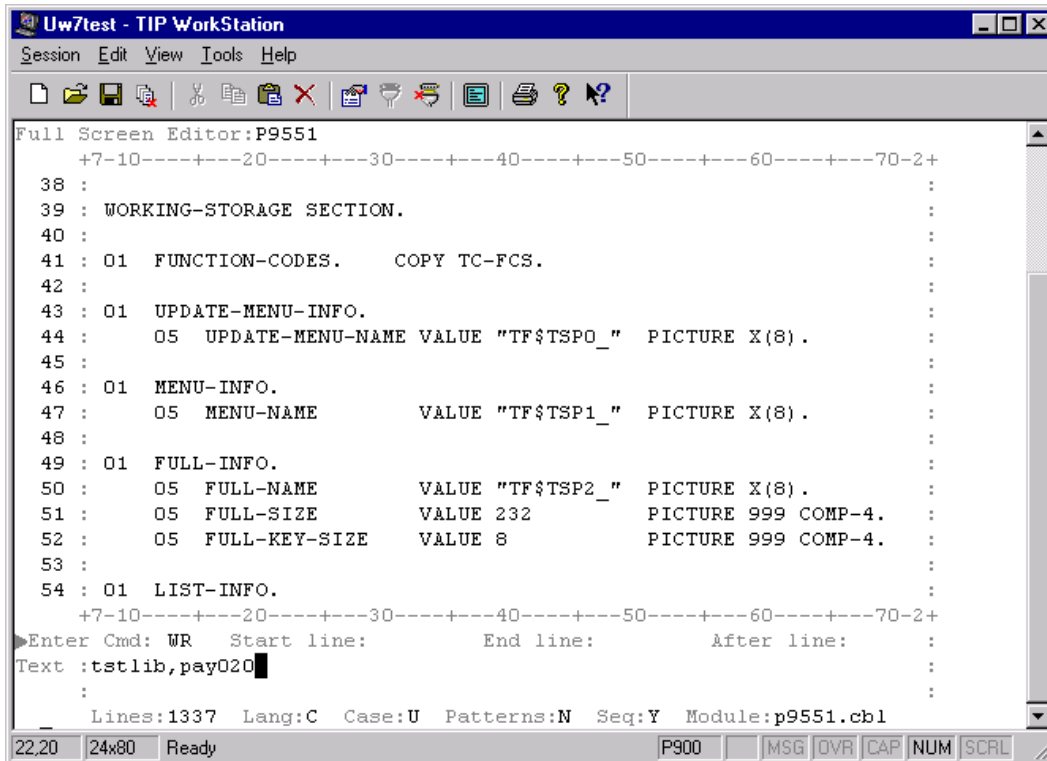
This example illustrates writing (the entire workspace) to the edit buffer "TEST" in the user's 1st elective group. If the user has no elective groups then edit buffer will be in the user's user id group.

```

Uw7test - TIP WorkStation
Session Edit View Tools Help
Full Screen Editor:P9551
+7-10-----20-----30-----40-----50-----60-----70-2+
38 :
39 : WORKING-STORAGE SECTION.
40 :
41 : 01 FUNCTION-CODES. COPY TC-FCS.
42 :
43 : 01 UPDATE-MENU-INFO.
44 : 05 UPDATE-MENU-NAME VALUE "TF$TSP0_" PICTURE X(8).
45 :
46 : 01 MENU-INFO.
47 : 05 MENU-NAME VALUE "TF$TSP1_" PICTURE X(8).
48 :
49 : 01 FULL-INFO.
50 : 05 FULL-NAME VALUE "TF$TSP2_" PICTURE X(8).
51 : 05 FULL-SIZE VALUE 232 PICTURE 999 COMP-4.
52 : 05 FULL-KEY-SIZE VALUE 8 PICTURE 999 COMP-4.
53 :
54 : 01 LIST-INFO.
+7-10-----20-----30-----40-----50-----60-----70-2+
Enter Cmd: WR Start line: End line: After line:
Text :test,E
:
:
Lines:1337 Lang:C Case:U Patterns:N Seq:Y Module:p9551.cb1
22.13 24x80 Ready P900 MSG DVR CAP NUM SCRL
    
```

Example 3:

This example illustrates writing (the entire workspace) to a TIP/ix library module. The library name is “TSTLIB” and the module (or file) in that library is “pay020”. Note that only the module name is case sensitive. The library name will be forced to upper case.



```

Uw7test - TIP WorkStation
Session Edit View Tools Help
Full Screen Editor:P9551
+7-10-----+---20-----+---30-----+---40-----+---50-----+---60-----+---70-2+
38 :
39 : WORKING-STORAGE SECTION.
40 :
41 : 01 FUNCTION-CODES. COPY TC-FCS.
42 :
43 : 01 UPDATE-MENU-INFO.
44 : 05 UPDATE-MENU-NAME VALUE "TF$TSP0_" PICTURE X(8).
45 :
46 : 01 MENU-INFO.
47 : 05 MENU-NAME VALUE "TF$TSP1_" PICTURE X(8).
48 :
49 : 01 FULL-INFO.
50 : 05 FULL-NAME VALUE "TF$TSP2_" PICTURE X(8).
51 : 05 FULL-SIZE VALUE 232 PICTURE 999 COMP-4.
52 : 05 FULL-KEY-SIZE VALUE 8 PICTURE 999 COMP-4.
53 :
54 : 01 LIST-INFO.
+7-10-----+---20-----+---30-----+---40-----+---50-----+---60-----+---70-2+
Enter Cmd: WR Start line: End line: After line:
Text :tstlib,pay020
Lines:1337 Lang:C Case:U Patterns:N Seq:Y Module:p9551.cb1
22.20 24x80 Ready P900 MSG OVR CAP NUM SCRL

```

WX - Write without Overwrite Check and End

This command combines the [WN](#) and [WE](#) commands.

WZ - Write without Overwrite Check and Quit

This command combines the [WN](#) and [WQ](#) commands.

+ - Forward Space Lines

The forward space command allows the user to go forward by a specified number of lines. Enter + as the command and enter a number in the {startline} field indicating the number of lines to move forward.

Default is one line.

Example:

This example illustrates moving the display "forward" ten lines.

```

Uw7test - TIP WorkStation
Session Edit View Tools Help
Full Screen Editor:P9551
+7-10-----+---20-----+---30-----+---40-----+---50-----+---60-----+---70-2+
33 : ENVIRONMENT DIVISION.                :
34 :                                       :
35 : CONFIGURATION SECTION.              :
36 :                                       :
37 : DATA DIVISION.                     :
38 :                                       :
39 : WORKING-STORAGE SECTION.            :
40 :                                       :
41 : 01 FUNCTION-CODES.      COPY TC-FCS.  :
42 :                                       :
43 : 01 UPDATE-MENU-INFO.                :
44 :    05 UPDATE-MENU-NAME VALUE "TF$TSP0_" PICTURE X(8).  :
45 :                                       :
46 : 01 MENU-INFO.                       :
47 :    05 MENU-NAME          VALUE "TF$TSP1_" PICTURE X(8).  :
48 :                                       :
49 : 01 FULL-INFO.                       :
+7-10-----+---20-----+---30-----+---40-----+---50-----+---60-----+---70-2+
Enter Cmd: +   Start line: 10   End line:   After line:   :
Text :                                               :
:                                               :
Lines:1337 Lang:C Case:U Patterns:N Seq:Y Module:p9551.cb1
21,32 24x80 Ready P900 MSG OVR CAP NUM SCRL
    
```

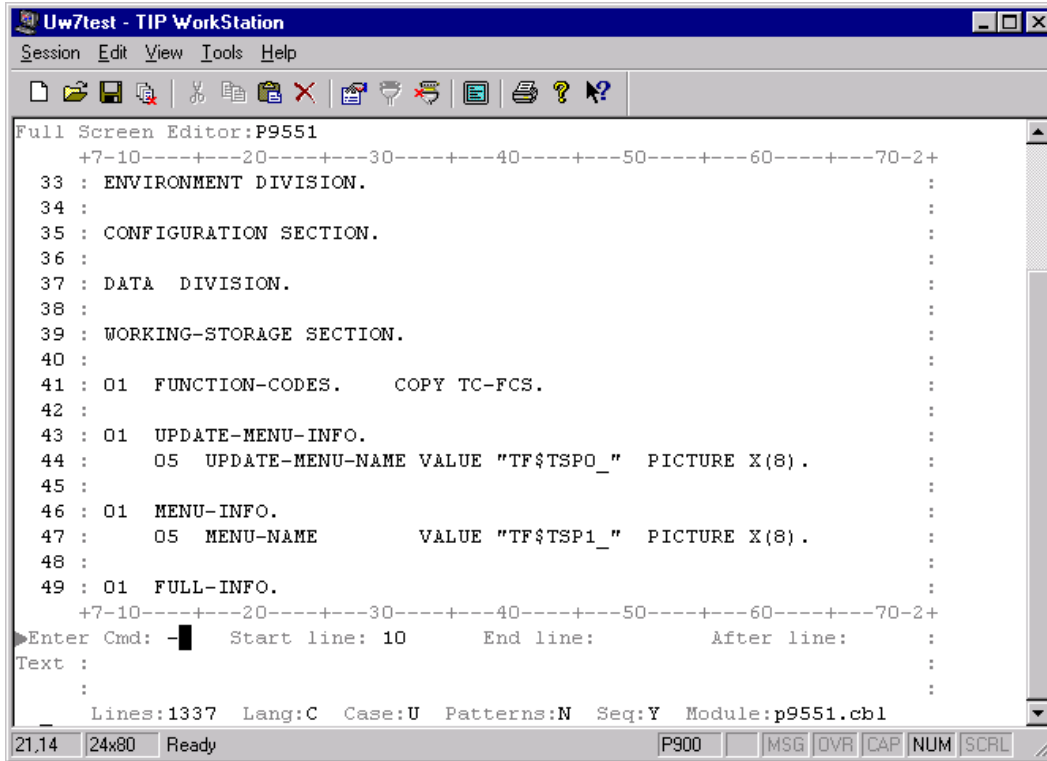
-- Backward Space Lines

The backward space command allows the user to go backward by a specified number of lines. Enter — as the command and enter a number in the {startline} field, indicating the number of lines to move backward.

Default is one line.

Example:

This example illustrates moving the display "back" ten lines.



```

Full Screen Editor:P9551
+7-10-----+---20-----+---30-----+---40-----+---50-----+---60-----+---70-2+
33 : ENVIRONMENT DIVISION.                                     :
34 :                                                           :
35 : CONFIGURATION SECTION.                                   :
36 :                                                           :
37 : DATA DIVISION.                                         :
38 :                                                           :
39 : WORKING-STORAGE SECTION.                                 :
40 :                                                           :
41 : 01 FUNCTION-CODES.      COPY TC-FCS.                     :
42 :                                                           :
43 : 01 UPDATE-MENU-INFO.                                       :
44 :     05 UPDATE-MENU-NAME VALUE "TF$TSP0_" PICTURE X(8).   :
45 :                                                           :
46 : 01 MENU-INFO.                                             :
47 :     05 MENU-NAME       VALUE "TF$TSP1_" PICTURE X(8).   :
48 :                                                           :
49 : 01 FULL-INFO.                                             :
+7-10-----+---20-----+---30-----+---40-----+---50-----+---60-----+---70-2+
Enter Cmd: █ Start line: 10 End line: After line:
Text :
:
:
Lines:1337 Lang:C Case:U Patterns:N Seq:Y Module:p9551.cb1
21.14 24x80 Ready P900 MSG DVR CAP NUM SCRL

```

= - Set Options

The = command is identical to the "[SE](#)" command previously described and is provided primarily for typing convenience.

< - Shift Display Left

The SHIFT LEFT command ("[<](#)") alters the visible portion of the record that is displayed in the upper area of the screen format. The user must specify the number of columns to shift the display in the {startline} field of the command area.

The display is shifted to the left by the number of columns specified OR until the right most column of the record is shown on the screen.

Example:

This example illustrates shifting the display to the left by 8 columns. If the workspace was declared to be COBOL language, the display would now show columns 15-80 instead of the (usual) 7-72.

```

Uw7test - TIP WorkStation
Session Edit View Tools Help
Full Screen Editor:P9551
+7-10-----+20-----+30-----+40-----+50-----+60-----+70-2+
33 : ENVIRONMENT DIVISION. :
34 : :
35 : CONFIGURATION SECTION. :
36 : :
37 : DATA DIVISION. :
38 : :
39 : WORKING-STORAGE SECTION. :
40 : :
41 : 01 FUNCTION-CODES. COPY TC-FCS. :
42 : :
43 : 01 UPDATE-MENU-INFO. :
44 : 05 UPDATE-MENU-NAME VALUE "TF$TSP0_" PICTURE X(8). :
45 : :
46 : 01 MENU-INFO. :
47 : 05 MENU-NAME VALUE "TF$TSP1_" PICTURE X(8). :
48 : :
49 : 01 FULL-INFO. :
+7-10-----+20-----+30-----+40-----+50-----+60-----+70-2+
Enter Cmd: < Start line: 8 End line: After line: :
Text : :
: :
Lines:1337 Lang:C Case:U Patterns:N Seq:Y Module:p9551.cb1
21,31 24x80 Ready P900 MSG DVR CAP NUM SCRL
    
```

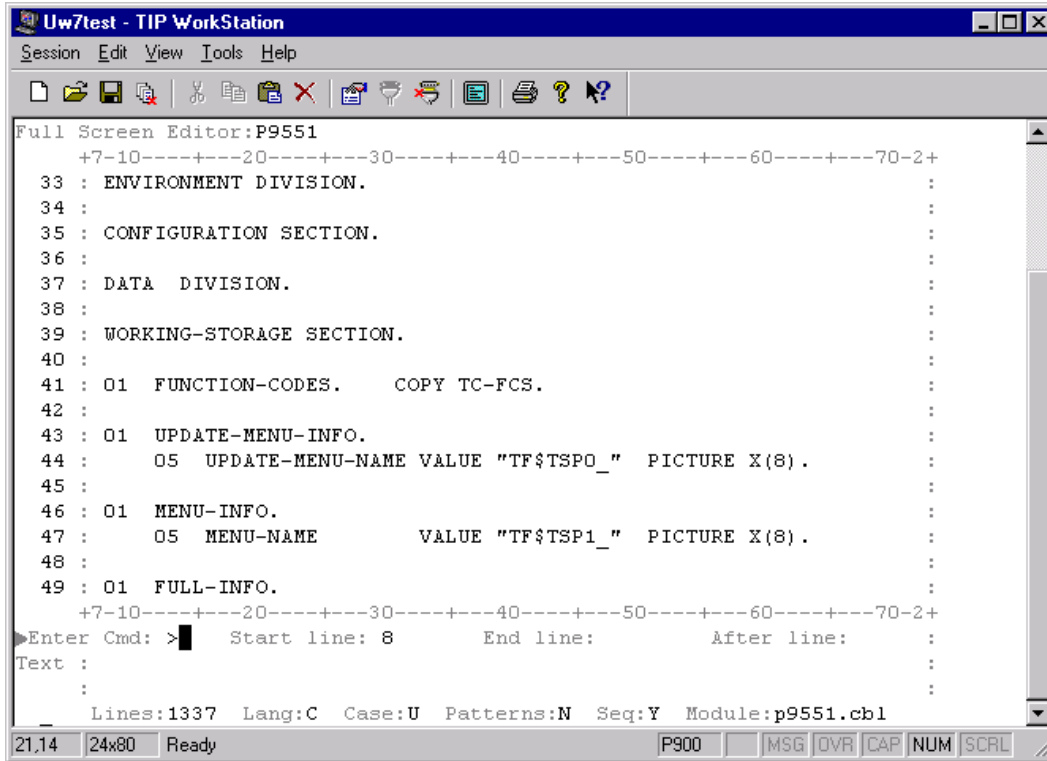
> - Shift Display Right

The SHIFT RIGHT command (" $>$ ") alters the visible portion of the record that is displayed in the upper area of the screen format. The user must specify the number of columns to shift the display in the {startline} field of the command area.

The display is shifted to the right by the number of columns specified OR until the left most column of the record is shown on the screen.

Example:

This example illustrates shifting the display to the right by 8 columns.



```

Full Screen Editor:P9551
+7-10-----+---20-----+---30-----+---40-----+---50-----+---60-----+---70-2+
33 : ENVIRONMENT DIVISION.                                     :
34 :                                                           :
35 : CONFIGURATION SECTION.                                  :
36 :                                                           :
37 : DATA DIVISION.                                         :
38 :                                                           :
39 : WORKING-STORAGE SECTION.                                 :
40 :                                                           :
41 : 01 FUNCTION-CODES.      COPY TC-FCS.                    :
42 :                                                           :
43 : 01 UPDATE-MENU-INFO.                                       :
44 :   05 UPDATE-MENU-NAME VALUE "TF$TSPO_" PICTURE X(8).    :
45 :                                                           :
46 : 01 MENU-INFO.                                             :
47 :   05 MENU-NAME          VALUE "TF$TSP1_" PICTURE X(8).  :
48 :                                                           :
49 : 01 FULL-INFO.                                             :
+7-10-----+---20-----+---30-----+---40-----+---50-----+---60-----+---70-2+
Enter Cmd: >      Start line: 8      End line:      After line:      :
Text :                                                    :
:                                                           :
:                                                           :
Lines:1337 Lang:C Case:U Patterns:N Seq:Y Module:p9551.cb1
21.14 24x80 Ready P900 MSG DVR CAP NUM SCRL

```

Call fse Recursively

The CALL **fse** command ("^") allows the user to invoke another copy or instance of **fse** to start another editing session without having to close the current editing session. The current editing session (or instance of **fse**) is saved by the system and is reactivated when the higher level session is terminated. This command is especially useful to permit the user to temporarily preempt the current editing session and invoke **fse** to edit something else and then return.

If desired, the **fse** command line parameters to read a file may be placed in the first text line area (see example that follows). If no parameters are supplied, **fse** is invoked without parameters and **fse** reacts by displaying the initial entry screen.

Example:

This example illustrates invoking another instance of **fse** to edit "filename" using the group and buffer specified. The "-R" indicates that **fse** is to be invoked in read-only mode.


```

Full Screen Editor:P9551
+7-10-----+---20-----+---30-----+---40-----+---50-----+---60-----+---70-2+
33 : ENVIRONMENT DIVISION.                                :
34 :                                                         :
35 : CONFIGURATION SECTION.                               :
36 :                                                         :
37 : DATA DIVISION.                                       :
38 :                                                         :
39 : WORKING-STORAGE SECTION.                             :
40 :                                                         :
41 : 01 FUNCTION-CODES.      COPY TC-FCS.                 :
42 :                                                         :
43 : 01 UPDATE-MENU-INFO.                                     :
44 :     05 UPDATE-MENU-NAME VALUE "TF$TSP0_" PICTURE X(8). :
45 :                                                         :
46 : 01 MENU-INFO.                                           :
47 :     05 MENU-NAME      VALUE "TF$TSP1_" PICTURE X(8).   :
48 :                                                         :
49 : 01 FULL-INFO.                                           :
+7-10-----+---20-----+---30-----+---40-----+---50-----+---60-----+---70-2+
Enter Cmd: ^      Start line:      End line:      After line:      :
Text :-R filename group buffer      :
:                                     :
:                                     :
Lines:1337 Lang:C Case:U Patterns:N Seq:Y Module:p9551.cb1
22.31 24x80 Ready P900 MSG DVR CAP NUM SCRL
    
```

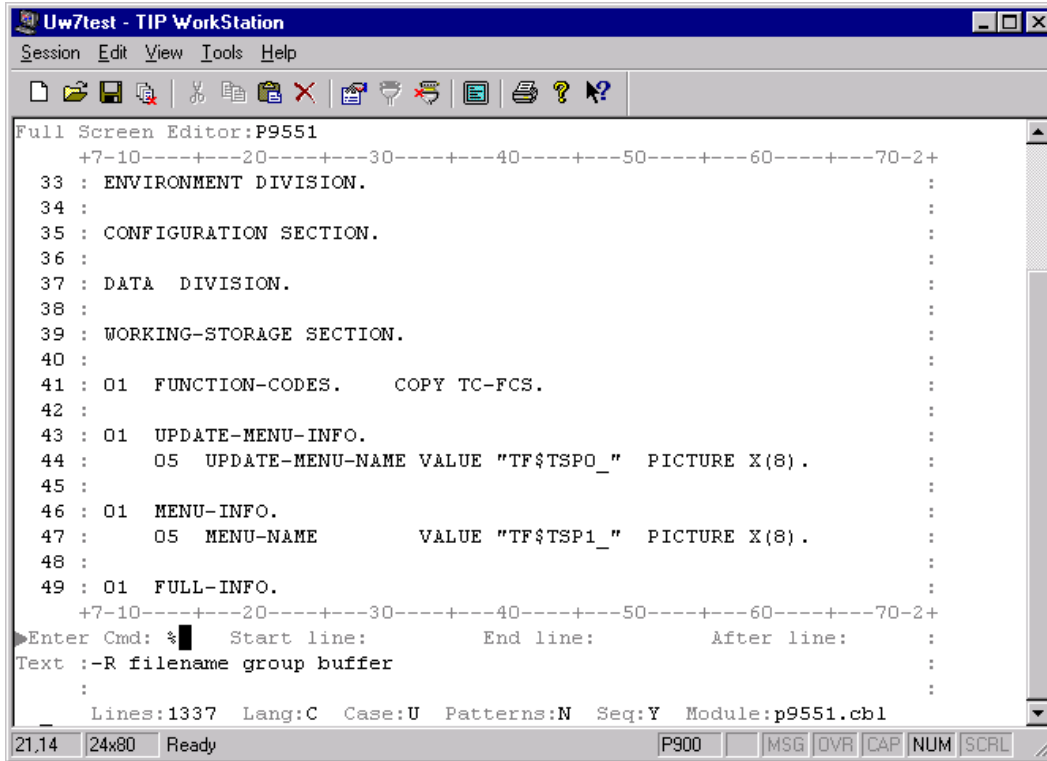
% - Call fse with No Return - Not Implemented

The CALL fse command ("%") allows you to invoke another copy or instance of fse to start another editing session without having to close the current editing session and then terminate the fse session. The current editing session (or instance of fse) is saved by the system and is not reactivated when the higher level session is terminated. This useful command allows you to permanently preempt the current editing session and invoke fse to edit something else.

If desired, you may place the fse command line parameters in the first text line area (see example that follows). If no parameters are supplied, fse is invoked without parameters and fse reacts by displaying the initial entry screen.

Example:

This example illustrates invoking another instance of **fse** to edit "filename" using the group and buffer specified. The "-R" indicates that **fse** is to be invoked in read-only mode.



```

Full Screen Editor:P9551
+7-10-----+---20-----+---30-----+---40-----+---50-----+---60-----+---70-2+
33 : ENVIRONMENT DIVISION.                                     :
34 :                                                           :
35 : CONFIGURATION SECTION.                                   :
36 :                                                           :
37 : DATA DIVISION.                                         :
38 :                                                           :
39 : WORKING-STORAGE SECTION.                                 :
40 :                                                           :
41 : 01 FUNCTION-CODES.      COPY TC-FCS.                     :
42 :                                                           :
43 : 01 UPDATE-MENU-INFO.                                       :
44 :     05 UPDATE-MENU-NAME VALUE "TF$TSP0_" PICTURE X(8).   :
45 :                                                           :
46 : 01 MENU-INFO.                                             :
47 :     05 MENU-NAME       VALUE "TF$TSP1_" PICTURE X(8).   :
48 :                                                           :
49 : 01 FULL-INFO.                                             :
+7-10-----+---20-----+---30-----+---40-----+---50-----+---60-----+---70-2+
Enter Cmd: %      Start line:      End line:      After line:      :
Text :-R filename group buffer    :
:                                  :
:                                  :
Lines:1337 Lang:C Case:U Patterns:N Seq:Y Module:p9551.cb1
21.14 24x80 Ready P900 MSG OVR CAP NUM SCRL

```

#d - Saving Line Numbers

fse maintains nine "registers" (1-9) that may be assigned line numbers. These registers can then be used in place of absolute line numbers in other commands.

The "#d" command is used to save a line number in register "d".

For example, line number 107 can be stored in register number 5 by issuing the "#5" command with "107" in the {startline} field of the command area. Thereafter, the user can reference that line by specifying line number "-5". For example LI(-5) is interpreted as "list lines beginning with the line number stored in register 5". If {startline} is omitted, the line number that is currently the first line in the upper portion of the display is assumed.

fse updates the actual line number in the register to "track" the line. If the location of the line changes due to the occurrence of a line delete or line add in front of the line, the register is updated to reflect the line's new position.

If a line has been noted in a register and the line itself is deleted, **fse** clears the register reference for that line.

Examples:

Depending on an option that is present on the option page, the user may or may not observe that saved line numbers are highlighted in the upper display.

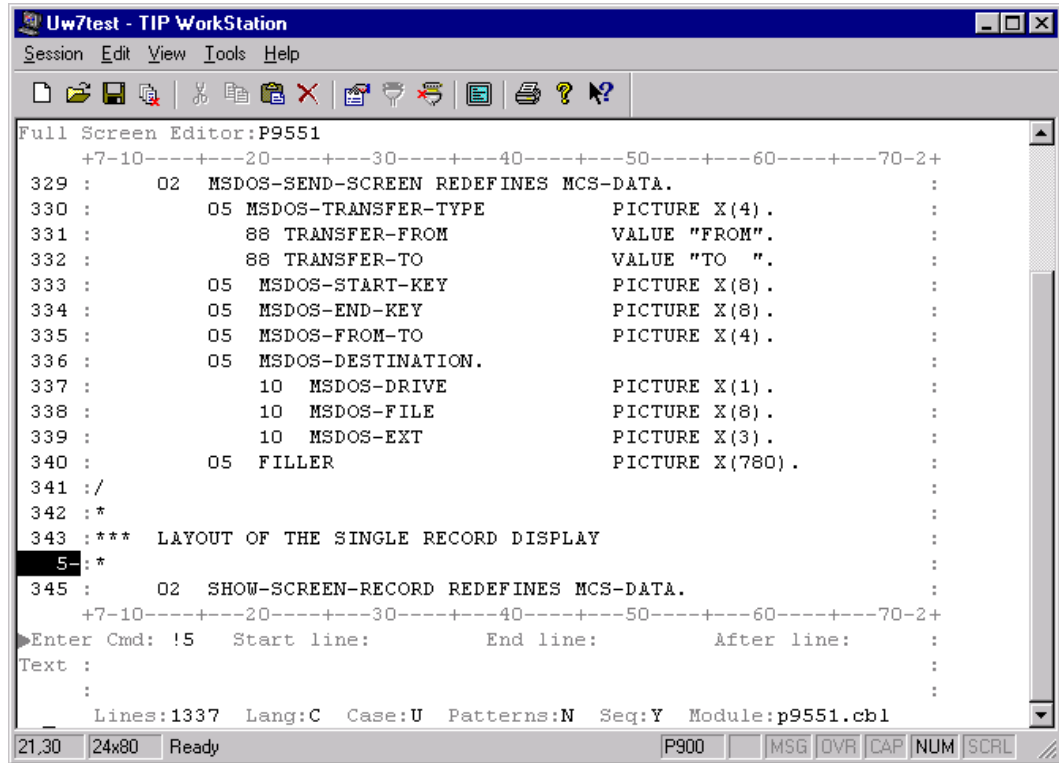
```

Uw7test - TIP WorkStation
Session Edit View Tools Help
Full Screen Editor:P9551
+7-10-----20-----30-----40-----50-----60-----70-2+
33 : ENVIRONMENT DIVISION.                :
34 :                                       :
35 : CONFIGURATION SECTION.                :
36 :                                       :
37 : DATA DIVISION.                        :
38 :                                       :
39 : WORKING-STORAGE SECTION.              :
40 :                                       :
41 : 01 FUNCTION-CODES.      COPY TC-FCS.   :
42 :                                       :
43 : 01 UPDATE-MENU-INFO.                  :
44 : 05 UPDATE-MENU-NAME VALUE "TF$TSP0_"  PICTURE X(8) . :
45 :                                       :
46 : 01 MENU-INFO.                          :
47 : 05 MENU-NAME          VALUE "TF$TSP1_"  PICTURE X(8) . :
48 :                                       :
49 : 01 FULL-INFO.                          :
+7-10-----20-----30-----40-----50-----60-----70-2+
Enter Cmd: #5  Start line: 107  End line:      After line:  :
Text :
:
Lines:1337 Lang:C Case:U Patterns:N Seq:Y Module:p9551.cb1
22.7 24x80 Ready P900 MSG OVR CAP NUM SCRL
    
```

When the line register display option is on (set to "Y"), **fse** displays the register number instead of the actual line number. The display has the "reverse video" attribute on (for terminals with that capability) and includes a trailing minus sign (for terminals that do not support reverse video).

The above example shows line number 344 when it has been noted in line register #5 (many terminals cannot display fields in reverse video — the trailing minus sign is more obvious).

Example:



Function Key Usage

The Full Screen Editor recognizes certain function keys as special commands. Invalid function keys result in an error message on the screen.

Key	Description
XMIT	<p>The interpretation of the XMIT key depends on the location of the cursor:</p> <p>If the cursor is in the upper area of the display, fse reads the text in the upper area of the display and alters the corresponding text in the workspace (if the text was altered).</p> <p>If the cursor is in the lower area of the display fse attempts to process the command. If there is no command, fse assumes the default command (F2 — display next screen full).</p>
MSG WAIT	<p>Pressing MSG WAIT is equivalent to entering an "End" command.</p> <p>fse terminates and the workspace is retained.</p>
F1	<p>Function key 1 causes fse to resend the last screen that was output. This may be important if the screen</p>

display was altered unintentionally or by the receipt of an unsolicited message.

- F2 Function key 2 is the "Forward Page" key. When this function key is pressed fse displays the next screen of source lines.

The number of lines that a screen of text implies is a user-definable number of lines (see the description of page size in SE Set Options).

- F3 Function key 3 is the "Backward Page" key. When this function key is pressed fse displays the previous screen of source lines.

The number of lines that a screen of text implies is a user-definable number of lines (see the description of page size in SE Set Options).

- F4 Function key 4 signals the fse program that the user wishes to abort the edit session (equivalent to the QUIT command).

If changes have been made since the last time the contents of the edit workspace was written, fse displays a warning message and requires the user to press the F2 key to confirm that the QUIT is to be performed.

- F5 Function key 5 causes fse to insert one blank line ahead of the line where the cursor is resting. The cursor must be placed on the appropriate line in the upper area of the screen before pressing F5.

The cursor remains at the location where the user pressed F5. If F5 is pressed immediately following the use of F6 (see next discussion), fse inserts the line that was deleted by F6 (a bizarre but deliberate reinstatement of a deleted line).

- F6 Function key 6 causes fse to delete the line where the cursor is resting. The cursor must be placed on the appropriate line in the upper area of the screen before pressing F6.

The cursor remains at the location where the user pressed F6. If F6 is pressed unintentionally, press F5 to reinstate the erroneously deleted line (you have one chance to do this — don't waste it!).

- F7 Function key 7 causes fse to "split" the line where the cursor is resting at the point where the cursor is located. The original line is turned into two lines.

The first line contains the characters up to (but not including) the character under the cursor.

The second of the two lines contains the remaining

characters of the first line.

After the line is "split" the cursor is restored to the original split point.

- F8 Function key 8 causes fse to "join" the line containing the cursor and the line following at the point where the cursor is positioned. This function is the inverse of F7.

After the line is "joined" the cursor is restored to the original cursor position.

- F9 Function key 9 is considered as a request to reissue the last find command (FI, FM, F-).

This means that another search begins for the string specified in the last search command (if the previous string included a column specification, that too is remembered).

The direction of the previous search command is remembered and honoured.

- F10 - Function keys 10 through 22 are user-definable by
F22 session. The F# command (described in a previous section) may be used to assign a specific command (with or without parameters) to a function key in this range.

That function key may then be pressed to invoke the command that was "soft coded" as that key

Pattern Matching

When specifying a search string, the desired string is often not explicitly known. Instead, the user is aware of the layout or structure of the string that is desired.

For example, you may wish to search for a particular type of field name that you know is constructed as follows: an alpha character followed by two digits followed by two more alpha characters (such as "A12DE").

To search for that type of string explicitly is not possible - there exists a large number of permutations of characters and digits that satisfy the stated rule.

To address this issue, **fse** allows the user to enter "pattern matching mode". In this mode, it is possible to specify search strings in an implicit manner by specifying a desired pattern of characters.

When pattern matching mode is enabled, several characters have reserved meaning and are not interpreted literally by **fse**. These characters are referred to as "meta characters".

. (period)

The period character may be used to "match" any character (essentially a wild card character).

% (percent)

The percent character may be used to "match" any alpha character ("a" through "z" or "A" through "A" through "Z").

(pound)

The pound sign character may be used to "match" any digit (0 through 9).

*** (star)**

The star character (asterisk) indicates that the match pattern is to allow **0 or more repetitions** of the character which preceded the star

For example: `#*` would match 0 or more digits in a row.

" (quote)

The double quote character indicates that the character which follows is to be taken literally and not interpreted as a reserved character in this instance.

This mechanism allows you to use the period, percent, pound, star (asterisk), double quote, left square bracket or right square bracket characters literally in a search string.

For example, to look for the string "A#B" when pattern matching is turned on, specify `A"#B` as the search string.

[]

Square brackets are used to enclose "tag expressions". More on that subject follows.

With these reserved characters, some reasonably sophisticated search patterns can be specified.

ExampleS:

`%##%%`

This is the example discussed at the start of this section. An alphabetic character, followed by 2 digits, followed by two alphabetic characters.

MATCHES: A12DB Z00XX R12BW

(.*)

Any number of arbitrary characters (including none at all!) enclosed in parentheses.

Note that this pattern literally implies: a (character, followed by zero or more occurrences of any character, followed by a) character.

MATCHES: (12), (more or less), (text1), (xxxxxxx) and ()

##/##/##

Date format.

MATCHES: 86/01/05 and 05/01/87

##"## Two digits followed by a # character.

MATCHES: 37# and 12#

#*"## Any number of digits (including no digits) followed by a # character.

MATCHES: 37#, 12#, 1234567#, 000034# and #

An advanced feature of pattern matching mode is the ability to segregate a portion of a search string and assign a "tag" or a "name" to whatever ultimately matches that portion of the search string.

Pieces of the string that are tagged may be used essentially as variables – as shown in the following example.

[A left square bracket marks the start of a "tagged" expression.

The pattern which follows is considered separately from the remainder of the search string.

]a A right square bracket followed immediately by an alphabetic character marks the end of a "tagged" expression.

The portion of the string that matches the "tagged" expression is assigned the one-character "tag" denoted by the alphabetic character which followed the "]" character.

& An ampersand character (&) can be used in the replacement text to recall the characters that were matched by the tagged expression. The ampersand means, literally, "what was matched".

Example:

Consider the following task to perform: assume there are a number of data fields defined in a COBOL record layout and that the definition of these fields has been copied from the DATA DIVISION to the PROCEDURE DIVISION:

```

Uw7test - TIP WorkStation
Session Edit View Tools Help
Full Screen Editor:NEW
+7-10-----20-----30-----40-----50-----60-----70-2+
225 :      10 CM-NUMBER                PICTURE X(8) .      :
226 :      10 CM-STATUS                PICTURE X(8) .      :
227 :      10 CM-COMPANY                PICTURE X(8) .      :
228 :      10 CM-ADDRESS-1              PICTURE X(8) .      :
229 :      10 CM-ADDRESS-2              PICTURE X(8) .      :
230 :      10 CM-ADDRESS-3              PICTURE X(8) .      :
231 :      10 CM-POSTAL                 PICTURE X(8) .      :
232 :      10 CM-TELEPHONE               PICTURE X(8) .      :
233 :      10 CM-TELEX                   PICTURE X(8) .      :
234 :      10 CM-PO-NUMBER                PICTURE X(8) .      :
235 :      10 CM-ATTN                     PICTURE X(8) .      :
236 :      10 CM-DP-MGR                  PICTURE X(8) .      :
237 :      10 CM-MACHINE                  PICTURE X(8) .      :
238 :      10 CM-MEMORY                   PICTURE X(8) .      :
239 :      10 CM-DISK                     PICTURE X(8) .      :
240 :      10 CM-TAPE                      PICTURE X(8) .      :
241 :      10 CM-NO-TERMINALS              PICTURE X(8) .      :
+7-10-----20-----30-----40-----50-----60-----70-2+
Enter Cmd: █      Start line:      End line:      After line:      :
Text :      :
:      :
:      :
Lines: 283 Lang:C Case:U Patterns:N Seq:Y Module:new.cbl
21.13 24x80 Ready P900 MSG DVR CAP NUM SCRL

```

The intention is to smash all of the data field descriptions to statements that initialize the fields to spaces.

One (brute force) approach would be to move the cursor to the upper part of the **fse** display and personally remove the "PIC" part of each statement (this is an instance where a destructive space bar is handy!) and then use a substitute command to change the string "10..." to the string "MOVE SPACES TO".

An alternative approach (which assumes that the user has first used the SE command to turn pattern matching ON) involves only a single substitute command:

```

Uw7test - TIP WorkStation
Session Edit View Tools Help
Full Screen Editor:NEW
+7-10-----+---20-----+---30-----+---40-----+---50-----+---60-----+---70-2+
225 :          10 CM-NUMBER                PICTURE X(8) .          :
226 :          10 CM-STATUS                PICTURE X(8) .          :
227 :          10 CM-COMPANY                PICTURE X(8) .          :
228 :          10 CM-ADDRESS-1             PICTURE X(8) .          :
229 :          10 CM-ADDRESS-2             PICTURE X(8) .          :
230 :          10 CM-ADDRESS-3             PICTURE X(8) .          :
231 :          10 CM-POSTAL                 PICTURE X(8) .          :
232 :          10 CM-TELEPHONE              PICTURE X(8) .          :
233 :          10 CM-TELEX                   PICTURE X(8) .          :
234 :          10 CM-PO-NUMBER               PICTURE X(8) .          :
235 :          10 CM-ATTN                    PICTURE X(8) .          :
236 :          10 CM-DP-MGR                  PICTURE X(8) .          :
237 :          10 CM-MACHINE                  PICTURE X(8) .          :
238 :          10 CM-MEMORY                   PICTURE X(8) .          :
239 :          10 CM-DISK                     PICTURE X(8) .          :
240 :          10 CM-TAPE                     PICTURE X(8) .          :
241 :          10 CM-NO-TERMINALS             PICTURE X(8) .          :
+7-10-----+---20-----+---30-----+---40-----+---50-----+---60-----+---70-2+
Enter Cmd: SU      Start line:           End line:           After line:           :
Text : '## *[%.*]Q .*' .                :
      :MOVE SPACES TO Q.                  :
      Lines: 283 Lang:C Case:U Patterns:Y Seq:Y Module:/home/ScottC/new.cb
23.24 24x80 Ready P900 MSG DVR CAP NUM SCRL
  
```

After the substitute command shown above, the lines appear as follows:

```

Uw7test - TIP WorkStation
Session Edit View Tools Help
Full Screen Editor:NEW 283 lines written to /home/ScottC/new.cb1
+7-10-----+---20-----+---30-----+---40-----+---50-----+---60-----+---70-2+
225 :          MOVE SPACES TO CM-NUMBER.    :
226 :          MOVE SPACES TO CM-STATUS.    :
227 :          MOVE SPACES TO CM-COMPANY.    :
228 :          MOVE SPACES TO CM-ADDRESS-1.  :
229 :          MOVE SPACES TO CM-ADDRESS-2.  :
230 :          MOVE SPACES TO CM-ADDRESS-3.  :
231 :          MOVE SPACES TO CM-POSTAL.     :
232 :          MOVE SPACES TO CM-TELEPHONE.  :
233 :          MOVE SPACES TO CM-TELEX.      :
234 :          MOVE SPACES TO CM-PO-NUMBER.  :
235 :          MOVE SPACES TO CM-ATTN.       :
236 :          MOVE SPACES TO CM-DP-MGR.     :
237 :          MOVE SPACES TO CM-MACHINE.    :
238 :          MOVE SPACES TO CM-MEMORY.     :
239 :          MOVE SPACES TO CM-DISK.       :
240 :          MOVE SPACES TO CM-TAPE.       :
241 :          MOVE SPACES TO CM-NO-TERMINALS. :
+7-10-----+---20-----+---30-----+---40-----+---50-----+---60-----+---70-2+
Enter Cmd: █      Start line:           End line:           After line:           :
Text :           :
      :           :
      Lines: 283 Lang:C Case:U Patterns:Y Seq:Y Module:/home/ScottC/new.cb
21.13 24x80 Ready P900 MSG DVR CAP NUM SCRL
  
```

In the above example, note the important appearance of the space after the [%.*]Q expression. The space character is required (in this case) to

inform **fse** how to determine where the tagged expression ends (in this case the tagged expression is the field name).

fxp - Printing Postscript Output

The **fxp** utility is a file printing program that can direct output to:

- your display
- a Hewlett-Packard Laserjet printer (or emulation)
- a Postscript printer (having first pre-processed the output)
- the Unix print spooler

Syntax:

```
fxp [ -options ] file 1... file n
```

Where:

No options

If you enter **fxp** with no options TIP/ix will display the program's help (syntax and options.)

• Output

- p Specify this option to pipe output of **fxp** into the Unix print spooler LP (check the TIPPRINT environment variable.)
- O Route output of **fxp** to standard output. Useful for piping output to some other program.

Example:

```
fxp -O | rsh sys2 lp -o raw -o nobanner
```

- X Print to your AUX printer (check the TIPPRINTAUX environment variable.)

• Postscript Printer Orientation

- a Print in (Adobe) Postscript Portrait mode.
- a=file Print in (Adobe) Postscript Portrait mode using "file" as the Postscript header file. **fxp** expects this file to be in the *\$TIPROOT/include* directory with a suffix of ".ps".
- l Print in Postscript Landscape mode using a point size of 10.
- l=file Print in Postscript Landscape mode using "file" as the Postscript header file. **fxp** expects this file to be in the *\$TIPROOT/include* directory with a suffix of ".ps".

- Z** Print in Postscript Landscape using a point size of 8 (160 chars/line for 8.5 x 11 paper).
- k** Print in Postscript Landscape mode using two columns.
This is the default print format.
- k=file** Print in Postscript Landscape mode using two columns and "file" as the Postscript header file. **fxp** expects this file to be in the *\$TIPROOT/include* directory with a suffix of ".ps".
- s** Print in Postscript Landscape mode using three columns and a small (3.7) point size.
-s=file Print in Postscript Landscape mode using three columns and "file" as the Postscript header file. **fxp** expects this file to be in the *\$TIPROOT/include* directory with a suffix of ".ps".

- **HP Printer**

- h** A Hewlett-Packard Laserjet printer. With this option you can also specify a format option to print comments in bold.

- **Formatting**

- b** Print program comments in bold. The default program type is **C**. To bold COBOL program comments you must specify the "-C" option in addition to "-b".

This option can only be used with a Postscript printer (see list above) or HP Laserjet ("-h").

- B** Turn bolding of comments off.

- C** Print a COBOL source file.

A "/" in column seven will force a page eject during printing. To print comments in bold you must also specify "-b" option.

- n** Print output with line numbering.

- N** Turn line numbering off.

- nn** Print while inserting tab characters where **nn** is the number of tabset characters from **2** to **8** (inclusive.)

The default value is 48 characters.

- **Input**

file 1... One, or a list of files you wish to print.

Examples:

This example will result in fxp printing the file “prog1.cbl” (from the current directory) treating it as a COBOL program (slash in column 7 will force a page eject) and printing the comments (asterisk in column 7) in bold. The output will be routed to standard output, which in this case has been piped to rsh. In this case rsh connects to the host “sys2” and executes “lp” on that system.

```
(/u/joe)342 $fxp -O -C -b prog1.cbl | rsh
sys2 lp -o raw -o nobanner
```

This example will result in fxp printing the file “myfile” (from the current directory) in portrait mode with no line numbering using the Postscript header file *\$TIPROOT/include/myport.ps* and piping the output of fxp to the UNIX print spooler LP.

```
(/u/joe)342 $fxp -p -N -a=myport myfile
```

The TIPPRINT environment variable must be set to PS in this case for the Postscript header file to be applied. Otherwise the Postscript header file will be ignored.

Additional Considerations:

- The filename(s) must be present and correct or fxp will issue an error message similar to the following:

```
ARC print utility. Lines/page: 66, Chars/line 89
roman.doc: No such file or directory
0 sheets of paper used
```
- You may also interweave options (such as line numbering) with the specified file names you wish printed.
- By default, the fxp utility passes an option of -o nobanner to the UNIX spooler lp. If you want different options, set the LPFLAGS environment variable to the desired options. To suppress the default (-o nobanner), set LPFLAGS to " " (nothing).

genmain - Write Unix main Program

The **genmain** utility writes a small C language **main** program that calls the real (COBOL) transaction program. Every program on UNIX must have a function called **main**. All transaction programs have a LINKAGE SECTION and are written to be subroutines of the transaction monitor (TIP/ix in this case).

genmain is normally used in the **makefile** as part of the compilation procedure for online COBOL programs (see references to "makefile".) **genmain** is not normally invoked at the command line!

Refer to the TIP/ix Installation & Operations manual for details.

groups - Change Users Active Groups

The **groups** transaction may be used to alter the active groups for a TIP/ix session. This change only applies to the current TIP/ix session and has no effect on other TIP/ix sessions for the user. The change to the active groups is temporary the alteration lasts until it is changed again or until the end of the TIP/ix session (user is logged off).

The "standard order of search" that TIP/ix employs to resolve transaction, file, and queue references is:

- user id group (group name equal to the user id)
- active elective groups (from the list of eligible elective groups)
- the universal group "TIP\$Y\$"

The **groups** transaction allows the user to alter the active elective groups for the current TIP/ix session. The user is able to change the active groups to any group to which the user has been granted access; either as an elective group or as a member of the user's logon set or group set. See **smuser** for details on how to assign a user access to groups.

A user can have up to 16 groups active at any time. However, the **groups** transaction can only be used to alter the first 8 active groups.

Syntax:

```
groups [ ,q] g1,g2,g3,g4,g5,g6,g7,g8
groups [ ,q] *
groups ,acct account
```

Where:

.q Command line option that may be used to suppress the screen format that is usually displayed (see examples which follow).

This is the "quiet" option. Most often used when the groups transaction is called by another transaction program.

,acct account

Start using specified account. If the specified account is not valid, or you are not authorized to use it, your default account number will be used instead.

If the first parameter is an asterisk (*), the groups program resets the user's elective groups to the groups that are set at logon time. This depends on the elective groups and

logon set defined in the user id definition (see smuser). If this parameter is specified then all other parameters are ignored. This provides the only possibility for the groups program to alter groups 9 through 16 of the user's active group list.

g1...g8 Up to eight positional parameters representing desired alterations to the elective groups to which the user belongs. The eight parameters correspond to the user's first eight elective groups.

If a parameter is omitted, the implication is taken that the corresponding elective group is not to be changed.

If a parameter contains exactly the character string NONE, the implication is that the corresponding elective group is to be cleared to spaces.

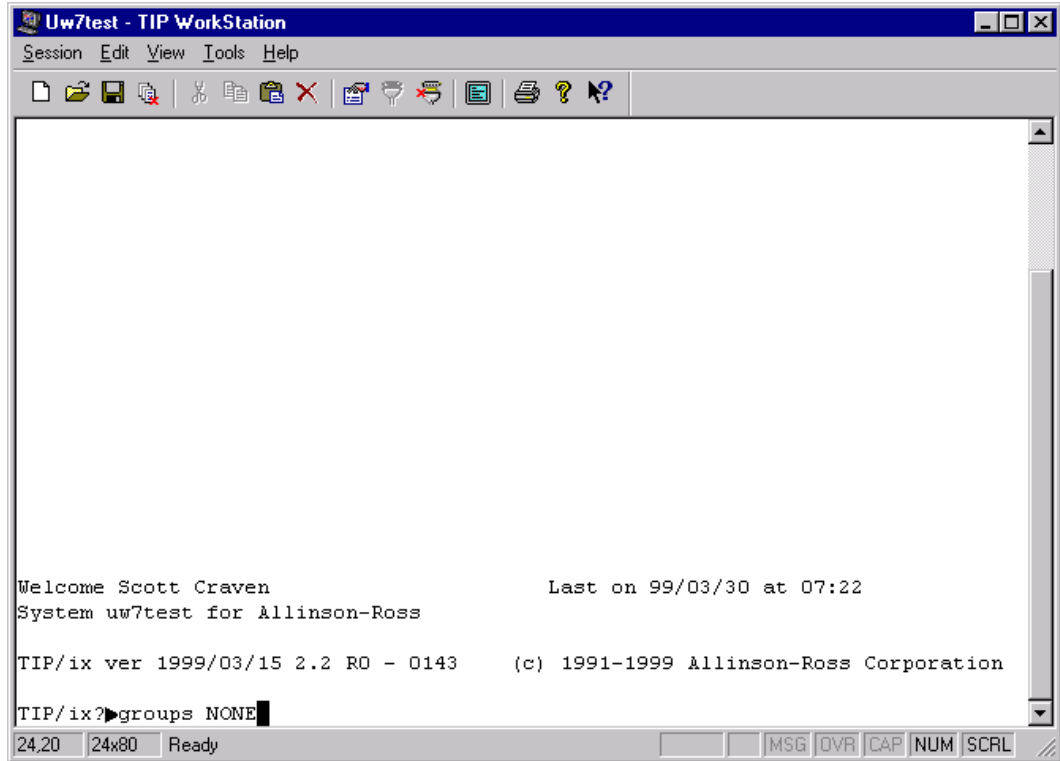
If a parameter contains a character string other than NONE, then it is presumed to be the name of the desired group for the corresponding position in the active group list. If the value supplied does not match one of the user's elective groups or one of the groups in the user's Group set or Logon set then the groups transaction will fail and the active groups will be left unchanged.

Additional Considerations:

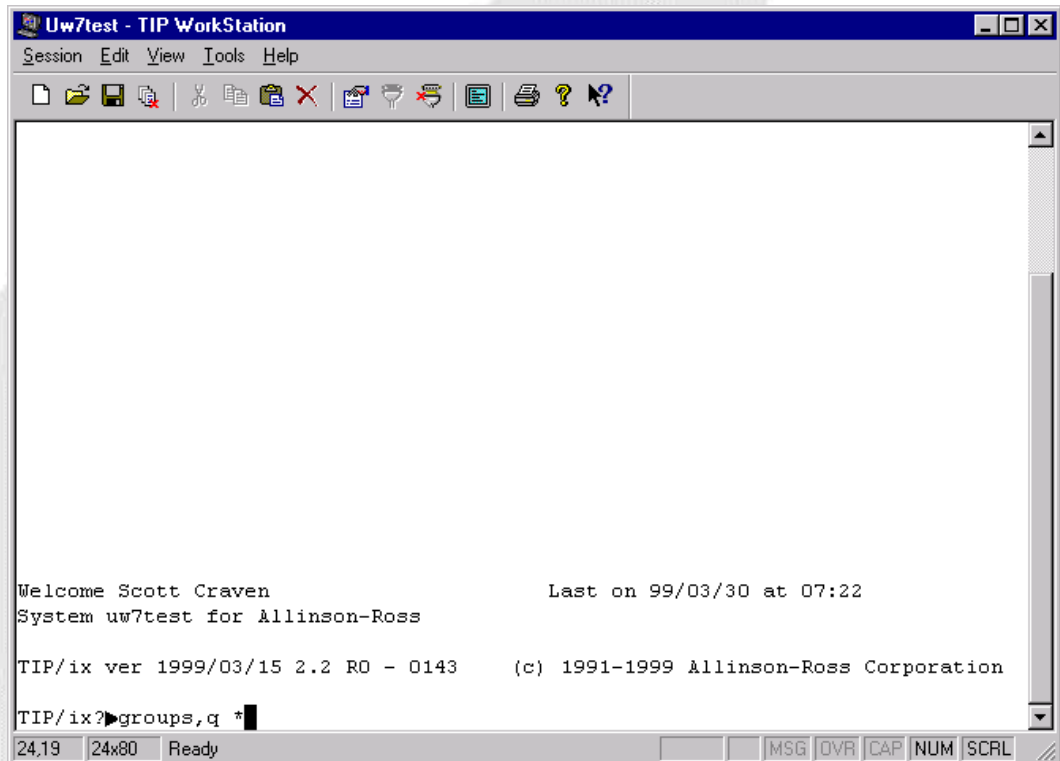
- If any of the group names supplied are not one of the user's eligible groups, the groups transaction will not change any of the active groups.
- groups is, therefore, an "all or nothing" operation - either all of the group names provided are valid or no action is taken.

Examples:

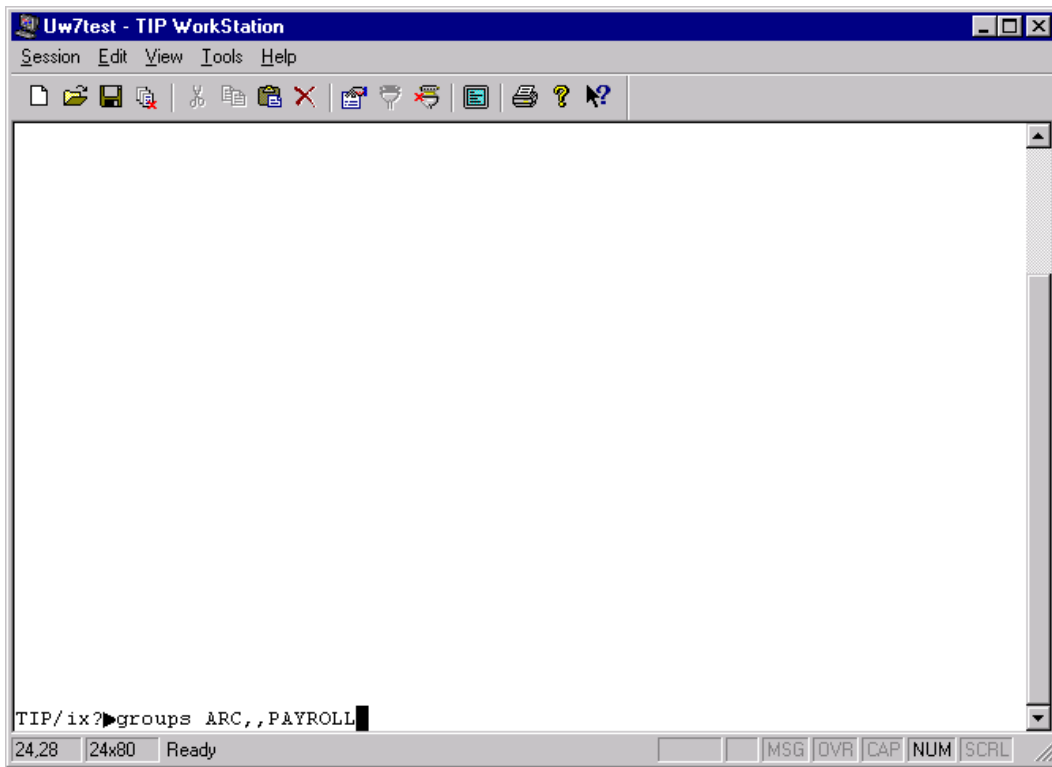
The following example will remove the first group in the active group list for this TIP/ix session. What was the second active group will now become the first active group. All other active groups will also shift up one position in the active group list.



The following example will set the user's active groups to the groups that are set at logon time. This is determined by the elective groups and Logon set in the user id definition (see **smuser**). The "q" option will suppress any screen output.



The following example will attempt to change the first and third active groups for the TIP/ix session. All other active groups for the TIP/ix session remain unchanged.



If the "q" command line is not specified and no errors occur then the **groups** program will display the following screen format. Notice that this is identical to the information displayed by the TIP/ix **wmi** program.

```

uw7test.tws - TIP WorkStation
Session Edit View Tools Help
Thursday April 29 1999

User-id: TEST1
Groups:  ARC      HR      PAYROLL  EDP
        MIS,29   DPD     TESTGRP, TIGERS
        YANKEES REDSOX,2 ORIOLES  TWINS
        METS    DODGERS PADRES   RANGERS

Security: 9
Account Number:
Terminal: 9750 - 192.168.1.190 #1

Site name: uw7test
TIP/ix Version: 1999/03/31 2.3 RO - 0000
Node: uw7test Locap: UW7TEST
UNIX version: UnixWare 7.0.1 i386

MBP Compiler: N/A
MF Compiler: 4.1.06-e
Using TIP/dbi: No
ORACLE version: N/A
Working Directory: /home/eduardov

TIP/ix?
24.9 24x80 Ready MSG OVR CAP NUM SCRL
    
```

The active groups are displayed in order from top to bottom and from left to right. Therefore, in the preceding screen “ARC” is the first active group and “RANGERS” is the sixteenth active group.

If a number is displayed to the right of a group name then that represents the user’s security level within that group. If no number is displayed next to a group then the user’s general security level applies to access within that group. In the preceding example the security level within the group “ARC” is 9 (value displayed for the field “Security”) and the security level within the group “MIS” is 29.

Error Conditions:

- The groups transaction reports errors by displaying a message on the terminal (unless the "q" command line option is specified). Programs which intend to invoke the groups transaction (via TIPSUB for example) normally set the CDA-OPTION field to "q" before issuing the call to transfer control.
- The groups transaction clears the first 64 bytes of the CDA to spaces (the eight parameter areas) if the requested changes were NOT completed; otherwise, the CDA parameters will be set to the user's first eight elective groups as set by the call to groups.

help - Display Help

The **help** program is a utility that displays help information for a specified online program. The user may ask to see the help information for utility programs supplied with TIP/ix.

Help information may also be provided (by the installation administrator) for the installation's user programs (or any other item of interest).

Syntax:

```
help [name] [,page#] [,libname] [,prefix]
```

Where:

name Identifies the name of the program or item for which help is requested. If omitted, the **help** program displays a menu listing all items for which help is available.

page#

Identifies which page of help information is to be displayed first. The default is the first page. If a page number is specified that is out of range (including a non-numeric value), the **help** program displays the first page of help information.

libname

The TIPFCS name of a TIP/ix library file where the **help** program will look for help information. If this parameter is supplied then **help** will only display help information from this TIP/ix library (UNIX directory).

If "libname" is not specified then **help** will first look for help information in the TIP/ix library with a TIPFCS name of "HELP". If the requested information is not found then it will look in the TIP/ix library with a TIPFCS name of "TIPHELP".

TIP/ix has a file type of library that is used to associate an 8 character TIPFCS name with a Unix directory. TIPFCS provides a mechanism for accessing individual files (library elements) in a library.

prefix

Specify this parameter to override the default library element name prefix that the **help** program uses to construct the library element name to locate.

If this parameter is omitted, the default prefix is "THU". See also "Additional Considerations" below.

Only the first three characters of this parameter are relevant.

Example:

Display the supplied help information for the program "whoson".

```
help whoson
```

Error Conditions:

The requested help information may not be available.

If the **help** program gets PIB-LOCKED status in response to a request to open an element from one of the HELP libraries then **help** will display a message indicating that the library is unavailable and that [FOPEN](#) may be required to access the library.

Additional considerations:

- Help information is stored as a library element with the name of the element formed as follows: "THUxxxxx" (the prefix THU may be overridden on the command line). The last five characters of the element name are the first five characters of the name the user is expected to key in as command line parameter.
For example: THUWHOSO for [WHOSON](#).
- The help program first attempts to read this element from a library with the logical file name "HELP". If the element (or file) does not exist, a second read is attempted from the "TIPHELP" library.
- You can establish your own library containing help information by creating elements in a library which has been defined by smsec and smfile with the name "HELP".
- Avoid modifying the TIPHELP library directly because it is completely rebuilt when a new release of TIP/ix is installed (and any alterations would be lost).

Call Another Help Module

The CALL directive is used to display help that is subordinate to the help currently being displayed.

Syntax:

```
/CALL code,suffix[,security] [,prefix][,page#]
```

Where:

code The 1 to 8 character value that will be keyed on the HELP screen.

prefix Specify this parameter to override the default library element name prefix that the **help** program uses to construct the library element name to locate.

If omitted, the default prefix is "THU". Only the first three characters of this parameter are relevant.

page#

Identifies which page of help information is to be displayed first. The default is the first page. If you specify a page number that is out of range (including a non-numeric value), the **help** program displays the first page of help information.

suffix The 1 to 5 character suffix that will be used to generate a **help** module name (that is, THUsuffi).

security

The minimum security required to see the help defined by this directive. Default is 255.

Additional considerations:

/CALL causes **help** to TIPSUB to itself.

Chain to Another Help Module

The CHAIN directive is used to display help that is equivalent to the help currently being displayed.

Syntax:

```
/CHAIN code,suffix[,security][,prefix][,page#]
```

Where:

code The 1 to 8 character value that will be keyed on the HELP screen.

suffix The 1 to 5 character suffix that will be used to generate a HELP module name (that is THUsuffi).

security

The minimum security required to see the help defined by this directive. Default is 255.

prefix Specify this optional parameter to override the default library element name prefix that the HELP program uses to construct the library element name to locate.

If omitted, the default prefix is "THU". Only the first three characters of this parameter are relevant.

page#

This optional parameter identifies which page of help information is to be displayed first. The default is the first page. If you specify a page number that is out of range (including a non-numeric value), the HELP program displays the first page of help information.

Additional considerations:

/CHAIN causes HELP to TIPXCTL to itself.

Display Another Help Screen Full

End the current screen full (max 22 lines). Any remaining data will be displayed on subsequent screens.

Syntax:

/EJECT

Define a Module Title

Supply data for the Title line of the HELP system's screen format (max 60 characters).

Syntax:

/TITLE ----- 60 character title -----

Define Help Module Security

Define the security level required to examine a HELP module.

Syntax:

/SECUR [security]

Where:

security

The minimum security required to see this help. Default is 255.

Sorted Call/Chain Table

Indicate that the Call/Chain table is sorted in ascending order (HELP finds an entry quicker).

Syntax:

/SORTED

Display Call/Chain Table

Display all the entries in the Call/Chain table as a 6, 7 or 8 column list on the terminal.

Syntax:

```
/TABLE [number][,skip]
```

Where:

number

The number of entries (6 through 8) to be displayed on each line. Default is 8.

skip

The number of screen lines to skip before the table is displayed. Default is 1.

isonline - is file available

Check if the specified file is valid and is available to TIP/ix. This can be used in shell scripts to control branching.

We recommend using logical filenames to check if a file is available to TIP/ix. See Additional Considerations for the reasons.

Syntax:

```
isonline -h  
isonline [-v | -d] filename  
isonline [-v | -d] -l logicalfn
```

Where:

- h** Displays help text.
- v** Verbose. Display message. If not specified, **isonline** just exits with the appropriate status code.
- t** Just test if TIP/ix system is up or down.
- d** Use detailed access exit codes. Then number that is returned can be looked up in the Unix error code table located in

```
/usr/include/errno.h
```

or

```
/usr/include/sys/errno.h
```


filename

The physical file name to be checked. The filename is case sensitive. (See Label/Path in [smfile](#).)

If the filename begins with a slash (/), it is interpreted as an absolute path. For example:

```
isonline /data/red/orders/log
```

Otherwise, the filename is interpreted as a suffix (trailing string). The following examples could all match the same file:

```
isonline log
isonline orders/log
isonline ders/log
isonline og
```

- l** Treat parameter as a logical file name. When this option is used the file name supplied must match a complete TIP/ix file definition name (no partial matches).

logicalfn

File definition name (TIP\$SYS file name). The value supplied is converted to uppercase because file definition names are always uppercase.

See [smfile](#) for a description of TIP/ix file definitions.

The **isonline** utility returns a status code that can be tested in a script:

Code	Meaning
0	Filename is now available to TIP/ix.
1	Filename will be closed as soon as users stop using it. (Close is pending.)
2	Filename is not a valid TIP/ix filename.
4	TIP/ix is not currently running.
10	Filename is currently closed to TIP/ix.

Example:

You can use **isonline** to delay processing in a shell script until TIP/ix closes a file.

```
fclose abcfile
# fclose returns exit status of 1 if close
is pending (there are
TIP/ix users accessing the file)
if [ $? -eq 1 ]; then
status=1
while [ $status -eq 1 ]
```

```

do
sleep 60 # wait 60 seconds
isonline -l abcfile
status=$?
echo "Waiting for TIP/ix to close file
abcfile"
done
if [ $? -eq 10 ]; then
echo "File abcfile closed."
fi
fi

```

Additional Considerations:

isonline scans an internal table of files known to TIP/ix (the same table that is displayed via the "[status i](#)" command).

When the '-l' option is **not** used, **isonline** looks for the first *physical* name that matches. This could cause a problem if there are two files with the same physical name. For example:

```

fclose myfile
isonline foo

```

Logical File Name	Path/Label (Physical File)	Status
YOURFILE	/here/foo	open
MYFILE	/there/foo	closed

Unfortunately the first foo in the table is the wrong one.

- When the '-l' option is used, **isonline** looks for the first *logical* name that matches. Normally there is no problem because [smfile](#) forces logical file names to be unique.

```

isonline -l myfile

```

- However, the same physical file can be defined more than once (not a recommended practice). This could lead to confusion if one file definition was set offline to TIP/ix and the other was online to TIP/ix. For example:

```

fclose XYZ
isonline -v foo

```

Logical File Name	Path/Label (Physical File)	Status
ABC	/data/foo	open
XYZ	/data/foo	closed

isonline reports "file available" because ABC is before XYZ in the table and we have not closed ABC.

isreorg – Reorganize ISAM files

This is a batch utility that can be used to copy/reorganize ISAM files. Do this periodically will recover disk space occupied by deleted records.

The isreorg usage information follows:

```
TIP/ix 2014/09/18 2.5 R0 - 0275 ©1991-2013 Inglenet Business Solutions
Re-Organize an ISAM file
isreorg [-opts] filename
where:
-i inputfile      File to copy
-i @inputfile    2200 style file name to copy; Checks TIPECLQUAL env var
-o outputfile    Default: overwrite input
-o @outputfile   2200 style file name to write over
-q              quiet; No messages
-V             Convert input C/D-ISAM to VBISAM format for COBOL-IT
-D             Convert COBOL-IT VBISAM file into C/D-ISAM file format
-r             Must be 'root'
-t             Report timing information
-p             Add index before copy. Default is after
-P            Do NOT retain input file permissions and naming
              Default: is to keep things the same
-d            Use 4 byte duplicate key control for D-ISAM
-f            Force output file creation; Erase before start
-I            INIT output file; I.e. Copy no data
-x n          Re-Organize by index 'n'
-X bksz       Use index block size; 1024 or 2048

-l            Create symbolic link of file --> file.dat
-L            Do NOT create link of file --> file.dat
-R n          Read input as 'record sequential' of record size 'n'
              (record slot size on disk is 'n + 1')
              (input file has records end with LF if good
              or NUL if deleted)
-k pos:len    Defines index as position 'pos' for length 'len'
              'len' may be followed by D for DUPS, N for NODUPS
-e pos:val    Skip records if 'val' in position 'pos' (zero relative)
              val may be a letter or 0xHH where HH is a hex value
-s            Write output as a 'record sequential' file
-m            file is 'line sequential' instead of 'record sequential'
```

A simple example of use would be:

```
isreorg tspfile
```

If you wanted to change the file format from C-ISAM (D-ISAM) to VB-ISAM the command would be:

```
isreorg -V tspfile
```

The input and/or output file names may be a complete path name or just a file name if in current directory.

If the file name starts with an at-sign '@' then the name is taken to be a 2200 style file name. The TIPHOMEDATA environment variable defines the base directory where the data files are stored. You can use names like MYQUAL*FOOBAR, MYQUAL*FOOBAR(+1) for the next cycle of the file, or just FOOBAR or FOOBAR(-1). If the file qualifier is omitted then the environment variable TIPECLQUAL defines the default qualifier to

use. Due to the * and brackets using 2200 file names you will likely have to enclose the file name in quotes to avoid the Unix shell doing something you do not intend. Example `-o '@ME*MAST(+1)'`

jrnswap - Swap Journal Files

If you are using two journal files (dual journaling), the **jrnswap** transaction provides a way to manually close one journal file and open the alternate journal file. You must be a TECH-level user to use this transaction.

To enable dual journaling, set two different JRNFILE parameters in the `tipix.conf` file.

To automatically swap journal files, set a maximum journal file size with the JRNFSIZE parameter in `tipix.conf`. Once the active journal file reaches this size, TIP/ix automatically performs a journal file swap.

When TIP/ix swaps journal files (either automatically or as a result of executing **jrnswap**) it simply switches between the two files named in the PARAM JRNFILE= statements in `tipix.conf`. Since TIP/ix cannot swap to an existing journal file, you must ensure that the “swapped out” (used) journal file is *saved, renamed or moved before* the next swap takes place. You should create a script that does whatever your site requires to save the swapped out (used) journal. Your script must at least rename the swapped out journal so that it is out of the way of the next journal swap.

Once the journal file has been swapped, TIP/ix *attempts* to run the journal swap script. By default, TIP/ix looks for the journal swap script in **\$TIPROOT/scripts/tipjrnsw**. You can specify a different journal swap script with the **JRN_SWAP_SCRIPT=** parameter in `tipix.conf`.

When TIP/ix executes the swap script, it passes the name of the journal file that it has just finished using (swapped from) as an argument to the swap script.

Syntax:

```
jrnswap
```

Example:

A typical journal swap script will use the parameter it receives from TIP/ix to save the journal file under another name. To keep multiple journal files, append the saved name with a date or sequence number. For example:

```
SUFFIX=`date '+%y%m%d.%H%M%S'`
SAVEJRN=/data/savejrn
mv $1 $SAVEJRN.$SUFFIX
```

```
compress $SAVEJRN.$SUFFIX
```

Error Conditions:

If you do not have dual journalizing configured, you will get the following error message:

```
Dual journaling not configured.
```

- If the journal file you are swapping to has journal records in it, you will get the following message:

```
Journal file xxxxxxxx is not empty.
```

- If TIP/ix successfully executes this transaction, you will get the following message:

```
Journal file swapped.
```

- Once the journal file has been swapped, TIP/ix *attempts* to run the journal swap script. If there is no swap script, or if an error occurs, you will get the following message:

```
Error executing swap script swapscriptname
```

Additional Considerations:

For operational considerations, see the *File Recovery* chapter in the *TIP/ix Installation and Operation manual*.

lopen - Open Connection to LOCAP

The **lopen** transaction opens a connection path to one or more specified remote LOCAPs that are running TIP Distributed Transaction Processing (DTP).

The connection to other LOCAPs may be closed by using the [lclose](#) utility.

This program does not operate interactively.

Syntax 1- online:

```
lopen, [q] locap 1 ... [,locap 8]
```

Syntax 2 – batch:

```
lopen [-q] locap 1 ... [,locap 8]
```

Where:

- ,q** This option (quiet) instructs lopen to suppress the usual informational messages that are displayed.
- q** This option (quiet) instructs lopen to suppress the usual informational messages that are displayed.

locap 1...

The names of the LOCAPs to be opened. At least one name must be specified. You may specify up to eight LOCAPs on the command line. Each LOCAP name may be specified as a unique name or may use prefix notation.

Examples:

Open all the LOCAPs that start with "TST" and the two specified LOCAPs (FROG and TOAD). Note that "*TST" is TIP prefix notation (not a UNIX shell regular expression).

```
lopen *TST FROG TOAD
```

- This is the batch version of the previous example.

```
lopen -q \*TST FROG TOAD
```

Return Codes:

Code	Meaning
0	Successful.
1	Close is held pending.
2	Item does not exist.
4	Security violation.

Additional Considerations:

When running the **lopen** program in batch, the TIPROOT environment variable indicates which TIP/ix system is being referenced.

Iclose - Close Connection to LOCAP

The **lclose** utility closes the connection path to one or more specified remote LOCAPs that are running TIP Distributed Transaction Processing (DTP).

The connection to other LOCAPs may be re-opened by using the [lopen](#) utility.

This program does NOT operate interactively. Up to eight LOCAP names may be supplied on the command line. Each LOCAP may be specified as a unique name or may use prefix notation.

This program may operate in an online or batch mode.

Syntax 1 – online:

```
lclose[,q] locap 1 ... [,locap 8]
```

Syntax 2 – batch:

```
lclose [-q] locap 1 ... [,locap 8]
```

Where:

- ,q** This option (quiet) instructs lclose to suppress the usual informational messages that are displayed.
- q** This option (quiet) instructs lclose to suppress the usual informational messages that are displayed.

locap 1...

The names of the LOCAPs to be closed. At least one name must be specified.

Examples:

The following command closes the specified LOCAPs.

```
lclose TST1 TST2
```

- The following command closes all LOCAPs with a prefix of HR. Note that “*HR” is TIP prefix notation (not a UNIX shell regular expression).

```
lclose *HR
```

Return Codes:

Code	Meaning
0	Successful.
1	Close is held pending.
2	Item does not exist.
4	Security violation.

Additional Considerations:

When the path to a LOCAP is closed, attempts to make a connection to that LOCAP (by the [connect](#) transaction for example) are denied. Existing connections are not affected by the **lclose** operation.

The **lclose** operation may be held pending until all users have relinquished control of the LOCAP. As soon as the **lclose** is issued however, the LOCAP is marked as "unavailable for logon" - new requests to use the LOCAP will not be honored.

When running the **lclose** program in batch, the TIPROOT environment variable indicates which TIP/ix system is being referenced.

Ireset - Reset Connection to LOCAP

The **Ireset** transaction closes then reopens a connection path to one or more specified remote LOCAPs that are running TIP Distributed Transaction Processing (DTP).

This program does not operate interactively.

Syntax 1 – online:

```
Ireset, [q] locap 1 ... [,locap 8]
```

Syntax 2 – batch:

```
Ireset [-q] locap 1 ... [,locap 8]
```

Where:

- ,q** This option (quiet) instructs Ireset to suppress the usual informational messages that are displayed.
- q** This option (quiet) instructs Ireset to suppress the usual informational messages that are displayed.

locap 1...

The names of the LOCAPs to be reset. At least one name must be specified. You may specify up to eight LOCAPs on the command line. Each LOCAP name may be specified as a unique name or may use prefix notation.

Examples:

Reset all the LOCAPs that start with “TST” and the two specified LOCAPs (FROG and TOAD). Note that “*TST” is TIP prefix notation (not a UNIX shell regular expression).

```
Ireset *TST FROG TOAD
```

- This is the batch version of the previous example.

```
Ireset -q \*TST FROG TOAD
```

Return Codes:

Code	Meaning
0	Successful.
1	Close is held pending.
2	Item does not exist.
4	Security violation.

Additional Considerations:

When running the **Ireset** program in batch, the TIPROOT environment variable indicates which TIP/ix system is being referenced.

mail - TIP/ix Mail System

The TIP/ix electronic mail utility (**mail**) allows a TIP/ix user to communicate with other TIP/ix users individually or in groups.

Each user's mail is stored in a mailbox until he wishes to read it. A mailbox can contain up to 376 messages.

The documentation for the **mail** program has been divided into four sections:

Glossary

Definitions of terms used in mail.

User Guide

An introduction to using mail.

Syntax

A description of the commands that mail recognizes.

Programmer Guide

A guide to calling mail from another online TIP/ix program.

Mail System Glossary

Confidential Mail

A letter that has been sent "confidentially" (see the SEND command for details) is not immediately displayed on the terminal for its recipient. First a preview, showing the letter's subject and author, will be displayed on the terminal with a warning that the letter has been marked "confidential" by the sender.

Distribution List

A list of valid TIP/ix user ids, maillist names, printer names, or queue destinations that defines the recipients of the letter being Sent or Forwarded.

A distribution list must be supplied as part of the SEND or FORWARD command or the mail system will prompt the sender for the list.

maillist

A list of valid TIP/ix user ids, names, printer names, or queue destinations that are pre-defined in the maillist file (see the documentation of the [maillist](#) program in this manual.) A one to seven character alphanumeric identifier that **MUST BE** prefixed with the greater-than character (>) to differentiate it from a user id.

Memo Mail

A letter that has been read but is being kept active. New mail becomes Memo mail if you press **MSG WAIT** while reading it or if you enter the **End** or **Quit** commands.

Memo mail is converted to Old mail if you press **XMIT** while reading it.

The advantage of Memo mail is that it allows you to mark the new letters you receive as having been read and left for further action in your mailbox. Memo mail also prohibits [RETraction](#) of mail from another user id once it has been looked at by that user.

Must Reply Mail

A letter that has been sent "must reply" (see the [SEND](#) command for details) forces any recipient(s) of the letter to **REPLY** to the letter's author when the letter is first read.

New Mail

A letter that has never been read. New mail becomes Memo mail if you press **MSG WAIT** after reading it. It becomes Old mail if you press **XMIT**.

Old Mail

A letter that has been read and marked old. New mail and/or Memo mail is marked "Old" when you press **XMIT** after reading it.

Pending Mail

A letter that has been sent with a future date; the letter is in the mailbox but it is **NOT** available until some future date.

Registered Mail

A letter that has been sent "registered" (see the [SEND](#) command for details) will send mail to the author of the letter to indicate that the recipient has read the letter.

This registration is sent the first time each recipient reads the letter.

user id

A valid user of the TIP/ix system as defined in the TIP/ix catalogue.

User-prefix

A partial user id that allows some commands to act on groups of user ids. TIP/ix prefix notation is allowed; for example, ***PAY** means all user ids that begin with the string **PAY** and **!PAY** means all user ids that do not begin with the string **PAY**.

Mail User Guide

This section is intended for those who have little or no previous experience with the TIP/ix **mail** system. After reading this section, you will be able to send, read and delete letters.

The example that follows illustrates an interaction between two TIP/ix users:

JDOE (Jane Doe)
BJONES (Brian Jones)

Following a successful LOGON, JDOE enters the transaction name **mail** to invoke the TIP/ix mail system.

The message "Working, please wait" may be displayed while **mail** accesses the user's mailbox.

*******Screen shot missing*******

mail will display a summary of the NEW letters in JDOE's mailbox:

*******Screen shot missing*******

Field	Description
Let#	The reference number of the letter in JDOE's mailbox (copies of the letter in other mailboxes will probably have different reference numbers).
From	The user id of the person who sent the letter.
Date	The date the letter was sent.
Time	The time the letter was sent.
Type	Listed below: <ul style="list-style-type: none"> New New letter. Memo Memo letter. Old Old letter. CON Confidential letter. REG Registered letter.
Subject	A short description of the letter's contents.

The directory screen also contains a line to be used to enter **mail** commands and any parameters that they may require.

The [Send](#) command is fully documented in the **mail** syntax section. In its simplest form, it is entered as "[Send](#)" or "[S](#)", followed by a space and then a list of recipients of the letter. In this case BJONES, the user id of the person to whom Jane wishes to send a letter.

*******Screen shot missing*******

When Jane presses the **XMIT** key, the **mail** system verifies that "BJONES" is a valid TIP/ix user id and then enters letter composition mode.

"Subject" is used to supply a concise description of the letter's contents.

Pressing **XMIT** sends the letter (see "Continue(y/n/page#)" below).

Pressing **F2** displays the next screen full of data.

Pressing **F3** displays the distribution list.

When in the distribution list screen, press **F10** to obtain a list of valid TIP/ix users; press **F11** to invoke **maillist**.

Pressing **MSG WAIT** stops letter composition without sending the letter.

A portion of the letter's body is next (20 lines per screen full to a maximum of 100 lines).

The field "Continue(y/n/page#)< >" affects the action taken when the **XMIT** key is pressed:

(space), 'n' or 'N'

Send the letter to its intended recipients.

y or Y

Cause the next screen of data to be displayed.

a number ('1' to '5')

Displays the specified screen of data.

When Jane is finished composing the letter, she tabs to the "Continue" field, presses the **XMIT** key and her letter is sent to BJONES.

*******Screen shot missing*******

Termination of the send command returns Jane to the directory screen and displays a message that indicates the status of the send; in this case, successful completion.

Jane's letter is now in BJONES's mailbox and will stay there until he reads it.

Since it's the end of the day and time to leave, Jane exits from **mail** using the **END** command (pressing the **MSG WAIT** key has the same effect).

*******Screen shot missing*******

The next time Jane starts up a TIP/ix session the number of new mail messages in her mailbox will be displayed. This message is only displayed if there are some new mail messages. At the end of the message is a "MAIL RN" command with a convenient tab stop behind it. Just tab to this location and press **XMIT** to read all new mail messages.

*******Screen shot missing*******

When Jane invokes **mail**, she sees a directory that shows two new messages; one is from BJONES and has the same subject as her letter to him.

*******Screen shot missing*******

To read that letter, Jane keys the number "1" and presses the **XMIT key**. The **mail** system displays the requested letter and allows Jane to read it.

The line of information that was displayed in the directory, is repeated at the top of this display.

The next line shows the one character codes and Function keys that are valid while reading mail.

Pressing the **XMIT key** performs the action indicated by the supplied 1 character code (see the syntax section of this documentation for a description of these codes).

Pressing **F2** displays the next screen of data.

Pressing **F3** displays the distribution list.

When in the distribution list screen, press **F10** to obtain a list of valid TIP/ix users; press **F11** to invoke [maillist](#).

Pressing **F4** ends the read command.

Pressing **MSG WAIT** stops reading this letter.

The first portion of the letter's body is displayed; notice that the information that Jane sent to BJONES is separated from BJONES's response by a line that indicates the date, time and identity of the person who made the response.

To respond Jane enters "r" (reply) and presses the **XMIT key**.

*******Screen shot missing*******

mail re-displays the letter with a line that identifies Jane and shows the date and time of her response.

Jane enters her reply on the line that follows, tabs to the "continue" field and presses the XMIT key to send the modified letter back to BJONES.

*******Screen shot missing*******

The original letter is re-displayed, with a message indicating that the reply completed successfully, to allow Jane to dispose of it as she wishes

Pressing the **MSG WAIT** key retains the letter as New and returns Jane to the directory.

Pressing the **XMIT** key retains the letter as old and returns Jane to the directory.

Entering a "D" and pressing the **XMIT** key deletes the letter from Jane's mailbox and returns her to the directory.

*******Screen shot missing*******

The directory is displayed without the letter that was just deleted and with a message indicating that 1 letter has been deleted.

Jane exits **mail** using the END command.

*******Screen shot missing*******

Mail Program Syntax

The **mail** program may be invoked:

with parameters to perform a single function and then terminate, or

without parameters to display a list of New mail (if any is available), or a list of old mail (See Directory command) and wait for further interactive commands.

Syntax:

```
mail[/?] [parameters]
```

Where:

? Command line option that indicates that mail should continue execution only if there is new mail available. The mail program terminates immediately if "?" is specified in the command line options and there is no new mail available.

parameters

A valid mail command (see table following).

Example:

Invoke mail to send a letter to the user id "BOSS".

```
mail s boss
```

Command Summary:

The **mail** program recognizes the following commands (uppercase letters in the command indicate required letters; lowercase letters are optional):

Command	Description
Access	Grant access to your mailbox to another user
Change	Customize mail for individual preferences.

DElete	Delete a letter from a mailbox.
Directry	Display a summary of letters (directory) in a mailbox.
End	Terminate the mail program.
EXport	Print (then mark old) a letter from a mailbox.
Forward	Pass a letter on to another user or group of users.
FIN	Terminate the mail program (and log off).
Help	Display a summary of mail commands and parameters.
Import	Bring data into a letter via the COPY utility.
LOGOFF	Terminate the mail program (and log off).
LOOK	Lists all of the mail you have sent to a specific user id and allows you to read or print any or all of those letters.
New	Read from another user's mailbox (if permitted).
Print	Print a letter from a mailbox.
Quit	Terminate the mail program (and log off).
Read	Read a letter from a mailbox.
REStore	Restore a previously saved letter from a library.
REtract	Delete unread letters from another user's mailbox.
SAve	Save a letter in a library.
Send	Send a letter to a user (or group of users).
Update	Use a TIP/ix editor to compose a letter. Not implemented.
USers	Invoke USERS transaction to list user ids.

A - Access a Mailbox

This command is used to grant access to your mailbox to another user or group of users. Allows secretarial access to a manager's mailbox.

Syntax:

```

Access user id
Access group
    
```

Where:

```

user id
    The id of the user to have access to your mailbox.
    
```

group

The group to which you want to grant access to your mailbox. Any user who belongs to this group can access your mailbox. (smuser is used to assign group membership.)

C - Change Preferences

This command is used to customize **mail**. The following screen formats show the customization screens and the default values for each option.

Syntax:**Change**

*****Screen shot missing*****

*****Screen shot missing*****

DEL - Delete Mail

This command is used to delete old mail from a user's own mailbox.

Syntax:

```
DELeTe # [,#,#,#...]
DELeTe user-prefix
DELeTe #-#[,type[type]]
```

Where:

[,#,#,#...]

One or more letter numbers.

#-#

A letter number range (example: "1-12"). If you specify an invalid range, mail deletes all valid letters that are possibly in that range.

[,type[type]]

The letter types Old or Memo. New mail cannot be deleted before it is read.

user-prefix

A user id or partial user id.

Examples:

Delete letters 1, 5 and 12 from the mailbox.

```
DE 1,5,12
```

Delete all letters from "DAVE".

DE *DAVE

Delete all Old and Memo letters from "3" to "25" (inclusive).

DE 3-25,OM

Additional Considerations:

DEL *

Deletes only Old mail.

DEL *,m

Deletes all Memo mail.

DEL *,mo

Deletes all Memo and Old mail.

DI - Mailbox Directory

This command is used to display summary information about letters in the mailbox.

Syntax:

DIRECTRY [user id] [,type] [,#]

Where:

user id

A valid TIP/ix user id, the default is all user ids.

type

Old mail, New mail or both, the default is both.

#

A starting letter number.

Example:

Display a summary of all letters in the mailbox.

Dir

Display a summary, starting at letter # 21, of all old letters in the mailbox that were NOT sent by "JOE".

D !JOE,O,21

Additional Considerations:

mail commands may be entered on the Directory screen.

E - End Mail Program

This command is used to terminate **mail**.

Syntax:**End****EX - Export Mail**

This command is used to print letters and mark them old.

Syntax:

```

Export # [,#,#,#...] [,type] [,printer]
  [,case]
Export user-prefix [,type] [,printer]
  [,case]
Export #-# [,type [type] [type]] [,printer]
  [case]

```

Where:**# [,#,#,#...]**

One or more letter numbers.

#-# A letter number range (example: "1-12").**user-prefix**

A user id or partial user id.

printer

The name of the printer to be used (default is the printer defined via the Change command).

case

The desired case of the printout. the default is Upper when printing on a main site printer; Lower when printing on an auxiliary printer.

[,type [type] [type]]

The letter types Old, Memo or New. OMN is equivalent to "*" or space.

Example:

Print all letters, sent to you by "JOE", using the main site printer (PRNTR), in lower case.

```
EXPORT *JOE, , , LOWER
```

Export all Old and Memo letters from 3 to 25 inclusive.

```
EX 3-25, OM
```

FIN - End Mail and Logoff

This command is used to terminate **mail**. If possible, the user of **mail** will be logged off TIP/ix.

Syntax:

```
FIN
```

F - Forward Mail

This command is used to redirect mail received from one user to another user or group of users. The letter may be modified before it is redirected.

Syntax:

```
Forward letter# [,distribution]  
F[modifiers] letter# [,distribution]
```

Where:

letter#

The number of a letter in the mailbox.

distribution

List of user ids, maillist names, or printer names. See mail glossary.

modifiers

One or more of the following codes:

C	send as a confidential letter.
M	send as a Must Reply letter.
Q	send quietly (no unsolicited message).
R	send as a registered letter.
U	send urgently (with unsolicited message).

Example:

Forward letter 1 to TOM, DICK and HARRY and to all users currently in the mail list MANAGER.

```
FC 1 , TOM , DICK , HARRY , >MANAGER
```

Additional Considerations:

The sender of a letter will receive a copy of the letter if:

- he is included in the distribution list
- the "always receive a copy" Change command flag is set
- he is in one or more of the maillists being sent to and the "remove from maillists" Change command flag is NO.

Multiple occurrences of a user id (in the distribution list or in maillists) will be detected and deleted.

Function key 2 (**F2**) may be used to page forward through the letter.

Function key 3 (**F3**) may be used to display (and alter) the distribution list that will be used to send the letter.

Error Conditions:

The specified letter may not be found, or one or more of the specified user ids may not be valid.

H - Help for Mail Commands

This command is used to invoke the TIP/ix HELP system to display a summary of this documentation.

Syntax:

Help

I - Import Letter Text - Not Implemented

This command is used to send a pre-composed letter from a file.

Syntax:

Import library,element,type
Import group,buffer,E

Where:

library,element,type

A "library element" that contains the letter body:

library name

element name

element type - S (source)

group,buffer,E

A TIP/ix edit buffer that contains the letter body:

edit buffer group

edit buffer name

edit buffer type (E)

Example:

Read the MSDOS file C:LETTER.001 and allow it to be sent to any valid TIP/ix user id or maillist.

Import C:LETTER.001

Additional Considerations:

The letter distribution screen will be displayed after the letter body has been Imported (and modified if necessary).

L - Look at Mail

This command allows you to read or print mail that you have sent to another user id, which resides in that user id's mailbox.

Syntax:

```
Look user id
```

Where:

```
user id  
A valid TIP/ix user id.
```

Example:

```
Look JDOE
```

Results in:

```
*****Screen shot missing*****
```

LOGOFF - End Mail and Logoff

This command is used to terminate **mail**. If possible, the user of **mail** will be logged off TIP/ix.

Syntax:

```
LOGOFF
```

N - Read Another Mailbox

This command is used to allow one user of **mail** access to another user's mailbox. Allows "bulletin board" mailboxes to be set up. Allows secretarial access to a manager's mailbox.

Syntax:

```
New user id
```

Where:

```
user id  
A valid TIP/ix user id. NO prefix specification allowed.
```

Example:

Open the mailbox for user "MESSAGES".

```
New MESSAGES
```

Additional Considerations:

An unmodified Directory command will be executed if access to the alternate mailbox is granted.

Only the following commands may be performed against the alternate mailbox: Directry, End, Quit, FIN, LOGOFF, Help, New, Print and Read (a letter's status will not be changed).

Error Conditions:

Access to the alternate mailbox, may be denied (access to another user's mailbox is granted if the value in the MAILREAD= keyword of that user's TIP/ix catalogue record is equal to the user id, group-1 or group-2 of the user requesting access or to "TIP\$Y\$").

A command issued after access is granted may not be permitted.

P - Print Mail

This command is used to print letters.

Syntax:

```
Print #[,#,#,#...] [,type] [,printer] [,case]
Print user-prefix [,type] [,printer] [,case]
Print #-
# [,type [type] [type]] [,printer] [,case]
Print #-# [,type [type] [type]] [, ,case]
```

Where:

[,#,#,#...]

One or more letter numbers.

#-# A letter number range (example:"1-12").

user-prefix

A user id or partial user id.

printer

The name of the printer to be used (default is the printer defined via the Change command).

case

The desired case of the printout. The default is Upper when printing on a main site printer; Lower when printing on an auxiliary printer.

[,type [type] [type]]

The letter types Old, Memo or Old. OMN is equivalent to "*" or space.

Examples:

Print all letters sent to you by "JOE" (or any user id prefixed by "JOE"), using the main site printer (PRNTR), in lower case.

```
PRINT *JOE, , ,LOWER
```

Print all Old and Memo letters from 3 to 25 inclusive.

```
PRINT 3-25,OM
```

Q - End Mail and Logoff

This command is used to terminate **mail**. If possible, the user of **mail** will be logged off TIP/ix.

Syntax:

```
Quit
```

R - Read Mail

This command is used to read letters from a mailbox.

Syntax:

```
# [,#,#,#...]
Read # [,#,#,#...] [,type]
Read #-# [,type[type][type]]
Read user-prefix [,type]
RN
RN #
RN user-prefix
```

Where:

[,#,#,#...]

One or more letter numbers.

#-# A letter number range (example: "1-12").

user-prefix

A user id or partial user id.

type

Old mail, New mail or both, the default is both.

[,type [type] [type]]

the letter types Old, Memo or New. ONM is equivalent to "*" or space.

RN All new letters (same as "Read *,N").

Starting letter number for the RN command.

Examples:

Read all new letters from DAVE (or all user ids prefixed by "DAVE".)

Read *DAVE,n

Read all Old and Memo letters from 3 to 25 inclusive.

Read 3-25,OM

Additional Considerations:

Function key 2 (**F2**) may be used to page forward through the letter.

Function key 3 (**F3**) may be used to display the distribution list that was used when the letter was sent.

The screen format associated with the Read command allows the following functions to be performed while reading a letter:

Entry	Description
space	Send the letter.
1-5	Display the specified page of the letter.
A	Print this letter (at the terminal's auxiliary printer).
B	Print this letter (at the bypass terminal's auxiliary printer).
D	Delete this letter.
E	End mail.
F	Redirect a copy of this letter (probably modified) to users who are on the original distribution list.
P	Print this letter (at the user's default printer as defined via the Change command).
Q	End mail and LOGOFF.
R	Redirect a copy of this letter (probably modified) to the author of the letter.

After the A, B, F, P or R functions, the original letter is re-displayed and another function may be entered.

RES - Restore Letter from Library - Not Implemented

This command is used to restore a letter from a library element.

Syntax:

REStore file,element

Where:

file The logical file name of the library.

element
The name of an element in that library.

Example:

Restore a letter from the element "LETTER1" in the library "MAILLIB".

```
REST MAILLIB/LETTER1
```

Error Conditions:

The specified element may not be found, or the specified element may not be in the format required by the restore command.

RET - Retract Letter

This command is used to delete mail from another user's mailbox or from your own mailbox.

A directory of all NEW mail sent by YOU to the other user is displayed on the terminal.

Enter the number of the letter to be RETracted and press **XMIT**.

That letter is deleted from the other user's mailbox, held in memory and displayed on the terminal.

The letter and/or distribution list may be modified and sent again (**XMIT**) or discarded completely and irrevocably (**MSG WAIT**).

Syntax:

```
RETRACT user id  
RETRACT maillist
```

Where:

user id
A valid TIP/ix user id. NO prefix specification allowed.

maillist
A maillist name. See mail glossary.

Example:

Retract any new mail that you have sent that is currently in THEBOSS's mailbox.

```
RET THEBOSS
```

Additional Considerations:

Another REtract command may be entered whenever the directory screen is being displayed.

Pending mail (See **mail** glossary) will be displayed in the directory and may be retracted.

Retracting by [maillist](#) functions as though a REtract command has been issued for every user id in the [maillist](#).

SA - Save Letter in Library - Not Implemented

This command is used to save a letter in a library element.

Syntax:

```
Save # [,#,#,#...],file,element
Save #-#,file,element
Save user-prefix,file,element
```

Where:

[,#,#,#...]

One or more letter numbers.

#-# A letter number range (example: "1-12").

user-prefix

A user id or partial user id.

file The logical file name of the library.

element

The name of the element that will be created in the library.

Examples:

Save letter number 1, as an element named "LETTR001", in the library identified in the TIP/ix catalogue as "MAILLIB".

```
SAVE 1,MAILLIB,LETTR
```

Save all letters from 3 to 25 inclusive as elements named "LETTR003", "LETTR004", and "LETTR005" in the library identified in the TIP/ix catalogue as "MAILLIB".

```
SAVE 3-25,MAILLIB,LETTR
```

Additional Considerations:

The first 5 characters of the element name will be concatenated with the 3 digit letter number to produce a unique element name for each letter.

Any blanks in the generated element names will be changed to dollar signs (\$).

The letter's subject is used as the comment for the saved library element(s).

Error Conditions:

The specified element(s) may already exist in the library named.

S - Send Letter

This command is used to send mail to another user or group of users. The letter is composed on the terminal.

Syntax:

```
Send [distribution]
S[modifiers] [distribution]
```

Where:

distribution

List of user ids, maillist names, printer names, or print queue destinations. See mail glossary.

>maillist

You must prefix a maillist name with a ">" character

\printqdest

You must prefix a print queue destination with a "\" character

modifiers

One or more of the following codes:

C	send as a confidential letter.
M	send as a Must Reply letter.
Q	send quietly (no unsolicited message).
R	send as a registered letter.
U	send urgently (with unsolicited message).

Examples:

Send a confidential letter to TOM, DICK & HARRY.

```
SC TOM,DICK,HARRY
```

Send a must-reply letter to everyone in the maillist named ALL.

```
SM >ALL
```

Send a letter to the ARC print queue destination.

```
S \ARC
```

Additional Considerations:

The sender of a letter will receive a copy of the letter if:

- he is included in the distribution list, or
- the "always receive a copy" Change command flag is set, or
- he is in one or more of the maillists being sent to and the "remove from maillists" Change command flag is NO.

Multiple occurrences of a user id (in the distribution list or in maillists) will be detected and deleted.

Function key 2 (**F2**) may be used to page forward through the letter.

Function key 3 (**F3**) may be used to display (and alter) the distribution list that will be used to send the letter.

U - Compose Letter with Editor - Not Implemented

This command is used to compose the text of a letter using a TIP/ix text editor.

Syntax:

```
U letter# [,distribution]
U edit/buff [,distribution]
```

Where:

letter#

The number of a letter to be modified via the text editor.

distribution

List of user ids, maillist names, printer names, or print queue destinations. See mail glossary.

edit/buff

The name of a TIP/ix edit buffer that contains text that is to be mailed.

Default value for "edit" is the first optional group of the user who invoked mail.

Example:

Invoke MAILEDT to allow modification of letter 1 and send the (modified) letter to TOM, DICK & HARRY.

```
U 1 , TOM , DICK , HARRY
```

Additional Considerations:

The sender of a letter will receive a copy of the letter if:

- he is included in the distribution list
- the "always receive a copy" Change command flag is set

- he is in one or more of the maillists being sent to and the "do NOT remove from maillists" Change command flag is set

Multiple occurrences of a user id (in the distribution list or in maillists) will be detected and deleted.

mail will invoke the transaction MAILEDT which by default executes the TIP/ix Full Screen Editor (**fse**).

If **fse** is used, then setting the Case to "L" and the update stamp to "NONE" is recommended (see the SE command in the documentation for **fse**)

Error Conditions:

One or more of the specified user ids may not be valid.

US - List user ids

This command invokes the TIP/ix utility transaction named "USERS". This can be used to verify the spelling of a user id.

Syntax:

USers

Mail Programming Guide

This section contains information intended for the programmer who intends to call the TIP/ix Mail system from on online TIP/ix program.

Retrieving Mailbox Counts

An online program can determine the number of letters that are currently in a user's mailbox by placing the appropriate data in the CDA and invoking **mail** via a call to TIPSUB.

Example:

```
MOVE 'MAIL $$$' TO CDA-PARAM (1) .  
MOVE 'COUNT ' TO CDA-PARAM (2) .  
MOVE user id TO CDA-PARAM (3) .  
MOVE 'MAIL' TO PIB-TRID .  
CALL 'TIPSUB' .
```

If **mail** is successfully invoked, it returns the following information in the CDA fields indicated:

CDA-PARAM (1)

The number of New letters in the mailbox.

CDA-PARAM (2)

The number of old letters in the mailbox.

CDA-PARAM (6)

The number of Memo letters in the mailbox.

CDA-PARAM (7)

The number of Pending letters in the mailbox.

CDA-PARAM (8)

The earliest date when 1 or more pending letters will "appear" in the mailbox.

CDA-TEXT

The result string '.....COUNT was completed successfully (leading period characters in the string represent spaces).

If **mail** cannot perform the requested function it returns:

CDA-TEXT

The result string '.....COUNT could NOT be performed (leading period characters in the string represent spaces).

Send Mail from an Online Program

An online program may send mail by:

including the copy module TC-MAIL immediately after the standard copy book TC-CDA

moving the appropriate data in the CDA (as defined in TC-MAIL and TC-CDA)

Invoking **mail** via a call to TIPSUB

Example:

```
MOVE 'MAIL $$$'      TO CDA-PARAM (1) .
MOVE 'SEND '        TO CDA-PARAM (2) .
MOVE mail data      TO various TC-MAIL fields
MOVE 'MAIL'         TO PIB-TRID .
CALL 'TIPSUB' .
```

If **mail** is successfully invoked, it returns the result string '.....SEND was completed successfully in CDA-TEXT (the leading period characters represent space characters).

If **mail** cannot perform the requested function, it returns the result string '.....SEND could NOT be performed in CDA-TEXT (the leading period characters represent space characters).

Additional Considerations:

Even though the TC-MAIL copy module only defines 20 lines of letter data, **mail** will accept up to 100 lines for a letter. As shown below, COPY

REPLACING may be used to modify the size of the MAIL-LINES array without permanently affecting the copy module.

The TIP/ix catalogue records for **mail** and for the program that is calling **mail** must have their CDA sizes adjusted appropriately, if more than 20 lines of letter data are to be sent.

```

01  CDA.                                COPY TC-CDA OF TIP.
    05 MAIL-CDA                          COPY TC-MAIL OF TIP
    REPLACING ==OCCURS 20== BY ==OCCURS 100==.
*
* LAYOUT OF CDA AREA RE SENDING MAIL VIA AN ONLINE PROGRAM
*
    10 MAIL-HEADER-DATA.
        15 FILLER                          PICTURE X.
        15 MAIL-REGISTRATION                PICTURE X.
            88 MAIL-REGISTERED              VALUE 'Y'.
        15 MAIL-CONFIDENTIALITY            PICTURE X.
            88 MAIL-CONFIDENTIAL            VALUE 'Y'.
        15 FILLER                          PICTURE 9(3) COMP-3.
        15 MAIL-LINE-COUNT                  PICTURE 9(5) COMP-3.
        15 MAIL-AUTHOR                      PICTURE X(8).
        15 MAIL-SUBJEC                      PICTURE X(40).
        15 MAIL-DATE                        PICTURE 9(6).
        15 FILLER                          PICTURE 9(4).
        15 MAIL-MUST-REPLY-FLAG            PICTURE X.
            88 MAIL-MUST-REPLY              VALUE 'Y'.
        15 FILLER                          PICTURE X(216).
        15 MAIL-SEND-QUIETLY-FLAG          PICTURE X.
            88 MAIL-SEND-QUIETLY            VALUE 'Y'.
        15 MAIL-DISTRIBUTION-LIST.
            20 MAIL-DISTRIBUTE              PICTURE X(8)
                                           OCCURS 28 TIMES
                                           INDEXED BY I-MAIL-DIST.
    10 MAIL-LETTER-DATA.
        15 MAIL-LINES                       PICTURE X(80)
                                           OCCURS 100 TIMES
                                           INDEXED BY MAIL-LINE-INDEX.
    
```

maillist - TIP/ix Mail List System

The **maillist** utility transaction provides maintenance services for MAIL system user name lists (also referred to as "mail lists"). Mail lists may be used in the TIP/ix MAIL system to send mail to a named list of TIP/ix users (see the documentation of the [MAIL](#) transaction in this manual.)

A mail list consists of a number of TIP/ix user ids, maillist names, or printer names, in any combination. To distinguish the specification of a user id from a mail list name, the MAIL program (and this **maillist** utility) restricts mail list names to a maximum of seven characters and assumes that any reference to a mail list name is preceded by a greater than character, for example: >DP.

The user who first creates a particular mail list is considered the default steward of that mail list. If that user desires, the stewardship of that mail list can be changed to any TIP/ix group name or prefix specification to permit users who are members of the specified group to alter the members of the mail list (or to further change the steward).

Syntax:

```
maillist [command]
```

Where:

command

Optional command and associated parameters to be executed.

Example:

This example command line invokes **maillist** to "inquire" (display) the mail list named "DP".

```
maillist I DP
```

If **maillist** is invoked with command line parameters, it performs the requested function and terminates without interacting with the terminal.

If **maillist** is invoked without parameters, it performs an automatic Directory display of all mail lists (see description of the **maillist** command "DIR") and prompts for a command.

Command Summary

maillist recognizes the following commands:

Command	Description
Add	Create a new mail list.
Change	Change a mail list.
DELeTe	Delete a mail list.
DIRectry	Directory of mail lists.
End	End the maillist program.
Find	Find all references to a user id, maillist name, printer name, or print queue destination.
Inquire	Inquire about a mail list.
Match	Compare two mail lists.
Print	Print a mail list.
REMove	Remove all references to a user id, maillist name, printer name, or print queue destination.

REName	Rename a mail list.
REStore	Restore a mail list from a file.
Save	Save a mail list in a file.

A - Add Mail List

The "A" command is used to create a new mail list. An existing mail list may be used as a template for the new mail list. The user that creates a mail list is automatically considered the steward of that mail list. The steward specification can be changed as part of the "Add" operation, or may be altered later by using the "Change" **maillist** command.

Syntax:

```
Add name [,oldname]
```

Where:

name The name of the mail list to be added (a required parameter).

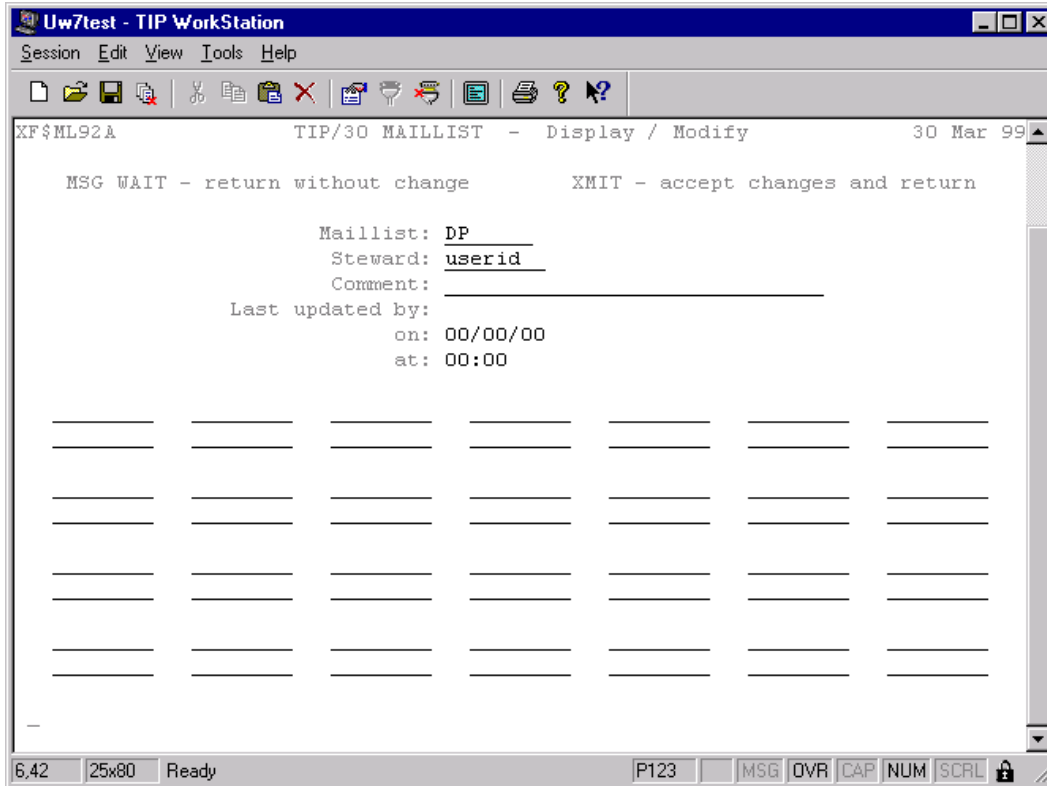
oldname Optional name of an existing mail list to be used as a template for the new mail list.

Example:

Create a mail list called "DP".

```
Add DP
```

The **maillist** program displays the following screen format to permit the user to enter TIP/ix user ids or other mail list names to be included in the definition of "DP". If an existing mail list is specified (to use as a template), its contents are included in the screen format as initial data.



Where:

Steward

The name of a user id or group to be considered the "steward" of this mail list. Any TIP/ix user who belongs to this group is granted change access to this mail list.

Standard prefix notation may be used in this entry. Note that a value of "TIP\$Y\$" implies unrestricted access.

The "Add" command automatically places the user id of the person who is adding this mail list in this field as the default value.

Comment

A descriptive comment may be placed in this field. The comment is displayed by the maillist "Dir" command.

Error Conditions:

The mail list being created may already exist, or the mail list to be used as a template does not exist. Each user id or mail list name entered is checked for validity. An appropriate error message is issued if the user id is not present in the TIP/ix Catalogue or if a mail list name is not recognized.

C - Change Mail List

The "C" command is used to modify an existing mail list.

Syntax:

Change name

Where:

name The name of an existing mail list to be modified.

Example:

Display the mail list called EDPLIST and allow it to be modified.

```
C EDPLIST
```

Additional Considerations:

Only users who are TECH level security or belong to the user group specified as the "steward" for a mail list may modify that mail list.

Error Conditions:

The requested mail list may not exist or cannot be modified by the user.

DEL - Delete Mail List

The "DEL" command is used to delete an existing mail list.

Syntax:

DELeTe name

Where:

name The name of an existing mail list to be deleted.

Example:

Delete the mail list named "DP".

```
DEL DP
```

Additional Considerations:

Only users who are TECH level security or belong to the user group specified as the "steward" for a mail list may delete that mail list.

Error Conditions:

The requested mail list may not exist or may not be deleted by the user.

DIR - Display Mail List Directory

The "DIR" command is used to display summary information about selected mail lists.

Syntax:

```
DIR [name]
```

Where:

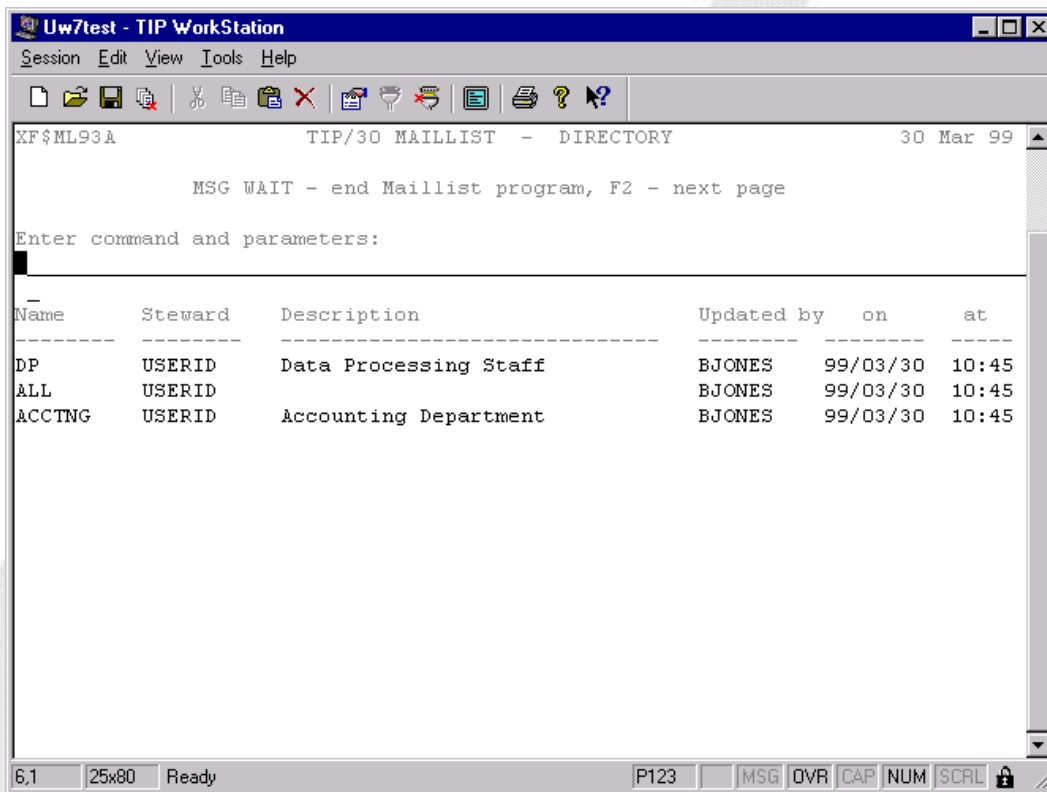
name The name of the mail list to be summarized. Standard TIP/ix prefix notation may be used. Default is all mail lists.

Example:

Display a summary list of all mail lists.

```
DIR
```

A screen such as the following is displayed. The screen contains an area where a **maillist** command can be entered.



Error Conditions:

The specified mail list name may not exist.

E - End Program

The "E" command is used to terminate the **maillist** utility.

Syntax:

End

The **maillist** program accepts **MSG WAIT** as a signal to terminate the program.

F - Find References

The "F" command is used to find all occurrences of a user id (or a mail list name) in all existing mail lists.

Syntax:

Find name

Where:

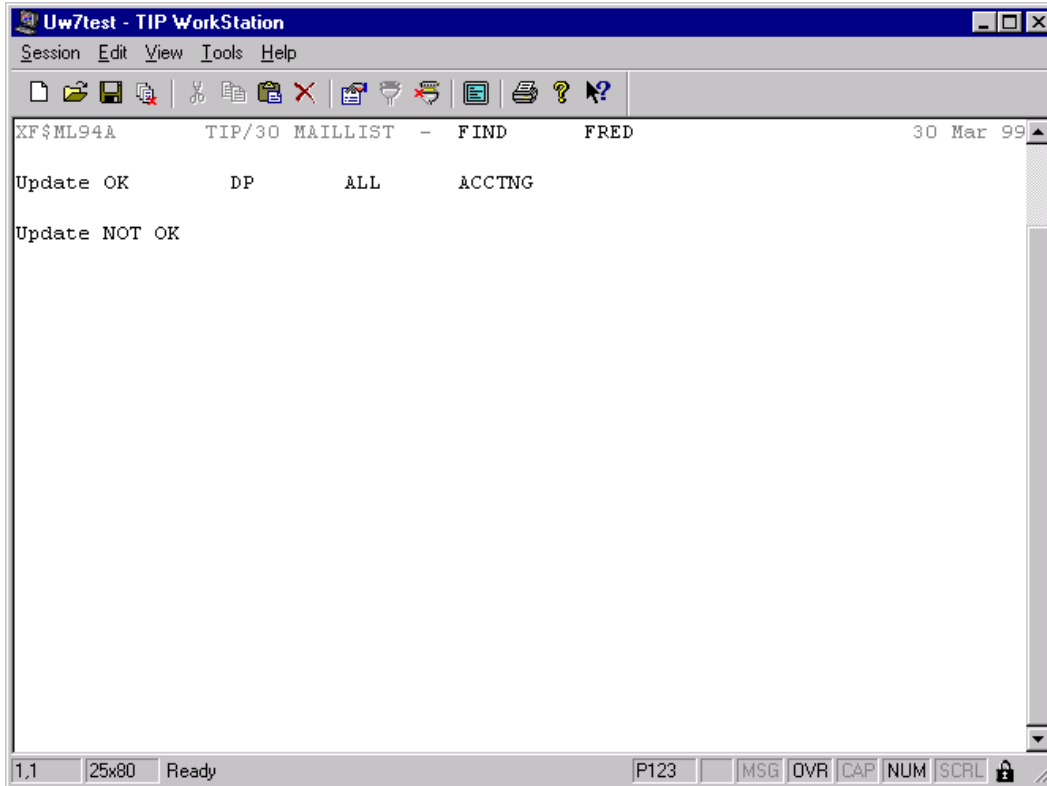
name A user id name, maillist name, printer name, or print queue destination to be found. If references to a mail list are to be found, the mail list name must be preceded by a greater than symbol (>) to distinguish the mail list name from a user name that is spelled the same.

Example:

Display all mail lists that contain "FRED".

F FRED

The "F" command results in a display similar to the following:



Additional Considerations:

When the user has finished viewing this screen, press **MSG WAIT** to return to a "Dir" display and a command entry field.

I - Inquire Mail List

The "I" command is used to examine an existing mail list.

Syntax:

Inquire name

Where:

name The name of an existing mail list to be examined.

Example:

Display the mail list named "EDPLIST". The mail list is displayed using the same screen format shown in the description of the **maillist** "Add" command.

I EDPLIST

Error Conditions:

The requested mail list may not exist.

M - Compare Mail Lists

The "M" command (match) is used to compare the contents of two mail lists. A summary is produced showing the unique names in each list and the names that appear in both lists.

Syntax:

```
Match name 1 name 2
```

Where:

name 1

The first mail list name to be compared.

name 2

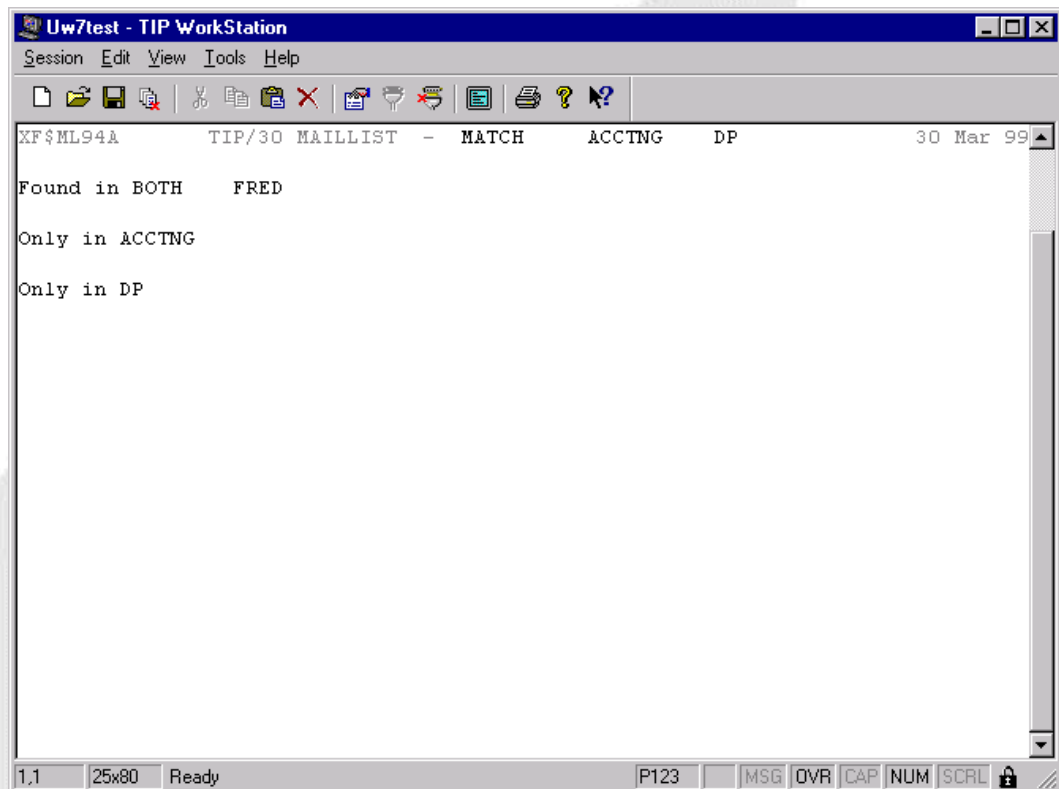
The second mail list name to be compared.

Example:

Display a comparison of the two mail lists "DP" and "ACCTNG".

```
M DP ACCTNG
```

The "M" command results in a display similar to the following:



Additional Considerations:

When the user has finished viewing this screen, press **MSG WAIT** to return to a "Dir" display and a command entry field.

P - Print Mail List

The "P" command is used print a hard copy report of a mail list at the main site printer, a terminal auxiliary printer or any printer destination recognized by TIPPRINT.

Syntax:

```
Print name [,printer] [,case]
```

Where:

name The name of an existing maillist to be printed.

printer

The name of the destination printer (default is PRNTR; another example is: AUX1). Any printer destination recognized by TIPPRINT may be specified.

case A choice between "Upper" and "Lower" indicating the desired case of the printout. "Upper" is the default when printing on the main site printer; "Lower" is the default when printing on an auxiliary printer.

Example:

Produce a hard copy printout of the maillist "EDPLIST" on the auxiliary printer attached to the issuing terminal (print in upper case).

```
PR EDPLIST ,AUX1 ,U
```

Error Conditions:

The requested mail list may not exist.

REM - Remove Name from Mail List

The "REM" command (remove) is used to find and remove all references to a name in existing mail lists. A user id or a mail list name can be removed.

Syntax:

```
REMove name
```

Where:

name The name of a user id, maillist name, printer name, or print queue destination that is to be removed from all mail lists.

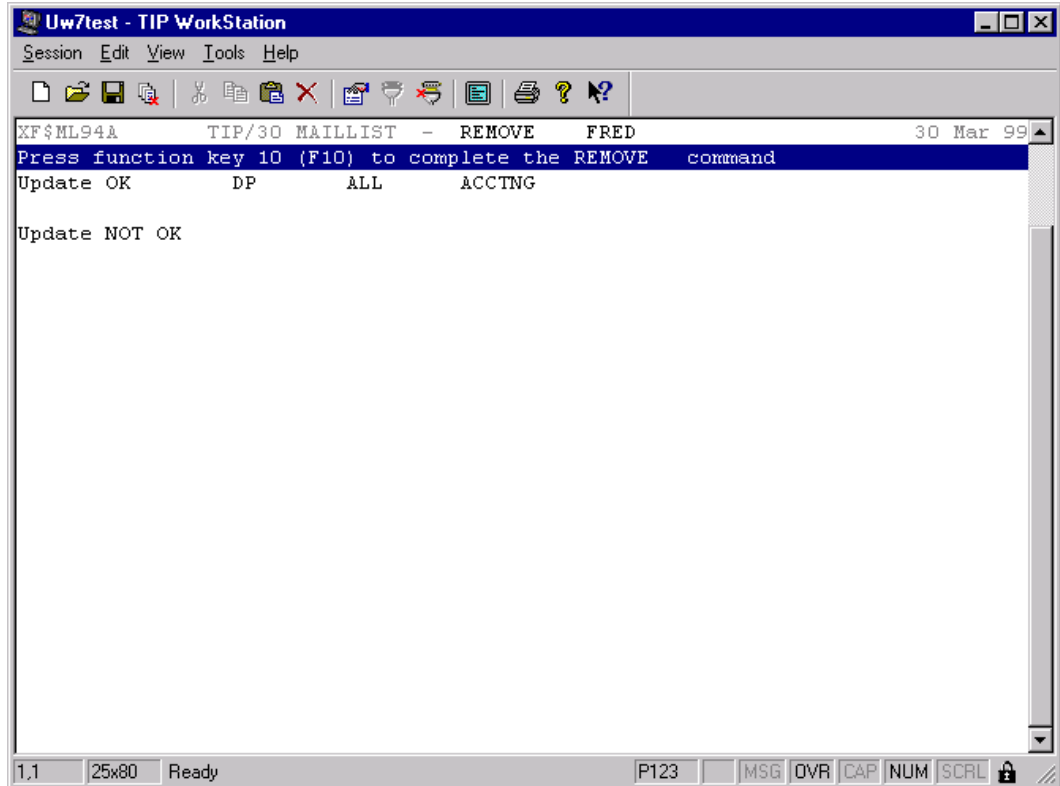
If a mail list name is intended, the mail list name must be preceded by a greater-than symbol (>) otherwise the maillist program assumes the value represents a user id.

Example:

Remove all occurrences of "FRED" from mail lists.

REM FRED

The "REM" command results in a display similar to the following:



When the user has finished viewing this screen, press **MSG WAIT** to return to a "Dir" display and a command entry field.

Additional Considerations:

Only users who are TECH level security or belong to the user group specified as the "steward" for a mail list may remove names from that mail list.

REN - Rename Mail List

The "REN" command is used to rename an existing mail list.

Syntax:

REName name , new name

Where:

name The name of an existing mail list to be renamed.

new name

The new name of the mail list.

Example:

Change the name of mail list "DATAList" to "EDPLIST".

```
REN DATAList,EDPLIST
```

Additional Considerations:

Only users who are TECH level security or belong to the user group specified as the "steward" for a mail list may rename that mail list.

Error Conditions:

The requested mail list may not exist, or a mail list with the new name may already exist.

RES - Restore Mail List - Not Implemented

The "RES" command is used to restore a maillist that was previously saved by **maillist** in a file.

Syntax:

```
REStore name,file
```

Where:

name The name of the mail list to be restored.

file The file name where the mail list was saved.

Example:

Restore the maillist EDPLIST, which was saved in the file "EDPSAVE", to the **maillist** file.

```
RES EDPLIST EDPSAVE
```

Error Conditions:

A mail list with the name specified already exists.

The library or the element name is not found.

The library element is not in the format required by the restore command.

S - Save a Mail List - Not Implemented

The "S" command is used to save a mail list in a file.

Syntax:

```
Save name,file
```

Where:

name The name of an existing mail list to be saved.

file The file name where the mail list is to be saved. Default is the name of the mail list.

Example:

Save the mail list named "EDPLIST" as the file named "EDPSAVE".

S EDPLIST EDPSAVE

Error Conditions:

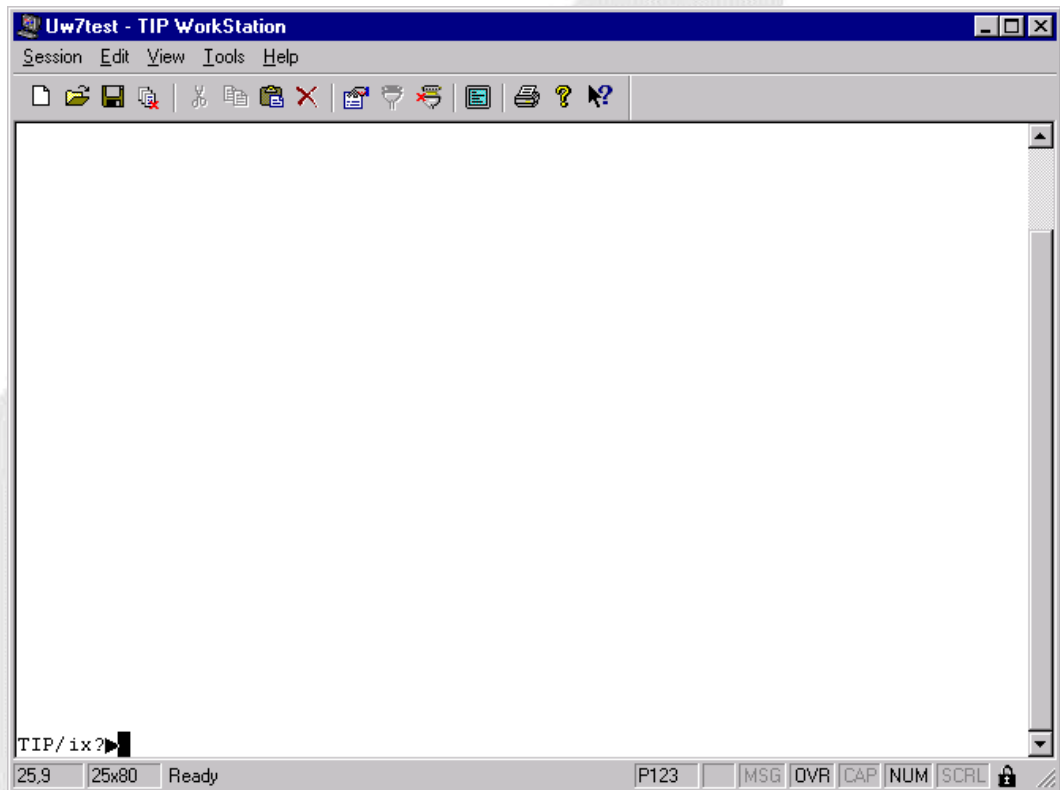
The requested mail list does not exist.

The file name may be unacceptable (cannot be created for example).

menu - TIP/ix Menu System

The TIP/ix MENU SYSTEM provides an alternative to the standard command line interface for the online system.

Instead of the standard system command line prompt:



the user is prompted by a numbered list of items, from which any item may be selected by keying its number in the field provided.

The command line may still be used by a more knowledgeable user (one who is familiar with a transaction and its parameter requirements).

The menu system makes it possible to implement an online system with less initial training. Memorization of transaction names and functions is not required and detailed online HELP information can be created for any or all of the items on any menu.

Each item on a menu can initiate any of the following actions:

- Invoke a subordinate menu
- Call a TIP/ix program with canned and/or user supplied data
- Call a TQL program with canned and/or user supplied data
- Start a batch job with canned and/or user supplied data
- End menu processing
- Return to the previous menu
- Logoff TIP/ix.

Menu System Requirements

The following table shows the transactions used to create and maintain menus online:

Transaction	Description
menu	MENU processor.
menu\$\$	Called (via TIPSUB) by menu for menu commands
menudef	Creating and updating menus.
menuar	Menu maintenance (save, print, rename, delete ...).

If the menu system is used with IMS programs that do not do succession it may be necessary to change the catalogue entry for the **menu** program to CMDLINE=NO. This prevents the TIP/ix prompt from rolling the screen up from the bottom at normal termination (causing the top line to be lost). This is not a problem when IMS succession is used.

Menu Processor

The Menu Processor is that part of the TIP/ix Menu System that prompts the user with menus and processes his requests. It may be invoked by entering the transaction name **menu** in response to the TIP/ix prompt. The menu program can be invoked automatically for a user by setting the user's Startup Program to menu (see [smuser](#) to see how to set a user's Startup Program).

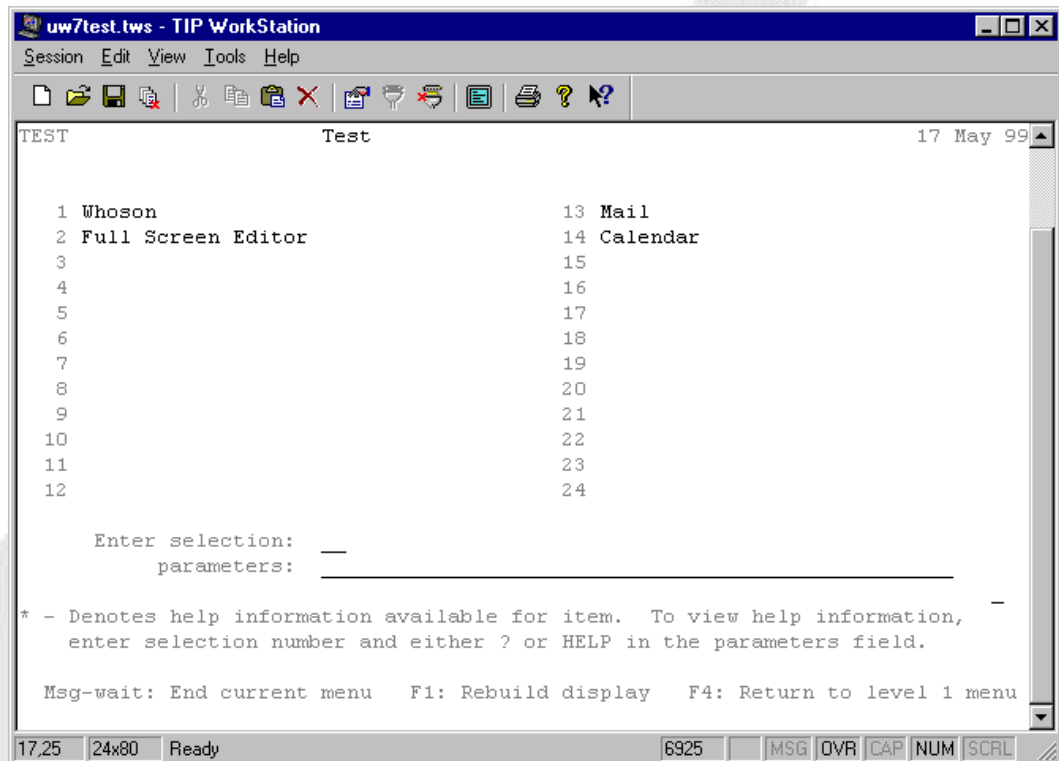
The first action of the **menu** program is to search for the appropriate menu to display. The **menu** program searches for the menu to display by following the standard order of search. This means that it looks for a menu with the same name as the user id, then it looks for a menu with the same name as one of the elective groups that the user has access to, and finally it looks for a menu with the name **TIP\$Y\$**. The first menu found that matches one of these criteria is the one that is displayed. This technique ensures that first level menus will be valid user id or group names.

Users with programmer security or higher may explicitly select a specific menu by entering a menu name as the first parameter to the **menu** transaction. This permits these users to access any menu for testing or other purposes.

►**menu MYMENU**

Prompts the user with the menu MYMENU.

When **menu** has located a menu it prompts the user with a screen format such as this:



An advertised item may be selected by keying its number in the "selection" field, and entering any necessary parameters in the "parameters" field.

The environment variable TIPMENUKEY may be set to the value "Y" to interchange the meaning of **F1** and **MSG-WAIT** for the run-time **menu** system. If this is done, the screen format may need to be altered to reflect

this preference (the screen format as shown above is the standard screen as shipped with TIP/ix).

The standard delimiters slash, comma, and any number of spaces may be used in the "parameters" field.

After pressing **XMIT**, **menu** processes the user's request. If the request calls an online program the user interacts with the program until the program relinquishes control. At this point, control normally returns to the menu from which the program was called. In some cases, however, the **menu** system may output a "pause" message requiring the terminal operator press **MSG WAIT** or any other function key to return to the menu.

If the user selects an item which calls another menu, he works from that menu as he would from the first menu. By pressing **MSG WAIT** the user indicates that he is finished with the current menu and wishes to return to the menu that called it. Pressing **MSG WAIT** from the main menu terminates menu processing.

When a menu is created, items can be associated with special functions that enable the user to logoff, terminate menu processing, or return to the previous menu.

Error Conditions:

Invalid selection

No item corresponds to the number given.

Required parameter(s) missing

Some necessary parameters were not provided. If the menu item has a help screen this message will be displayed from the help screen.

Additional Considerations:

You may now catalogue a transaction to retrieve a menu with the same name as the transaction ID (TRID). For example, TRID=DOUG1 would have a [SMPROG](#) entry similar to the entry for **menu**.

If the specified menu is not found, **menu** searches through the standard order of search for the first menu name matching the user's GROUPs. Failing the above, the following error message is issued:

You do not have any menus.

Menu Parameter Processing

The **menu** system allows user supplied positional parameters to be specified when a menu item is selected. **menu** users can enter up to 50 bytes of information when selecting a menu item.

When a menu item is defined, user supplied parameters can be designated by the variables &1 through &8, where the number denotes the position of the parameter. The variable &0 represents the entire line entered by the user with trailing spaces removed.

A character string enclosed in parentheses immediately following a variable serves as a default value. If an exclamation mark immediately follows a variable name, it indicates that the variable is required. The menu item is not invoked unless the user supplies the variable.

Built in parameters provided by the menu system may be used on the command line:

- &U** The TIP/ix user id of the menu user
- &T** The terminal name where the menu is being used
- &Y** The current date in the form YYMMDD
- &D** The current date in the form DDMMYY
- &M** The current date in the form MMDDYY

There are three special menu commands:

RETURN

Ends processing of current menu and returns to previous menu.

END End menu processing. Terminates current and all prior menus.

LOGOFF

Ends menu processing and logs user off the TIP/ix system.

Example of using Special Parameters:**FSE TSTSRC , &1 ! , &2 (M)**

The menu user is required to enter positional parameter 1 which is the name of the element to edit from the library TSTSRC. If positional parameter 2 is given it indicates the element type. Otherwise the type defaults to "M".

TQL &1

Invoke the TQL program specified by the user. If the user does not specify a program, the TQL transaction is invoked without parameters.

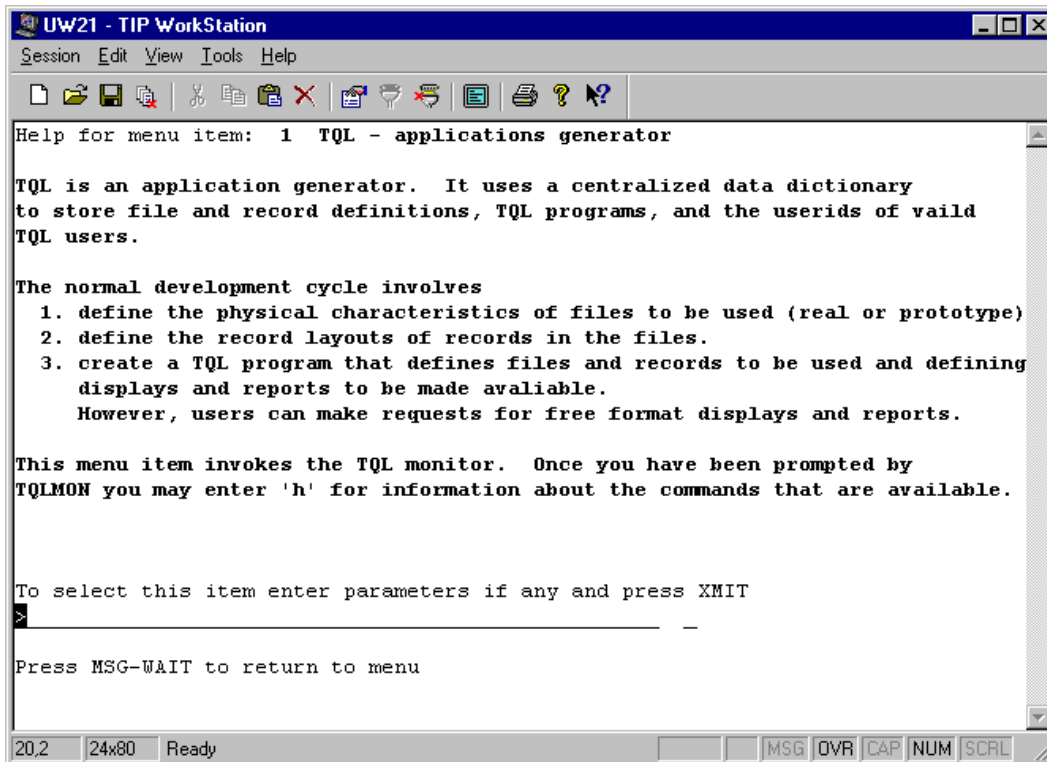
Menu Help Requests

You can request help information about a menu item by entering its number in the selection field and either the word "help" or a "?" in the

parameter field. **menu** informs you that there is available HELP for an item by displaying an "*" before the item number.

A help screen will be displayed if one was defined for this item (see how to create HELP information in the description of the [menudef](#) utility transaction).

There is a standard screen format used to display help information. The following example shows the format of the help screen:



```

UW21 - TIP WorkStation
Session Edit View Tools Help
Help for menu item: 1  TQL - applications generator

TQL is an application generator.  It uses a centralized data dictionary
to store file and record definitions, TQL programs, and the usersids of valid
TQL users.

The normal development cycle involves
  1. define the physical characteristics of files to be used (real or prototype)
  2. define the record layouts of records in the files.
  3. create a TQL program that defines files and records to be used and defining
    displays and reports to be made available.
    However, users can make requests for free format displays and reports.

This menu item invokes the TQL monitor.  Once you have been prompted by
TQLMON you may enter 'h' for information about the commands that are available.

To select this item enter parameters if any and press XMIT
_____

Press MSG-WAIT to return to menu
  
```

Help screens can be very useful for describing any parameters that are required for a menu item. They also serve to describe the functionality of a menu item. After reading the help information you can select the item described by entering the parameters for the item (if any) in the space provided and pressing the **XMIT** key. If you do not wish to invoke this item you return to the menu by pressing the **MSG-WAIT** key.

If required parameters are not entered for a menu item with a help screen then the help screen is automatically displayed.

Error Conditions:

If no help information was defined for the item the user receives the message "Help not available".

menuar - Menu Maintenance

The **menuar** utility transaction provides librarian and maintenance services for menus created with the TIP/ix Menu System. To use this program or any other program in the menu system you must have generated TIP/ix with a menu file.

menuar recognizes the following commands:

Command	Description
Copy	Create a copy of a menu.
DElete	Delete a menu.
DIrectory	Print a directory of menu names and information.
End	End interaction with menuar program.
HDElete	Delete help for a menu item.
Help	Display help information on terminal.
HScreen	Create or update help for a menu item.
List	List menu names and information.
Print	Print a menu and its help screens.
QPrint	Quick print of a menu (no help screens).
Quit	End interaction with menuar program and logoff.
REName	Rename a menu.
TImeout	Change the timeout value of a menu.

menuar may be used interactively or may be given a single command on the command line. If a single command is given on the command line, **menuar** attempts only that single command and then terminates normally. When used interactively, **menuar** prompts the user for each command.

C - Copy a Menu

This command will make a copy of a menu and its help screens.

Syntax:

Copy source target

Where:

source

The name of the menu to be copied (prefix specification is not allowed).

target

The name of the menu to be created.

Example:

Create a copy (called "testmenu") of the menu named "yourmenu".

```
C yourmenu testmenu
```

Error Conditions:

The source menu may not exist or the target menu may already exist.

DEL - Delete a Menu

This command will delete a menu and its help screens.

Syntax:

```
DELeTe name
```

Where:

name The name of the menu to be deleted (prefix specification is not allowed).

Example:

Delete the menu named "YOURMENU".

```
DEL yourmenu
```

Error Conditions:

The specified menu may not be found.

DI - Display Menu Directory

This command produces a printout containing information known about the selected menus. The information printed includes: menu name, author, date and time created.

Syntax:

```
DIRECTory *name [,printer]
```

Where:***name**

A single menu name or a prefix specification

printer

The output printer destination. The default destination is PRNTR (the site printer). The printer may also be specified as an auxiliary print device.

Example:

Produce a directory listing of all menus which have a name not starting with the string "TEST". The output is to be directed to the auxiliary printer for the issuing terminal.

```
DIR !test,aux1
```

E - End Program

This command causes the **menuar** program to terminate normally.

Syntax:

```
End
```

H - Display Help Information

This command causes the **menuar** program to display a summary of recognized commands and required parameter syntax.

Syntax:

```
Help
```

HDEL - Delete a Help Screen

This command is used to delete all help screens for a menu or to delete a help screen for a particular menu item.

Syntax:

```
HDElete menu name item
```

Where:**menu name**

The name of a menu

item The number of an item on the menu. The help screen for this menu item will be deleted. If the item is specified as '*' all help screens for the menu will be deleted.

Example:

Delete the help screen for item 5 of the menu "EDP".

```
HDEL edp,5
```

HS - Define Help Screen

This command allows you to create or update a help screen for a menu item. If you do not specify the item number you are prompted with a screen displaying the menu's items. You then select the item you wish to define a help screen for. The next prompt is a menu that allows you to enter up to 15 lines of help information. After entering the help information press the **XMIT** key. To cancel the definition press **MSG WAIT**.

Syntax:

```
HScreen menu name item
```

Where:

menu name

The name of a menu for which a help screen is to be defined.

item The item number that the user is defining a help screen for.

Example:

Define a help screen for item 4 of the menu "EDP".

```
HS edp 4
```

You are prompted with a display from which you enter the help information. If item 4 already has a help screen, the old help is displayed and you may modify it as desired.

L - List Information about Menu

This command displays (on the terminal) a list of selected date and time they were created.

Syntax:

```
List *name
```

Where:

***name**

A single menu name or a prefix specification

Example:

Produce a listing of all menus which have a name starting with the string "PAY".

```
LIST *pay
```

P - Print Menu

This command prints the specified menus. For each menu selected this includes the menu as it appears to the user followed by an action table indicating the commands associated with each menu item. The help screens for each menu will also be printed. The output may be routed to the site printer (default destination) or to an auxiliary print device (for example: AUX1).

Syntax:

```
Print *name [,printer] [,case]
```

Where:

***name**

A single menu name or a prefix specification

printer

The name of the destination printer (default is PRNTR; another examples is: AUX1).

case

A choice between "Upper" and "Lower" indicating the desired case of the printout. "Upper" is the default when the destination is the site printer; "Lower" is the default when the destination is an auxiliary printer.

Example:

Produce a hard copy printout of all menus with a name starting with the string "TEST" on the site printer and attempt to print lower case data. The listing includes help screens for the selected menus.

```
PRINT *test,,LOWER
```

QP - Quick Print of Menu

This command prints selected menus but not their help screens. Each menu is printed as it appears to the user followed by a table indicating the commands associated with each menu item. The output may be routed to the site printer (default destination) or to an auxiliary print device (for example: AUX1).

Syntax:

```
QPrint *name [,printer] [,case]
```

Where:

***name**

A single menu name or a prefix specification

printer

The name of the destination printer (default is PRNTR; another example is: AUX1).

case A choice between "Upper" and "Lower" indicating the desired case of the printout. "Upper" is the default when the destination is the site printer; "Lower" is the default when the destination is an auxiliary printer.

Example:

Produce a hard copy printout of all menus with a name starting with the string "TEST" on the site printer and attempt to print lower case data. Help screens for the selected menus are not printed.

```
QP *test, ,LOWER
```

Q - End and Logoff

This command will end the **menuar** program. If the user was executing the menu archiver at stack level 1 (that is: the menu archiver was NOT called from another program) then the user is logged off the TIP/ix system.

Syntax:

```
Quit
```

REN - Rename a Menu

This command renames an existing menu. The new name must not currently be in use.

Syntax:

```
REName name, new name
```

Where:

name The name of an existing menu

new name

The desired new name for the menu

Example:

Change the name of menu "MYMENU" to "YOURMENU".

```
REN mymenu, yourmenu
```

TI - Specify Timeout Value

This command allows the user to specify a new value to be used as the timeout for a specific menu. The timeout value **may** be specified when a menu is initially defined; if a different value is desired at a later time, this **menuar** command may be used.

Syntax:

```
Timeout name [seconds]
```

Where:

name The name of an existing menu

seconds

The number seconds to be used as the timeout value for this menu.

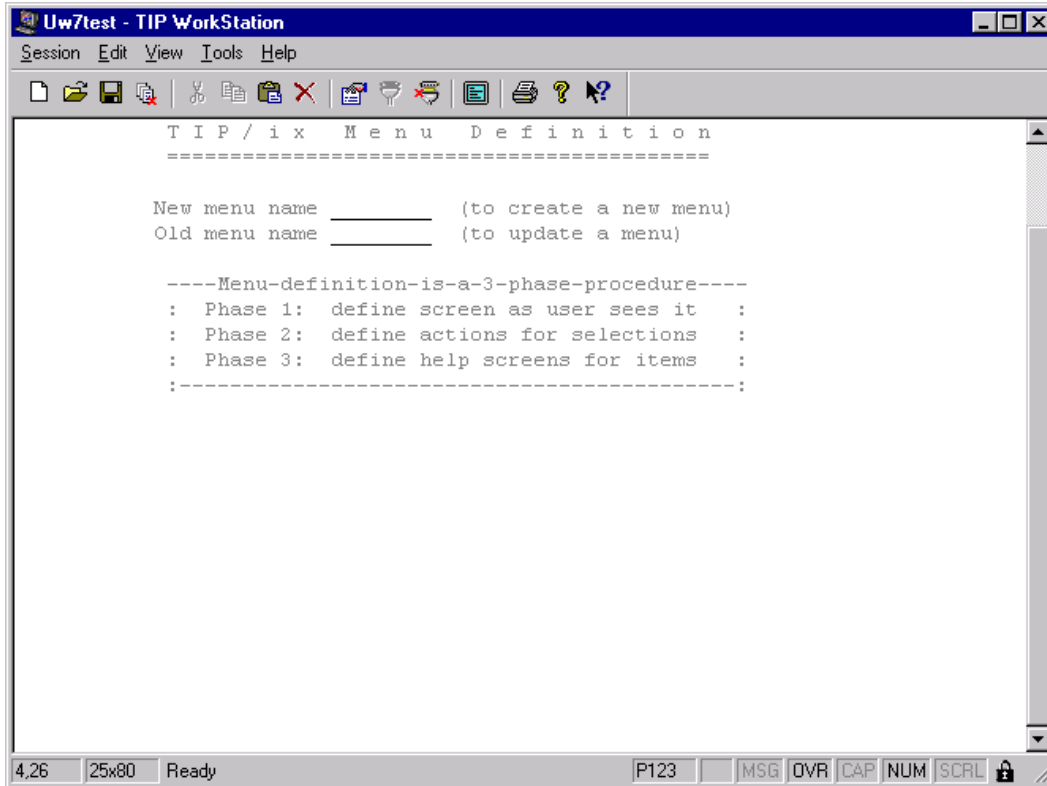
menudef - Menu Definition

This program is used to create and update menus for the TIP/ix Menu System. This can only be used if the TIP/ix system has been generated with a menu file.

There are three phases to creating a menu:

- Define the menu as seen by the user
- Associate each menu item with a particular action
- Define help information for menu items.

Upon invoking **menudef** the user is prompted with the following screen:



To create a new menu, enter the desired name in the "New menu" field and leave the "Old menu" field blank. To modify an existing menu enter its name in "Old menu" field and leave "New menu" blank. To make a new menu from an existing one, fill in both fields.

The user can skip this prompt screen by providing the parameters on the initial command line as follows:

Syntax:

```
menudef [,h] new-menu,old-menu
```

Where:

new-menu

is the name of a new menu to be created

This field can be left blank if you are updating an existing menu

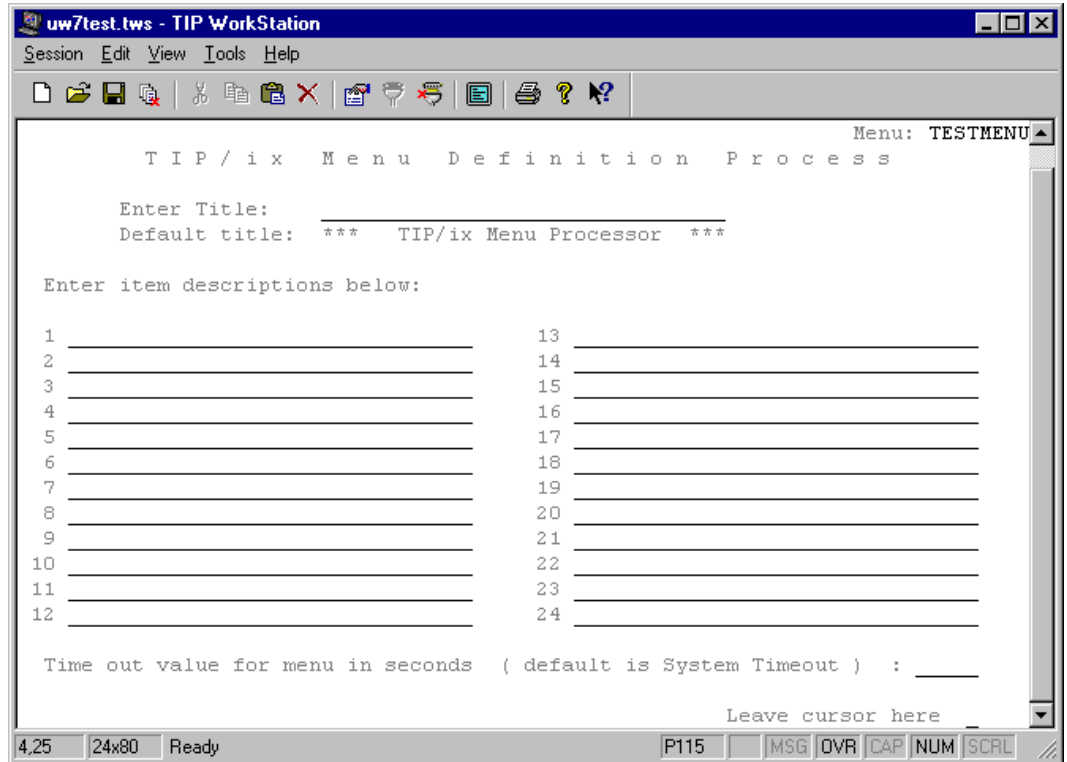
old-menu

is the name of an existing menu that is to be updated, or if new-menu is supplied then old-menu is used as the template for the new-menu

[,h] Specify this option to skip the first two phases and go directly to the third phase.

Phase 1 of Menu Definition

menudef prompts with the following screen so that you may define the menu as it is to be seen by the user:



Enter the menu title in the first field (if the default title is not desired).

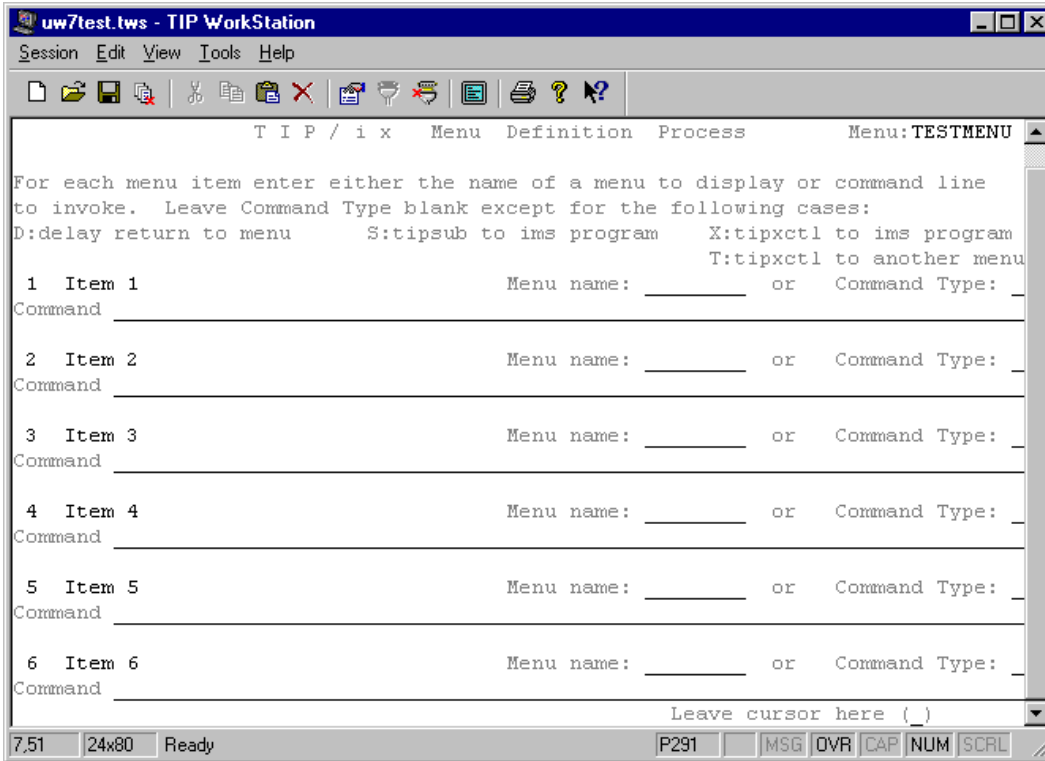
Enter item descriptions beside item numbers.

Specify a desired timeout value for this menu (at execution time the user is LOGGED OFF if a selection is not made within this number of seconds.)

Press **XMIT** from the cursor resting location provided. Pressing the **MSG WAIT** key at this point cancels the definition process.

Phase 2 of Menu Definition

This step defines the actions for each menu item. **menudef** prompts with the following screen:



uw7test.tws - TIP WorkStation

Session Edit View Tools Help

TIP / ix Menu Definition Process Menu: TESTMENU

For each menu item enter either the name of a menu to display or command line to invoke. Leave Command Type blank except for the following cases:
 D:delay return to menu S:tipsub to ims program X:tipxctl to ims program
 T:tipxctl to another menu

Item	Menu name	or	Command Type
1 Item 1 Command _____	_____	or	_____
2 Item 2 Command _____	_____	or	_____
3 Item 3 Command _____	_____	or	_____
4 Item 4 Command _____	_____	or	_____
5 Item 5 Command _____	_____	or	_____
6 Item 6 Command _____	_____	or	_____

Leave cursor here (_)

7,51 24x80 Ready P291 MSG DVR CAP NUM SCRL

menundef will continue to prompt with this screen until all actions have been defined for all items from phase 1. Actions can be entered for up to six items at a time. There are two lines for each item on this display. The first two fields for each item are the item number and description. This information is displayed by **menundef**. Fill in the remaining fields to define the processing required for each item.

Where:

Menu name

The name of a menu to be displayed when the menu item is selected. A menu item must either display a menu or invoke a command. If this field is left blank a command must be given.

Command

A TIP/ix command line to be invoked by the menu item. The command can be a TIP/ix program, an IMS program, or a special menu command. The command must be entered as it would be entered at the TIP/ix command line.

If the first character is a backslash, the MENU system will replace the backslash with a start of entry code (to allow the IMS program that is being invoked to observe an soe character in byte 1 of the IMA - some IMS programs are written to expect an soe character in that position).

Command Type

This field should be left blank if the menu item calls a menu

or invokes a TIP/ix program and the user should return to the menu immediately after the program terminates. This field must be used if the command invokes an IMS program.

- D** This type code indicates the command is a TIP/ix program and control should return to the menu only after some user acknowledgment. After the program terminates, the user returns to the menu only after pressing the MSG WAIT key or some function key. This technique is normally used if the TIP/ix program does some output to the terminal just before terminating. If a delayed return was not specified, there may not be time for the user to read the screen before the menu reappeared. An example of such a transaction program is the WMI transaction.
- S** This type code is used if the command is an IMS program and normal termination is only done on completion of processing. In this case MENU calls the IMS program by using the TIPSUB subroutine. However, special processing must be done to place the canned or user supplied parameters in the IMA of the program. When the IMS program ends (normal termination) the user returns to the menu after pressing XMIT the MSG WAIT key or any function key.
- X** This type code is used if the command is an IMS program but normal termination is done at intermediate points. In this case MENU calls the IMS program by using the TIPXCTL subroutine. However, special processing is done to place the canned and user supplied parameters in the IMA of the program. This type code should only be used when the user should not return to the menu after normal termination. Type "X" commands should only be used from a level 1 (main) menu. When processing is completed the user must key in the transaction code MENU to resume menu processing. Alternatively, the last program called could be altered to succeed to the transaction named "MENU\$\$" by either delayed or external succession with the CDA set to spaces.
- T** This type code is used if the menu name field contains the name of a menu that control is to "transfer to". In this case, the named menu is invoked with no implied means to return to this menu.

Error Conditions:

Possible error conditions include the following situations:

- Both a menu name and a command line were provided for a menu item. These fields are mutually exclusive.
- Both the menu name field and the type field were given for a menu item. These fields are mutually exclusive.
- Neither a menu name nor a command line was given for a menu item. A value for one of these fields must be entered.

Additional Considerations:

Whenever errors are detected the fields in question will blink and the user may then correct them.

Menu Parameter Processing

The MENU system allows user supplied positional parameters to be specified when a menu item is selected. MENU users can enter up to 50 bytes of information when selecting a menu item.

When a MENU item is defined, user supplied parameters can be designated by the variables &1 through &8, where the number denotes the position of the parameter. The variable &0 represents the entire line entered by the user with trailing spaces removed.

A character string enclosed in parentheses immediately following a variable serves as a default value. If an exclamation mark “!” immediately follows a variable name, it indicates that the variable is required. The menu item is not invoked unless the user supplies the variable.

Built in parameters provided by the menu system may be used on the command line:

- &U** The TIP/ix user id of the menu user
- &T** The terminal name where the menu is being used
- &Y** The current date in the form YYMMDD
- &D** The current date in the form DDMMYY
- &M** The current date in the form MMDDYY

There are three special menu commands:

RETURN

Ends processing of current menu and returns to previous menu.

- END** End menu processing. Terminates current and all prior menus.

LOGOFF

Ends menu processing and logs user off the TIP/ix system.

Example of using Special Parameters:

FSE TSTSRC, &1!, &2 (M)

The menu user is required to enter positional parameter 1 which is the name of the element to edit from the library TSTSRC. If positional parameter 2 is given it indicates the element type. Otherwise the type defaults to "M".

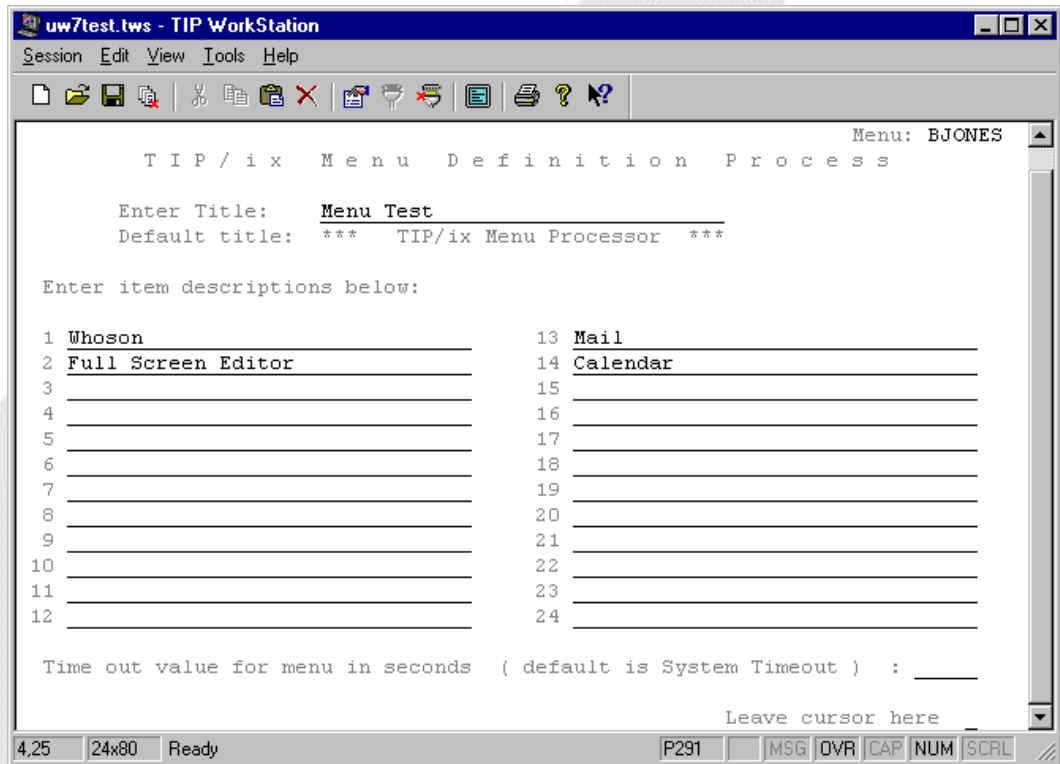
TQL &1

Invoke the TQL program specified by the user. If the user does not specify a program, the TQL transaction is invoked without parameters.

Phase 3 of Menu Definition

This phase is used to define help information for individual menu items. **menundef** prompts the user with the items defined for this menu. By selecting an item the user indicates that he wants to define help information for that item.

menundef issues a prompt like the following screen format:

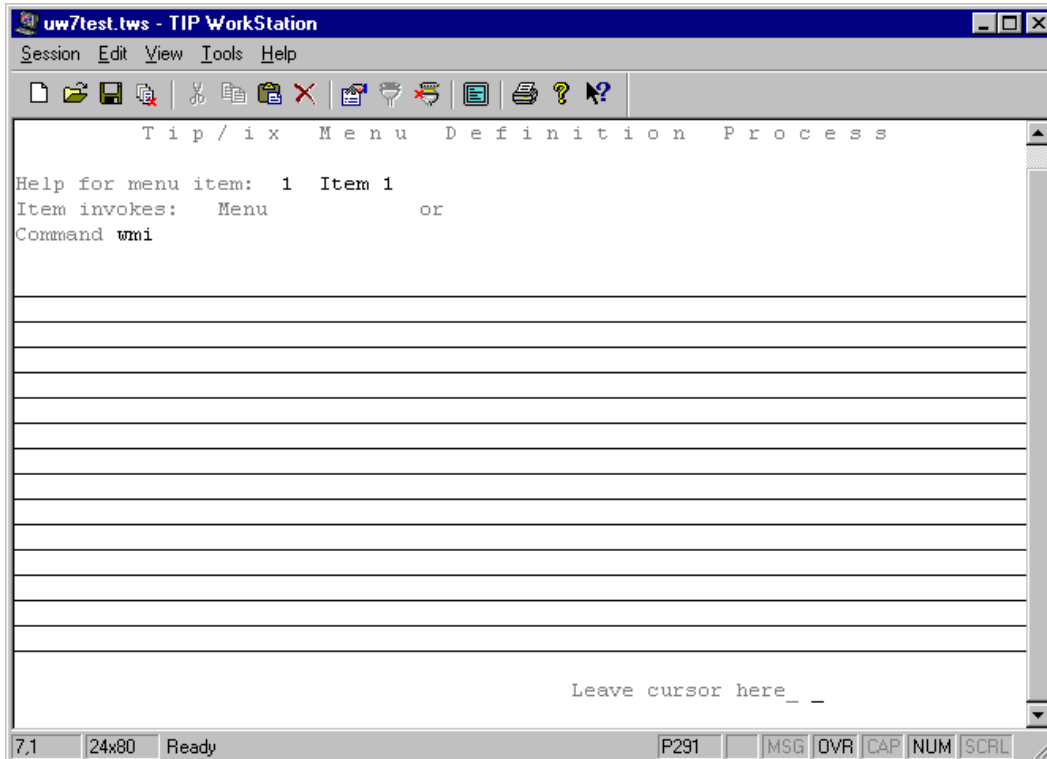


Alternatively, to skip directly to help definition from the command line you may enter:

menudef ,h BJONES

This places you directly at the above help definition screen for menu BJONES.

The following screen shows how **menudef** prompts the user to define help information for the selected MENU item.



Enter the help information in the space provided. Notice that the command line or menu name represented by this menu item is displayed as a reminder when entering the help text for the item.

Press **XMIT** when the help screen is complete. The help information is saved in the menu file and the user is returned to the help definition menu.

To cancel the definition of *this* help screen press the **MSG WAIT** key to return to the screen shown on the previous page (the beginning of "Phase").

Function Key Use

MSG WAIT

Cancels the definition and terminates the program in menudef phases 1, 2, and 3.

F1 or F5

Redisplay the current screen as it was last sent to the terminal.

F2 or F6

Used in menudef phase

mfm - Message File Maintenance

The Message File Maintenance Utility (**mfm**) allows you to manipulate the system messages contained in the TIP/ix system message file **TIP\$MSG**. Using **mfm** you can perform a number of message maintenance functions:

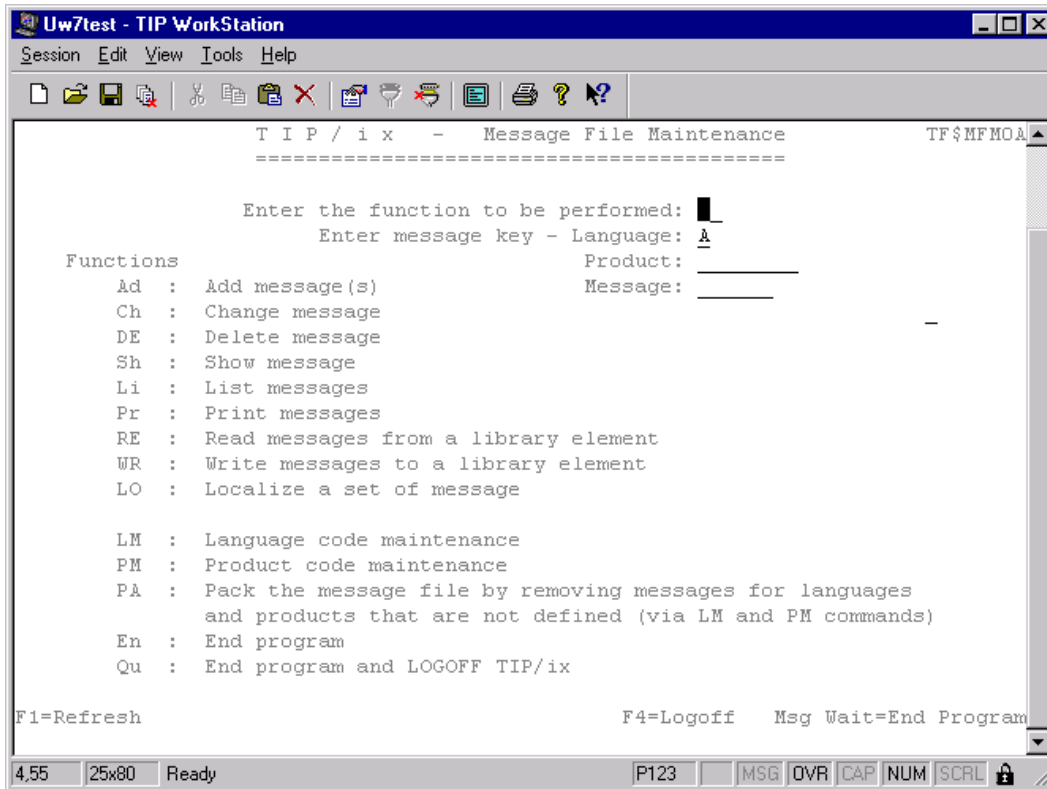
- Add messages
- Change messages
- Delete messages
- List messages
- Print messages
- Show (display) individual messages
- Read messages from a library element
- Write messages to a library element
- Read a file from a library element
- Write a file to a library element
- Localize a set of messages
- Maintain the language code
- Maintain the product code
- Pack the message file.

To invoke **mfm**, key in the following at the TIP/ix prompt:

Syntax:

mfm

TIP/ix displays the following main mfm screen:



Where:

Enter function:

Enter one of the two-character function codes from the list displayed on the right hand side of the screen.

Example: AD

Language:

Specify a national language. Default: American English.

Product:

A product name (for example: TIPIX.)

Message:

Message number (for example: CAL001.)

If you view the current TIP/ix messages (for example, see [LI List Messages](#)) you will notice that several of the messages contain edit codes. The table below explains these codes:

Edit Code	Description	Type
<	Start blink.	Variable input data required.
>	End blink.	Variable input data required.
?	Start of entry	Variable input data required.

	character.	
::	Set tab.	Variable input data required.
::	Semi-colon characters (multiple semi-colons).	Variable input data required.
;0	(semi-colon, zero) Leading zero suppression.	Variable input data required.
;A	Alphanumeric current date. Format: DD MMM YY Example: 15 Feb 91	Fixed length result, independent of number of redundant edit codes.
;C	Current time. Format: HH:MM Example: 11:59	Fixed length result, independent of number of redundant edit codes.
;D	Current date (day). Format: DD/MM/YY Example: 28/02/91	Fixed length result, independent of number of redundant edit codes.
;E	Extended current time. Format: HH:MM:SS Example: 11:59:03	Fixed length result, independent of number of redundant edit codes.
;F	Full date Format: MMMMMMMMMM DD, YYYY Example: March 10, 1998	Fixed length result, independent of number of redundant edit codes.
;Hhh;	Enclosed pairs of characters converted to hexadecimal.	Fixed result. no variable input.

;I	Site identifier Size: 12 bytes	Fixed length result, independent of number of redundant edit codes.
;J	Current Julian date. Format: YYDDD Example: 91/103	Fixed length result, independent of number of redundant edit codes.
;L	LOCAP identifier. Size: four bytes.	Fixed length result, independent of number of redundant edit codes.
;M	Current date (month). Format: MM/DD/YY Example: 02/02/91	Fixed length result, independent of number of redundant edit codes.
;N	Network name. Size: four bytes.	Fixed length result, independent of number of redundant edit codes.
;S	Trailing space suppression.	Variable input data required.
;T	Terminal identifier of caller. Size: four bytes.	Fixed length result, independent of number of redundant edit codes.
;U	user id of caller.	Fixed length result, independent of number of redundant edit codes.
;V	Variable field, input is terminated by 'C'; or 'X'00'	Variable input data required.
;X	Literal substitution, no suppression.	Variable input data required.
;x	Single character literal substitution. (";", followed by any character other than defined edit code.)	Variable input data required.
;Y	Current date (year) Format:	Fixed length result, independent of number of redundant edit codes.

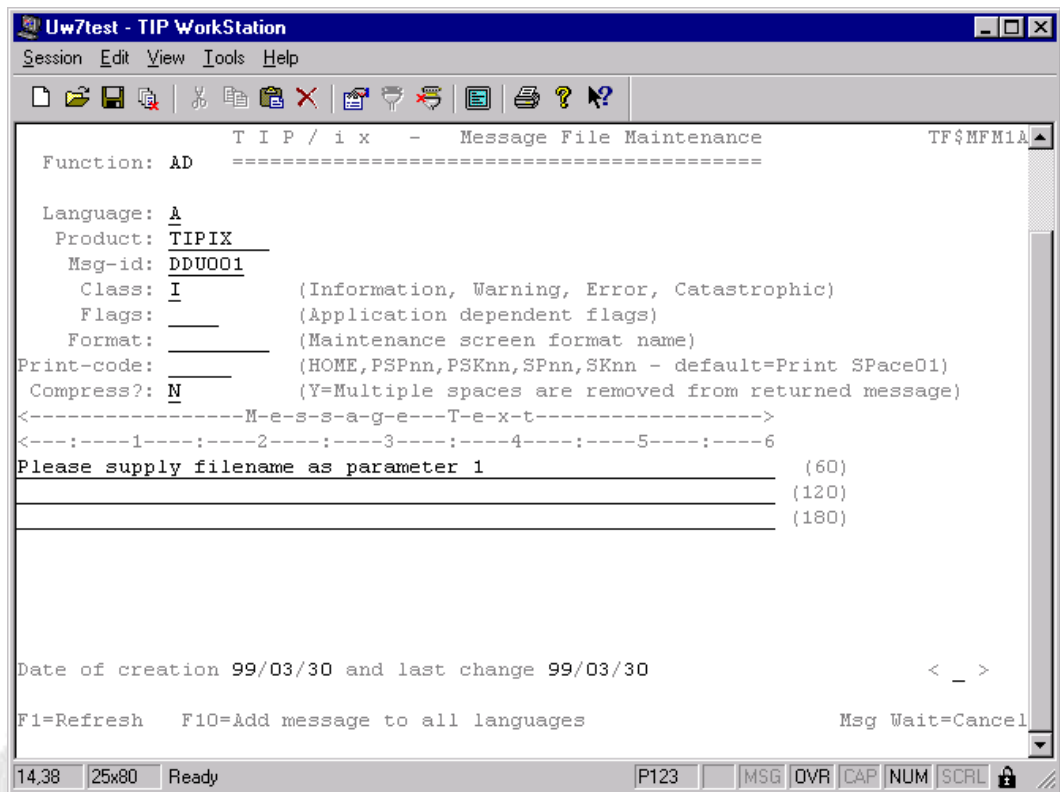
YY/MM/DD

Example:
91/05/21

;Z Leading zeros Variable input data required.
 replaced by
 spaces.

AD - Add Messages

Select this function to add messages to those already in the message file. **mfm** displays the following screen when you select **AD**:



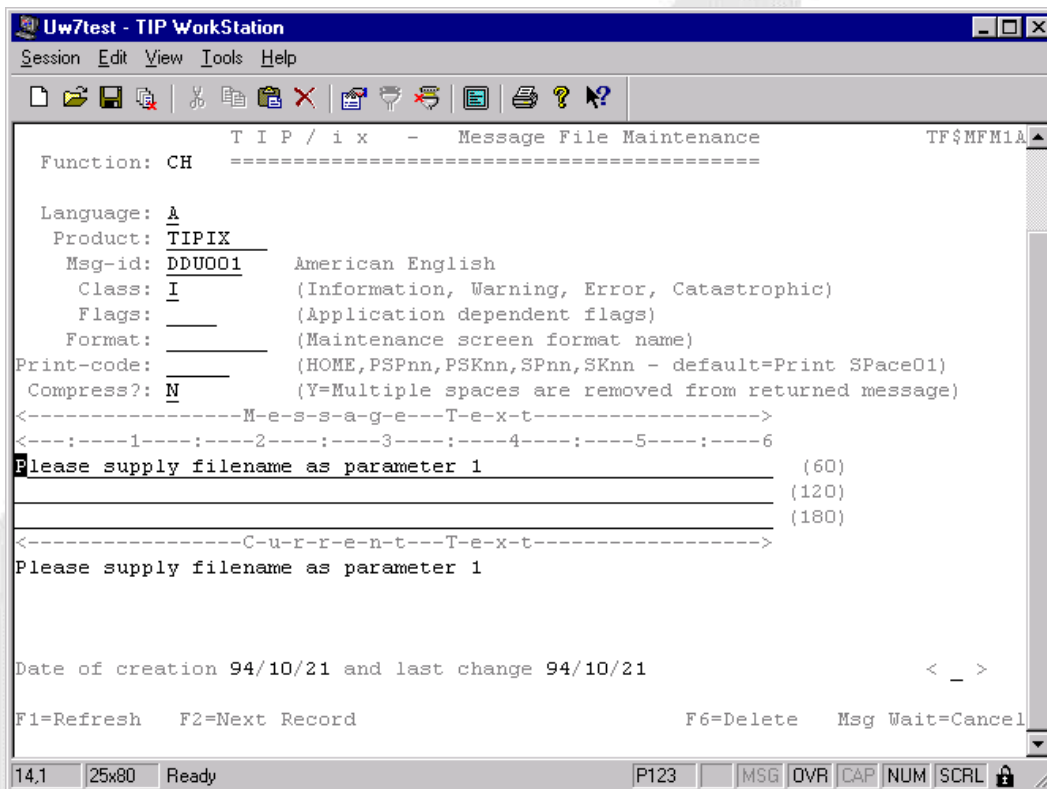
This screen format is used by a majority of **mfm** functions. Consequently, the description of the screen's fields (below) are not repeated in later sections that describe **mfm**'s functions.

Field	Description
Function:	The function you are performing on the current screen.
Language:	Specify a national language. Default: American English.
Product:	A product name (for example: TIP/ix)

Message:	Message number (for example: ALL000)
Class:	The message class: Information, Warning, Error or Catastrophic
Flags:	Application dependent flags.
Format:	Maintenance screen format name.
Print-code:	HOME,PSPnn,PSKnn,SPnn,and SKnn. The default is Print and SSpace01.
Compress?:	If you specify "Y", multiple spaces are removed from returned message.
Message Text:	The message text, which can consist of three lines of sixty characters each.

CH - Change Messages

Select this function to change messages in the message file. **mfm** displays the following screen when you select **CH**:



```

Uw7test - TIP WorkStation
Session Edit View Tools Help
-----
T I P / i x - Message File Maintenance TF$MFM1A
Function: CH
-----
Language:  A
Product:  TIPIX
Msg-id:   DDU001   American English
Class:    I        (Information, Warning, Error, Catastrophic)
Flags:    _____ (Application dependent flags)
Format:   _____ (Maintenance screen format name)
Print-code: _____ (HOME,PSPnn,PSKnn,SPnn,SKnn - default=Print SSpace01)
Compress?: N      (Y=Multiple spaces are removed from returned message)
<-----M-e-s-s-a-g-e---T-e-x-t----->
<---:---1---:---2---:---3---:---4---:---5---:---6
Please supply filename as parameter 1 (60)
_____(120)
_____(180)
<-----C-u-r-r-e-n-t---T-e-x-t----->
Please supply filename as parameter 1

Date of creation 94/10/21 and last change 94/10/21 < _ >
F1=Refresh F2=Next Record F6=Delete Msg Wait=Cancel
14.1 25x80 Ready P123 MSG OVR CAP NUM SCRL

```

Where:

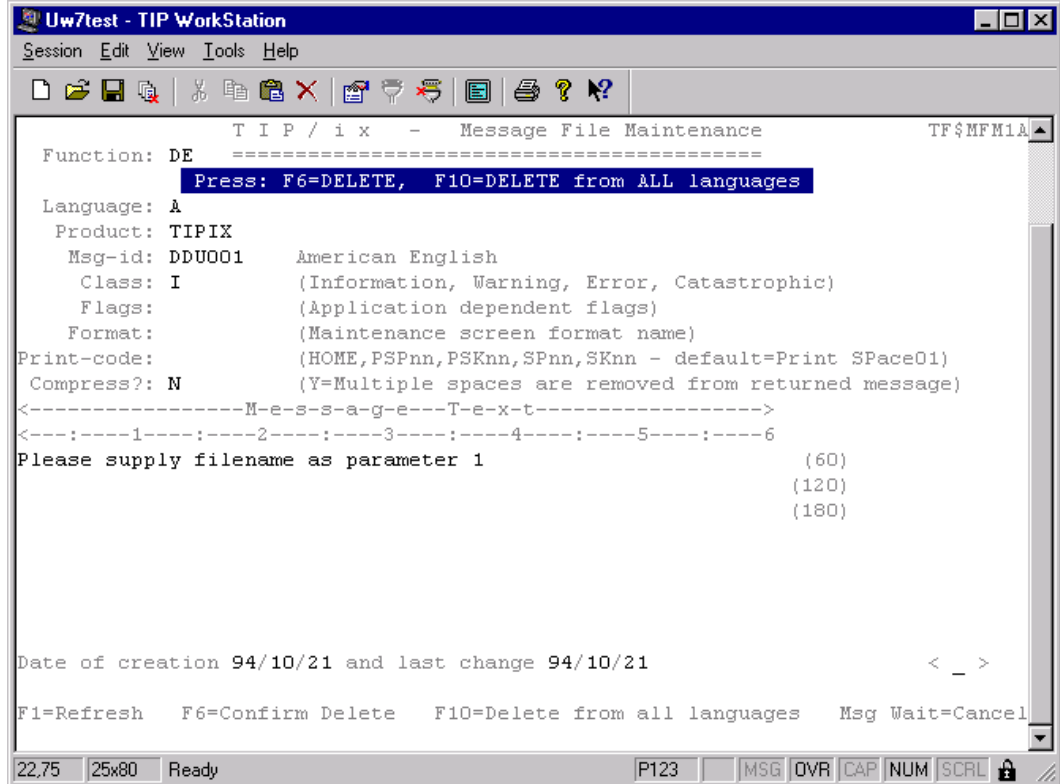
Most of the fields used on this screen are explained in AD Add Messages .

Current Text

This area displays the message while you change it in the Message Text area.

DE - Delete Messages

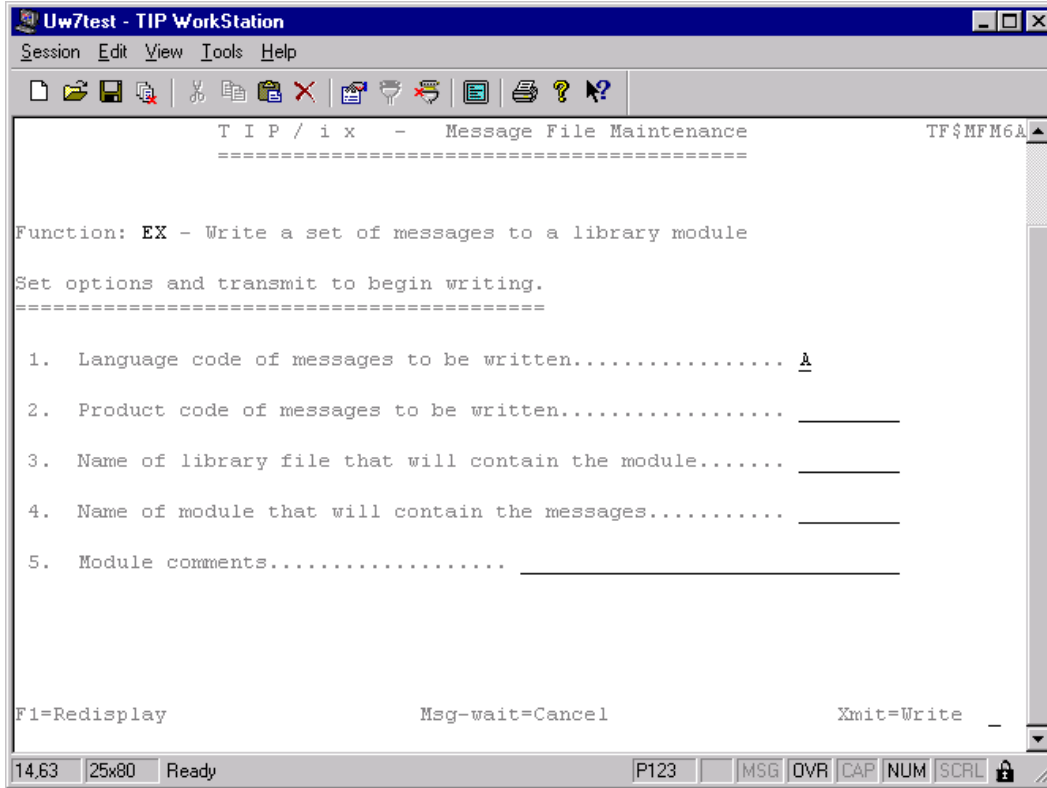
Select this function to delete messages in the message file. **mfm** displays the following screen when you select **DE**:



The fields in this screen are identical to those used in [AD Add Messages](#). Use the function keys to delete the selected message for the displayed language ("A") or for all languages.

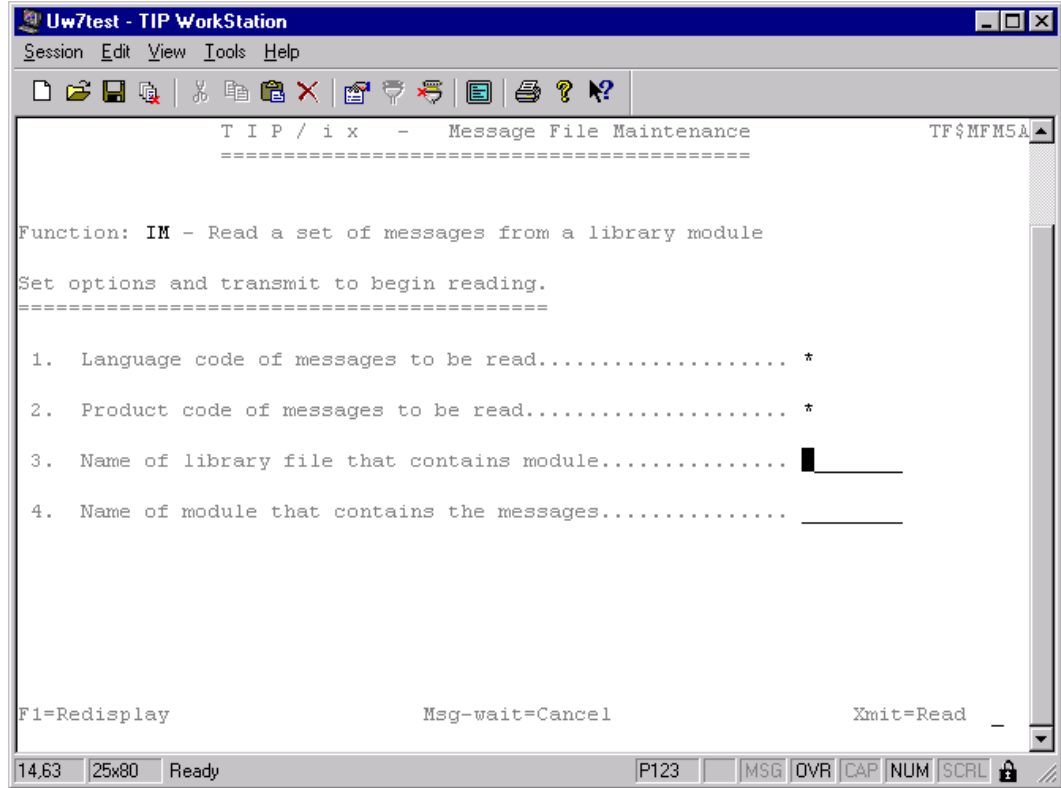
EX - Write Messages to a Library Element

Select this function to write messages to a library element. **mfm** displays the following screen when you select **EX**:



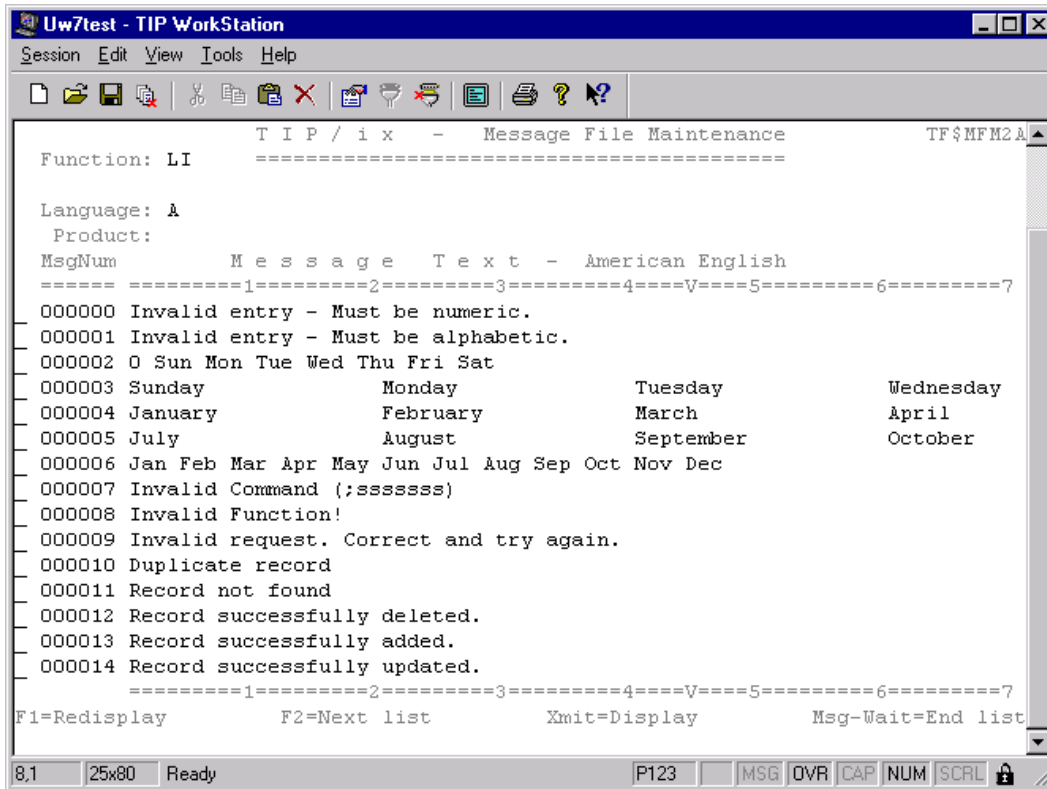
IM - Read Messages from a Library Element

Select this function to read a set of messages from a library element. **mfm** displays the following screen when you select **IM**:



LI - List Messages

Select this function to list messages from the message file. **mfm** displays the following screen when you select **LI**:



```

Uw7test - TIP WorkStation
Session Edit View Tools Help

TIP / i x - Message File Maintenance
Function: LI
Language: A
Product:
MsgNum Message Text - American English
=====1=====2=====3=====4====V====5=====6=====7
000000 Invalid entry - Must be numeric.
000001 Invalid entry - Must be alphabetic.
000002 0 Sun Mon Tue Wed Thu Fri Sat
000003 Sunday Monday Tuesday Wednesday
000004 January February March April
000005 July August September October
000006 Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
000007 Invalid Command (;ssssss)
000008 Invalid Function!
000009 Invalid request. Correct and try again.
000010 Duplicate record
000011 Record not found
000012 Record successfully deleted.
000013 Record successfully added.
000014 Record successfully updated.
=====1=====2=====3=====4====V====5=====6=====7
F1=Redisplay F2=Next list Xmit=Display Msg-Wait=End list

8,1 25x80 Ready P123 MSG OVR CAP NUM SCRL

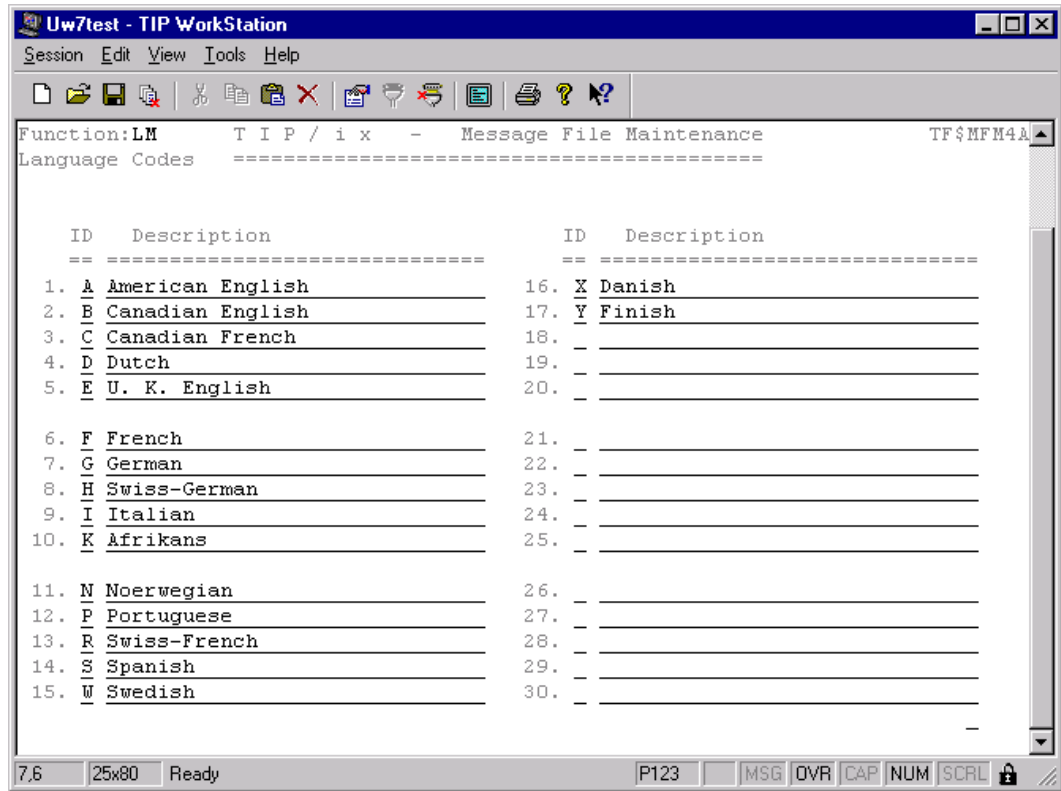
```

If you specify the language code, product code and message number **mfm** will begin listing messages beginning with the specified message.

If you position the cursor to the left of a selected message number and press **XMIT**, **mfm** will display the message details. Use the specified function keys to either delete or update the message at this time.

LM - Language Code Maintenance

Select this function to update the language codes in **mfm**. **mfm** displays the following screen when you select **LM**:



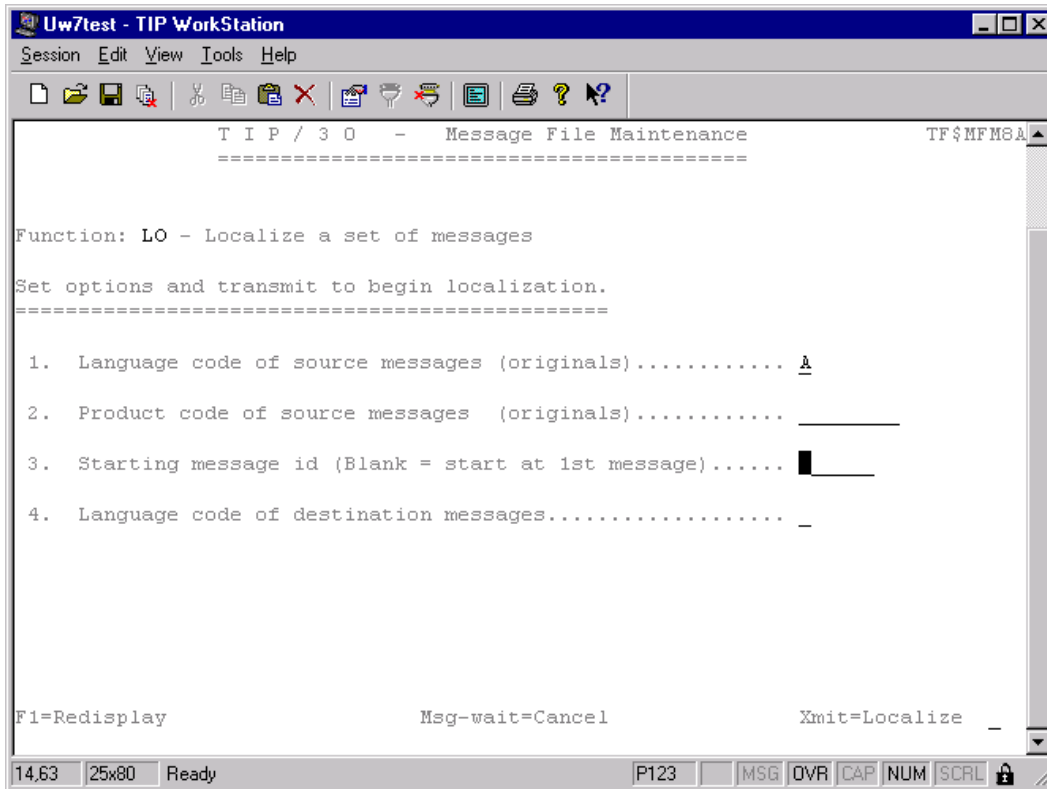
To add a language code enter it on this screen and press **XMIT**.

To refresh the screen press **F1**.

To cancel the update, press **Esc**.

LO - Localize a Set of Messages

Select this function to localize messages in the message file. **mfm** displays the following screen when you select **LO**:



If, for example, you wish to provide a French language variation of the system message **SPL001** you would supply the following information:

- Language code of source messages (originals) - **A**
- Product code of source messages (originals) - **TIP/ix**
- Starting message id (Blank = start at 1st message) - **DDU** and
- Language code of destination messages - **F**.

mfm would display the following screen:

```

Uw7test - TIP WorkStation
Session Edit View Tools Help

TIP / ix - Message File Maintenance TF$MFM1A

Function: LO
-----

Language: F
Product: TIPIX
Msg-id: DDU001      French
Class: I           (Information, Warning, Error, Catastrophic)
Flags:             (Application dependent flags)
Format:            (Maintenance screen format name)
Print-code:        (HOME, PSPnn, PSKnn, SPnn, SKnn - default=Print SSpace01)
Compress?: N       (Y=Multiple spaces are removed from returned message)
<-----M-e-s-s-a-g-e---T-e-x-t----->
<---:---1---:---2---:---3---:---4---:---5---:---6
Please supply filename as parameter 1                (60)
-----                                           (120)
-----                                           (180)
<-----American-English----->
Please supply filename as parameter 1

Date of creation 94/10/21 and last change 94/10/21      < _ >

F1=Refresh  F2=Next Record                F6=Delete  Msg Wait=Cancel

14.1 25x80 Ready                                P123 MSG OVR CAP NUM SCRL
    
```

You would enter the French text of the message in the *Message Text* area of the screen. The original language version appears in the *American-English* section of the screen.

PA - Pack the Message File

The PAcK command scans the canned message file and removes all messages for which there is not corresponding Language or Product defined.

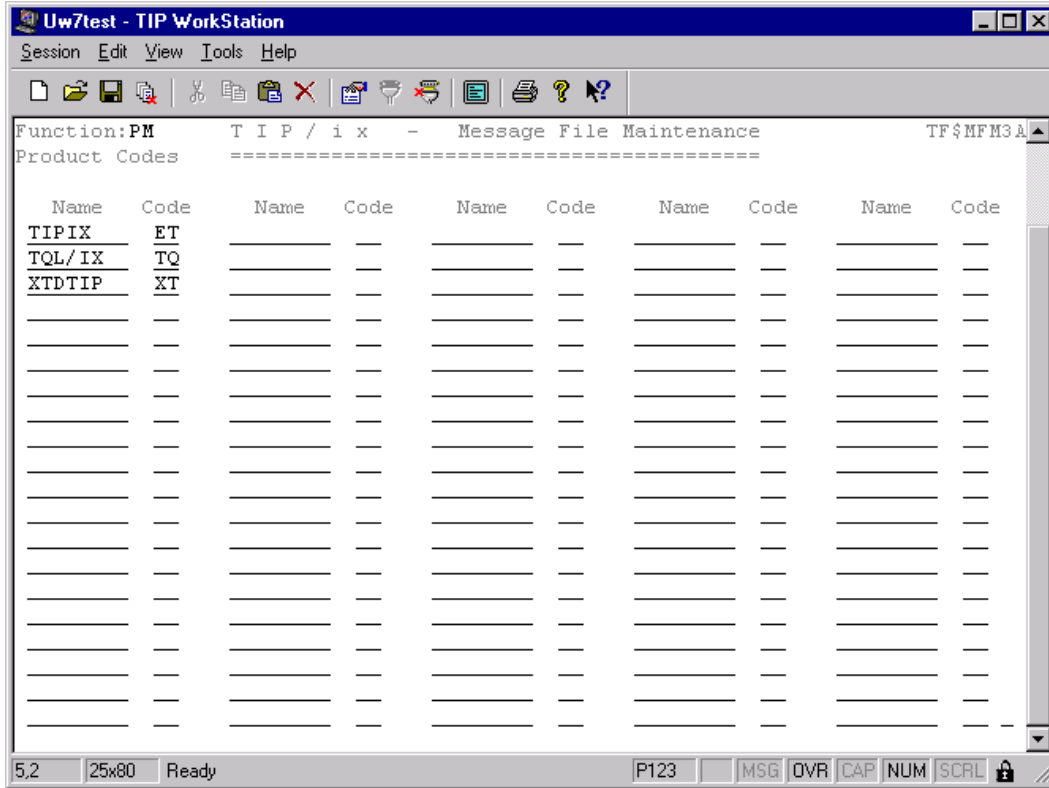
For example, if you want to remove all messages for a Product, you would use PM and remove the Product, then PA would delete any messages defined for that product.

mfm displays the message when you select **PA** and **mfm** has completed this function:

File PAC completed successfully

PM - Product Code Maintenance

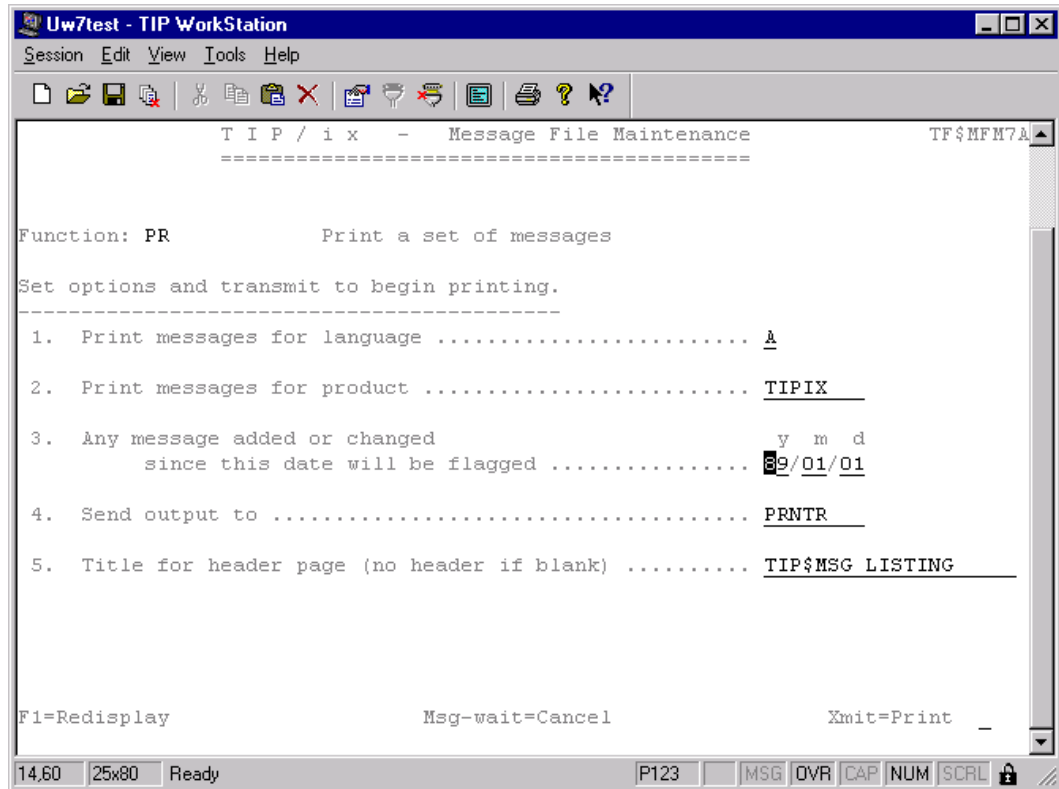
Select this function to update the list of product codes in **mfm**. **mfm** displays the following screen when you select **PM**:



- To add a product code enter it on this screen and press XMIT.
- To refresh the screen press F1.
- To cancel the update, press Esc.

PR - Print Messages

Select this function to print messages from the message file. **mfm** displays the following screen when you select **PR**:

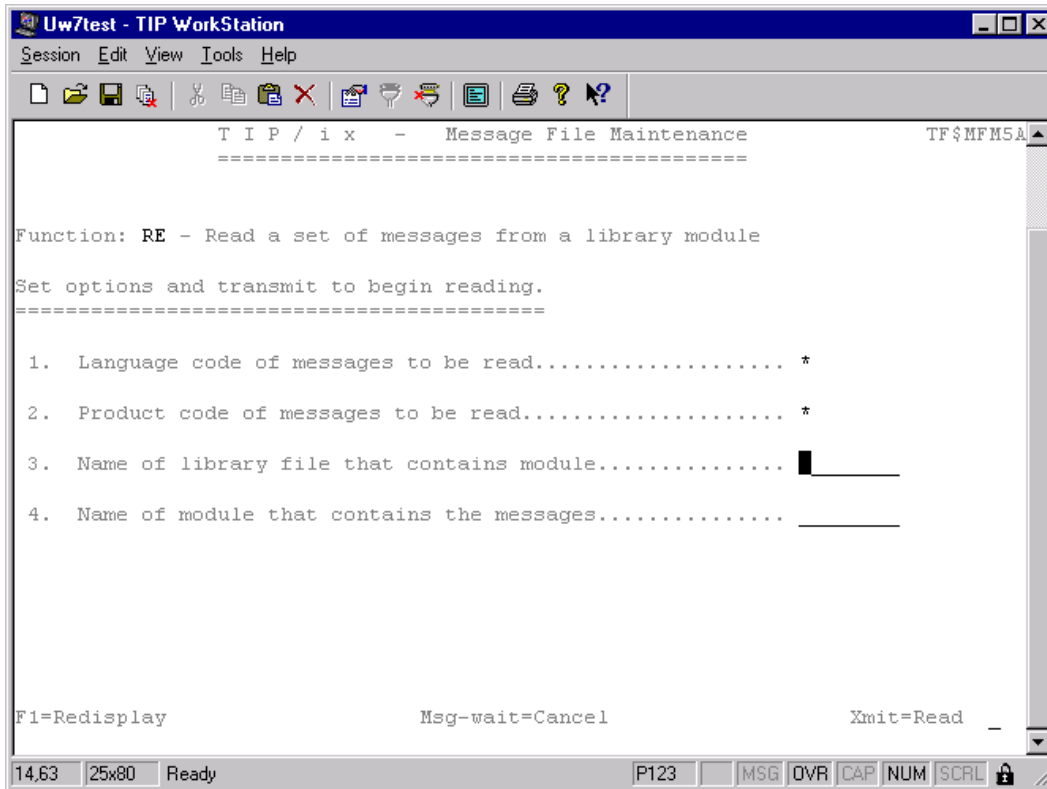


This screen allows you to:

- Print messages for a specific language.
- Print messages for a specific product.
- Flag any message added or changed since a specified date in your printout.
- Send output to a specific print destination.
- Specify the title text for the printout header page. No title page is printed if you leave this blank.

RE - Read File from a Library Element

Select this function to read a file from a library element. **mfm** displays the following screen when you select **RE**:



SH - Show Individual Messages

Select this function to show individual messages from the message file. You would use this function if you already knew the message number of the message. **mfm** displays the following screen when you select **SH**: Even though you entered **SH** in the Function section on the screen below you will see **DI** displayed. These two commands are interchangeable.

```

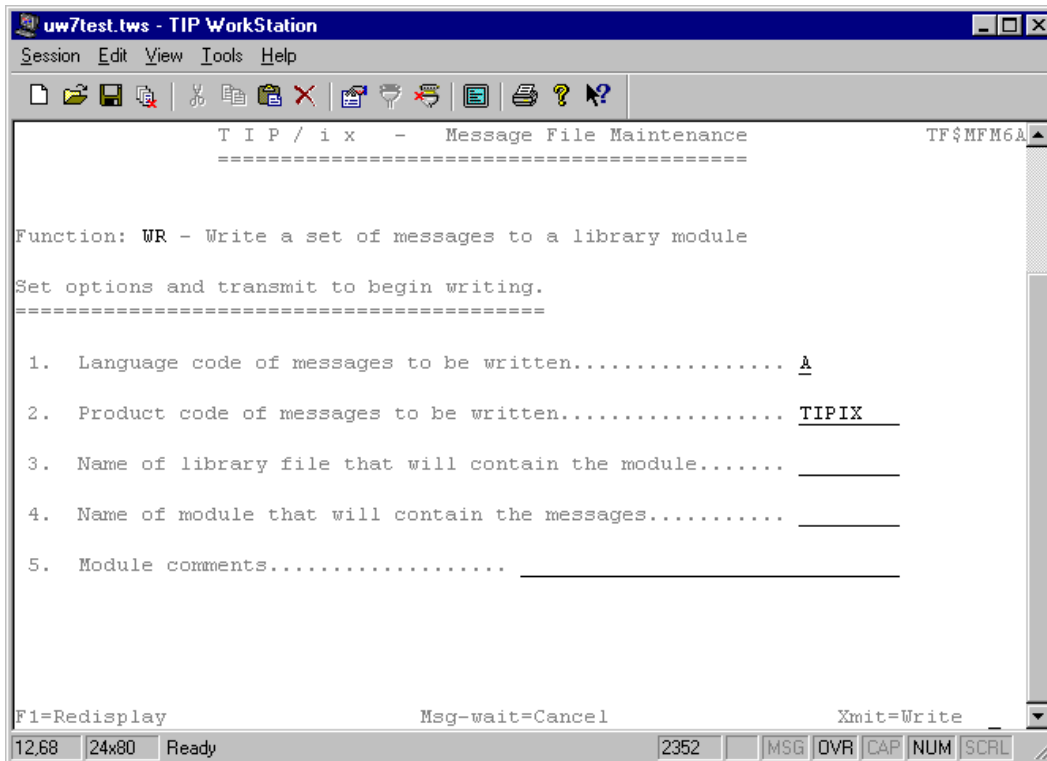
Uw7test - TIP WorkStation
Session Edit View Tools Help
-----
Function: DI
Language: A
Product: TIPIX
Msg-id: DDU001 American English
Class: I (Information, Warning, Error, Catastrophic)
Flags: (Application dependent flags)
Format: (Maintenance screen format name)
Print-code: (HOME, PSPnn, PSKnn, SPnn, SKnn - default=Print SSpace01)
Compress?: N (Y=Multiple spaces are removed from returned message)
-----M-e-s-s-a-g-e---T-e-x-t----->
<---:---1---:---2---:---3---:---4---:---5---:---6
Please supply filename as parameter 1
                                           (60)
                                           (120)
                                           (180)

Date of creation 94/10/21 and last change 94/10/21
F1=Refresh F2=Next F3=List F4=Update F6=Delete Msg Wait=Cancel
6,13 25x80 Ready P123 MSG OVR CAP NUM SCRL
    
```

The fields in this screen are identical to those used in [AD Add Messages](#). Use the function keys to list all messages that follow this message, or to update or delete the selected message.

WR - Write File to a Library Element

Select this function to write messages from the message file to a library element. **mfm** displays the following screen when you select **WR**:



msg - Send Terminal a Message

The **msg** program allows a terminal user to send a one line message to a logged on TIP/ix user, a specific terminal, to all terminals running a program, or to all terminals where a specific file is being accessed.

The message text is sent as an unsolicited message. A BEL character is sent to the terminal to indicate the presence of a message. If a recipient of a message is running TIP/fe then the bell can be seen (single flash) and/or heard (single beep) and the TIP/fe status bar will indicate a message is available (after any key at that TIP/fe session is pressed).

The message is displayed at the cursor location at the time MSG WAIT is pressed! It is a good idea to first move the cursor to an area of the screen that is not in use.

Syntax 1 - From the TIP/ix command line:

```
msg text
msg, [[*]user id] text
msg, [[*]terminal name] text
msg, [[*]filename]text
msg, [[*]program name]text
```


Syntax 2 - From the UNIX command line:

```
msg [[\*]user id] text
msg [[\*]terminal name] text
msg [[\*]filename]text
msg [[\*]program name]text
```

Where:**no destination**

If no destination is specified, the terminal sending the message receives the message.

user id

Send the message to all TIP/ix sessions being operated by this user id.

terminal name

Send the message to all TIP/ix sessions at this terminal name.

filename

Send the message to the users that are using this file.

program name

Send the message to the users that are running this program.

[*] TIP/ix: TIP prefix notation may be used.
For example:

```
msg,*pay please logoff
```

to send anyone using a filename, program, terminal, or user id that is prefixed with "pay".

[*] UNIX: The backslash protects the asterisk from being expanded as a list of UNIX files.

text The text to be sent (maximum of 64 characters).

The text does not have to be enclosed in quotes. If quotes are included, they are treated as text.

The text of the message is translated to uppercase before the text is sent.

If you try to match a program name, a match only occurs if the program you want is active at the highest stack level of a TIP/ix session.

For example, run "tcm" then select "user id definition". In this case tcm is running at stack level 1 and smuser is running at stack level 2.

So "msg smuser" would match this session but "msg tcm" would not.

Examples:

```
MSG,DAVE TRY TO LOG ON NOW!  
MSG,MANUFIL MANUFACTURING FILES BEING CLOSED SOON!  
MSG NO TARGET SPECIFIED SO I GET THIS MESSAGE MYSELF.
```

Error Conditions:

No errors are reported.

msgar - Message Format Utility

The **msgar** program provides librarian services for TIP/ix screen formats.

Features of MSGAR

The MSGAR utility provides the following features:

- Adding and removing screen formats from the screen format file.
- Generating COBOL copybooks or C header files from a screen format.
- Generating SFG style source from a screen format and similarly creating a screen format from SFG style source.
- Importing DPS 1100 style source to create a screen format and generate an associated COBOL copybook.
- Importing Basic Mapping Support (BMS) style source to create a screen format.
- Displaying a list of the screen formats in the screen format file.
- Help text identifier functions such as adding and removing help from the screen format file.

Using these features, screen formats can be copied or moved from one TIP system such as TIP/ix or TIP/30 to another TIP system. As well, the corresponding COBOL copybooks and help text associated with each screen format will be carried from one TIP system to another.

The MSGAR utility syntax is as follows:

Syntax:

```
msgar command [parameters]
```

Where:**Command**

A valid MSGAR command from the table of valid commands described below.

Parameters

One or more parameters.

name	A screen format name.
file	A file name.
pattern	A name with wild cards. "?" matches one character. "*" matches several.
row col	Screen coordinates.
key	Function keys.

MSGAR Commands

MSGAR accepts several commands. If you enter the command **msgar** with no parameters you will receive a summary of available commands,

Command	Description
ALLOFF name	Disable full screen transmit.
ALLON name	Enable full screen transmit.
CObol name	Produce COBOL copybook of screen format.
CUrsor name row col	Coordinates of new cursor resting location.
DElete name	Delete a screen format.
DELETE pattern	Delete multiple screen formats using wildcards.
HDR name	Produce C header file from screen format.
HDelete pattern	Delete help text records using wildcards.
HImport infile name	Import ASCII help text files.
HList pattern [outfile]	List help text records.
HSetKey pattern syskey scrkey fldkey	Set context sensitive help functions keys.
HUpdate name	Update help text using vi.
HXport pattern [outfile]	Export ASCII help text file.
K	Display MSGAR exit codes.
List pattern [outfile]	List formats in screen format file.

Print name [outfile]	Print format to stdout.
Rename oldname newame	Change screen format name.
Import infile [name]	Import screen format from file.
IBms infile [name]	Import screen format from file in Basic Mapping Support (BMS) style text.
IDps infile [name]	Import screen format from file in DPS 1100 style text and generate associated COBOL copybook.
IDPSIMS infile [name]	Same as IDps command except generate only the 'DATA' COBOL copybook. Field names are constructed with the screen number as a prefix.
IDPSIMSX infile [name]	Same as IDPSIMS command except omit the screen number as the field name prefix.
ISFG infile [name]	Import screen format from file in SFG style text.
Xport pattern [outfile]	eXport screen format to file in ASCII mode.
XCport pattern [outfile]	eXport screen format to file in host Compatibility mode i.e., 2560 byte image with NO help text information, EBCDIC.
XDps pattern [outfile]	eXport screen format to file in DPS 1100 style text.
XSFG pattern [outfile]	eXport screen format to file in SFG style text.
UXport pattern [outfile]	Unload command. Same as Xport but exports to 'group' directory on output.
UXCport pattern [outfile]	Unload command. Same as XCport but exports to 'group' directory on output.
UXSFG pattern [outfile]	Unload command. Same as XSFG but exports to 'group' directory on output.
?	MSGAR examples.

ALLON or ALLOFF

This command sets full screen transmit either ON or OFF for the given screen. With full screen transmit ON, MCS-FUNCTION will be set to 'A'

for full transmit. The following example sets full screen transmit ON for the screen "ACCT/PAYROLL".

```
msgar allon acct/payroll
```

CObol name

Generates a COBOL copybook for a screen format to a file. The generated file will be the screen format name followed by ".cob". For example, the command "msgar co myformat" generates a COBOL copybook for the screen format "TIP\$\$/MYFORMAT" in the file "myformat.cob".

All fields will be named with the screen format name followed by a sequential number to distinguish between fields. A table for the FCC mods and cursor mods will be included as comments to be used if required.

The generated COBOL copybook will be similar to:

```
*-----
* MCS screen 'TIP$$/MYFORMAT' in COBOL format
*-----
02 MYFORMAT-DATA REDEFINES MCS-DATA.
   05 MYFORMAT-001.
      10 FILLER                PICTURE 9(4) .
   05 MYFORMAT-002.
      10 FILLER                PICTURE X(30) .
   05 MYFORMAT-003.
      10 FILLER                PICTURE 9(2) .
   05 MYFORMAT-004.
      10 FILLER                PICTURE ZZ9.
*
*02 MYFORMAT-FCC-MODS          OCCURS 4 TIMES
*                               INDEXED BY FCC-IDX
*                               PICTURE XX.
*
*02 MYFORMAT-CURSOR-MODS      OCCURS 4 TIMES
*                               INDEXED BY CUR-IDX
*                               PICTURE X.
*
* Total error field length: 10
```

CURsor name row col

Modifies the cursor resting location for a screen format. The new coordinates, *row* and *col*, are given as the new row and column, as in the following example:

```
msgar cu myformat 8 30
```

The new cursor resting location for the screen format "TIP\$Y\$/MYFORMAT" will be set to row 8 and column 30.

Specifying a row and col of 0 will set the cursor resting location to the first unprotected field. For example:

```
msgar cu myformat 0 0
```

DElete name

Deletes a single screen format. NO wildcard specifications are allowed with this command to delete multiple screen formats. See the following command for deleting multiple screen formats. The command is used as follows:

```
msgar de myformat - deletes format 'TIP$Y$/MYFORMAT'
msgar de arc/fmt1 - deletes format 'ARC/FMT1'
msgar de arc/* - invalid syntax
```

DELETE pattern

Deletes multiple screen formats using wildcard specifications given in 'pattern'. The wildcards possible are "?" to match a single character and "*" to match one or more characters. The command is used as follows:

```
msgar delete myfm* - deletes all screen formats
  matching 'TIP$Y$/MYFM*'
msgar delete arc/* - deletes all screen formats
  in group 'ARC'
msgar delete myfmt - deletes format 'TIP$Y$/MYFMT'
msgar delete a*/?/ - invalid syntax
```

HDR name

Generates a C header file for a screen format to a file. The file generated will be the screen format name followed by ".h". The command is used as follows:

```
msgar hdr myformat
```

will generate a C header file for the screen format "TIP\$Y\$/MYFORMAT" in the file "myformat.h". All fields will be named with the screen format name followed by a sequential number to distinguish between fields. A table for the FCC mods and cursor mods will be included as comments to be used if required.

The C header file will be similar to:

```
/*-----
MCS screen 'TIP$Y$/MYFORMAT' in C style format
```

```

-----*/
static struct McsData {
unsigned char myformat_001 [5];
unsigned char myformat_002 [5];
unsigned char myformat_003 [5];
unsigned char myformat_004 [5];
unsigned char myformat_005 [6];
}
/* myformat-fcc-mods unsigned char [5][2] */
/* myformat-cursor-mods unsigned char [5][1] */
/* Total error field length: 10 */

```

HDELETE pattern

Deletes help text from the screen format file. Wildcard specification is allowed to delete multiple help text identifiers. The command is used as follows:

```

msgar hd hlp1
- deletes help text identifier 'HLP1'
msgar hd hlp*
- deletes all help text identifiers matching 'HLP*'

```

Himport infile [name]

Imports ASCII help text files. A help text identifier will be created in the screen format file containing the imported ASCII help text. The input file name, "infile", must be given and an optional help text identifier, "name", may be specified to indicate the destination for the help text.

If a destination help text identifier is not given, then the help text identifier name will be taken from the "HELP= " line in the input file. The command is used as follows:

```

msgar hi hlp.txt
- Import the help text in file 'hlp.txt' to the help
  text identifier given in file
msgar hi hlp.txt help2
- Import the help text in file 'hlp.txt' to the help
  text identifier 'HELP2'

```

An ASCII help text file to be imported should be of the following form:

```

* These are comment lines possibly indicating
information
* about the help text file
HELP=HLP1 "up to 40 byte optional description"
Help text begins on this line following the above line
containing the HELP= command.
This is line 3 of the help text.
This is line 4 of the help text.
This is line 5 of the help text.

```

```

This is line 6 of the help text.
This is line 7 of the help text.
These are the last three lines of the help text since
they precede the help termination line of a single
period in the first position of the following line.
.

```

As described in the help text itself, the file must contain a line starting with "HELP=" and the name of the help text. Following this line is the help text until the help termination of a single period in the first position is reached.

HList pattern [outfile]

Displays a list of help text identifiers in the screen format file matching "pattern" to "outfile". Wildcard specification is allowed to select which help text to list. The default "pattern" is "*" and the default "outfile" is "stdout".

The command is used as follows:

```

msgar hl - list all help text to stdout
msgar hl * - list all help text to stdout
msgar hl hlp* - list all help text matching
'HLP*' to stdout
msgar hl * hlp.out - list all help text to the file
'hlp.out'

```

HSetKey pattern Syskey Scrkey Fldkey

Set the context sensitive help function keys for all screens matching pattern to the corresponding System level help (Syskey), Screen level help (Scrkey) and Field level help (Fldkey). The values for the function keys must be between F0 and F22 where F0 indicates to leave that function key alone and F1 to F22 indicate the function key to set for that level of help.

The command is used as follows:

```

msgar hs arc/scrna F1 F2 F3
- set the function keys to F1 F2 and F3 for the System,
Screen
and Field level help respectively for the screen
'ARC/SCRNA'.
msgar hs arc/* F0 F6 F7
- set the function keys to F6 and F7 for the Screen and
Field
level help respectively while leaving the System level
help
alone for all screens in the group 'ARC'.

```


HUpdate name

Updates help text. The help text identifier given is exported to an ASCII file and "vi" is called to edit that file. After editing is completed, the text file is imported back into the screen format file under the same help text identifier. The command is used as follows:

```
msgar hu myhelp
- Exports help text identifier 'MYHELP' to a
  file and calls 'vi' to edit the file and then
  returns the file to 'MYHELP'
```

HXport pattern [outfile]

Exports a help text identifier to an ASCII help text file. The output file name is given by the help text identifier followed by ".txt" unless an optional file name is given in "outfile". Wildcard specification is allowed to export multiple help text identifiers using "pattern".

The command is used as follows:

```
msgar hx myhelp
- Export help text identifier 'MYHELP' to
  the file 'myhelp.txt'
msgar hx myhelp hlp.out
- Export help text identifier 'MYHELP' to
  the file 'hlp.out'
msgar hx myh*
- Export all help text identifiers matching
  'MYH*' to separate files 'myh*.txt'
```

K

Displays a list of the MSGAR exit codes. Different exit codes are returned from MSGAR due to errors or other warning indications such as files do not exist or they already exist or I/O errors. The command is used as follows:

```
msgar k - Display a list of MSGAR exit codes
```

List pattern [outfile]

Displays a list of screen formats in the screen format file matching "pattern" to "outfile". Wildcard specification is allowed to select which screen formats to list. The default "pattern" is "*"/*" and the default "outfile" is "stdout". The command is used as follows:

```
msgar l
- list all formats in all groups to stdout
msgar l */*
```

```

- list all formats in all groups to stdout
msgar l *
- list all formats in group 'TIP$Y$' to stdout
msgar l arc/*
- list all formats in group 'ARC' to stdout
msgar l ar?/xy*
- list all formats matching 'AR?/XY*' to stdout
msgar l */* scr.out
- list all formats in all groups to file 'scr.out'

```

Print name [outfile]

Prints formats to "outfile". Wildcard specification is allowed to select which screen formats to print. The default destination for "outfile" is "stdout". The command is used as follows:

```

msgar p myfmt
- prints format 'TIP$Y$/MYFMT' to stdout
msgar p myfmt myfile
- prints format 'TIP$Y$/MYFMT' to file 'myfile'
msgar p myfmt LP
- spools format 'TIP$Y$/MYFMT' to UNIX 'lp'

```

The output generated from printing would show a few comment lines containing the format name, the number of error bytes, the number of data bytes and some of the screens attributes. Following this, the screen format with the row and column coordinates would be displayed.

Rename oldname newname

Changes a screen format named "oldname" to "newname". The command is used as follows:

```

msgar rename fmt1 fmt2
- renames the screen format 'TIP$Y$/FMT1'
  to 'TIP$Y$/FMT2'
msgar rename arc/f1 foo/f2
- renames the screen format 'ARC/F1'
  to 'FOO/F2'

```

Import infile [name]

Imports a screen format from file "infile" into the screen format file. The file must be a binary image of a screen format which was probably created via an export command or downloaded from a host TIP/30 system. Using the import and export commands, screen formats can be moved between TIP systems on TIP/ix or TIP/30.

An optional screen format name can be specified to indicate the name that the format is to be imported to. If a name is not specified, then the

name within the screen format in the file is used. If any help text identifiers associated with the screen format exist in the file, they will be added as well.

If the screen format already exists or any help text identifiers exist, then a prompt will be displayed asking to overwrite the format or help. If overwrite is selected, then the format and all associated help will be imported. An optional screen format name can be specified to indicate the name that the format is to be imported to. If a name is not specified, then the name within the import file will be used. The command is used as follows:

```
msgar i myfmt.fmt
- Imports the screen format in 'myfmt.fmt' to
  the screen format name given in the file
msgar i myfmt.fmt arc/fmt2
- Imports the screen format in 'myfmt.fmt' to
  the screen format 'ARC/FMT2'
```

IBms infile [name]

Imports an ASCII text file in Basic Mapping Support (BMS) style and generates a screen format.

If the screen format already exists, then a prompt will be displayed asking to overwrite the format. If overwrite is selected, then the format will be imported.

An optional screen format name can be specified to indicate the name that the format is to be imported to. If a name is not specified, then the name within the BMS text file will be used.

The command is used as follows:

```
msgar ibms myscr.bms
- Imports the BMS style text file 'myscr.bms' and
  creates a screen format named in the file
msgar ibms myscr.bms arc/bms2
- Imports the BMS style text file 'myscr.bms' and
  creates a screen format named 'ARC/BMS2'
```

IDps infile [name]

Imports an ASCII text file in DPS 1100 style and generates a screen format and corresponding DPS 1100 COBOL copybook. The generated copybook will include both an "FCA" and "DATA" copybook.

If the screen format already exists, then a prompt will be displayed asking to overwrite the format. If overwrite is selected, then the format will be imported.

An optional screen format name can be specified to indicate the name that the format is to be imported to. If a name is not specified, then the name within the DPS 1100 text file will be used. The command is used as follows:

```
msgar idps myscr.dps
- Imports the DPS 1100 style text file 'myscr.dps' and
  creates a screen format named in the file and the
  corresponding DPS 1100 copybook
msgar idps myscr.dps arc/dps2
- Imports the DPS 1100 style text file 'myscr.dps' and
  creates a screen format named 'ARC/DPS2' and the
  corresponding DPS 1100 copybook
```

IDPSIMS infile [name]

This command is similar to the "IDps" command except that the screen format and only the "DATA" copybook will be generated.

IDPSFCA infile [name]

No longer supported.

ISFG infile [name]

Imports a screen format in SFG style text from a file into the screen format file. The file must be in SFG style text in order for a screen format to be generated. You can specify an optional screen format name to indicate the name that the format is to be imported to. If you do not specify a name, the name within the SFG style text file is used.

If the screen format already exists or any help text identifiers exist, a prompt will be displayed asking to overwrite the format or help. If overwrite is selected, the format and all associated help will be imported. The command is used as follows:

```
msgar isfg myfmt.sfg
- Imports the SFG style text file in 'myfmt.sfg' to
  the screen format name given in the file.
msgar isfg myfmt.sfg arc/fmt2
- Imports the SFG style text file in 'myfmt.sfg' to
  the screen format 'ARC/FMT2'.
```

Following is a sample SFG text file:

```
* Dated 95/06/07 13:13
*
*
* TIP$Y$/TESTSFG has 0 error bytes and 45 data bytes
* BoundingBox is (24,36) Using 4 data fields
```

```

* Erase screen, Tab fields, Start row 01, Cursor rests
at (06,18)
*
NAME TIP$Y$/TESTSFG
START ROW 01
HEADING(03,05 "This is a test SFG screen format" LOW)
HEADING(04,05 "===== " LOW)
HEADING(06,08 "Field 1 :" LOW)
FIELD (06,18 FILLER 'X(16)' UPPER CURSOR NORMAL)
HEADING(07,08 "Field 2 :" LOW)
FIELD (07,18 FILLER 'X(16)' UPPER NORMAL)
HEADING(08,08 "Number :" LOW)
FIELD (08,18 FILLER '9(6)' NORMAL)
HEADING(09,08 "Dollar :" LOW)
FIELD (09,18 FILLER '$ZZ,ZZ9.99' NORMAL)
HEADING(11,05 "End of Test [" LOW)
HEADING(11,18 " " UNPROT TAB CHANGED NORMAL)
HEADING(11,19 "]" LOW)
FORMATEND (ROWS=13)

```

Xport pattern [outfile]

Export a screen format to a file. A binary image of the screen format and any help text associated with the screen format is copied to the file. The file can then be moved between any TIP/ix systems.

The Xport command does *not* preserve any sfs, description, dps, copybook, data dictionary, or field name information. If you need this information, use **msgar XSFG** instead.

The Xport file cannot be moved to TIP/30 because help text specifications are not supported on the host. To move a screen format to TIP/30, use **msgar XCport** instead.

Wildcard specification is allowed to export multiple screen formats. The output files created will be indicated by the format name followed by ".fmt".

For exporting single screen formats, an optional file name can be specified to indicate the output file. If an optional file name is not specified then the format name is used followed by ".fmt". The command is used as follows:

```

msgar X myfmt
- Exports the screen format 'TIP$Y$/MYFMT' to the
  file 'myfmt.fmt' along with any associated help
  text information
msgar X myfmt fmt.out
- Exports the screen format 'TIP$Y$/MYFMT' to the
  file 'fmt.out' along with any associated help
  text information
msgar X arc/*
- Exports all screen formats in the group 'ARC' to
  separate files given by the format name followed

```

by '.fmt'

XCport pattern [outfile]

Exports a screen format to a file in host compatibility mode of 2560 byte EBCDIC image with NO help text information. A binary image of the screen format will be copied to the file in EBCDIC format but any help text information associated with the screen format will NOT be copied to the file.

This command is used to move screen formats between a TIP/ix system and a TIP/30 system.

Wildcard specification is allowed to export multiple screen formats. The output files created will be indicated by the format name followed by ".fmt".

For exporting single screen formats, an optional file name can be specified to indicate the output file. If an optional file name is not specified then the format name is used followed by ".fmt".

The command is used as follows:

```
msgar XC myfmt
- Exports the screen format 'TIP$$/MYFMT' to the
  file 'myfmt.fmt' in 2560 byte EBCDIC image with
  NO help text information
msgar XC myfmt fmt.out
- Exports the screen format 'TIP$$/MYFMT' to the
  file 'fmt.out' in 2560 byte EBCDIC image with
  NO help text information
msgar XC arc/*
- Exports all screen formats in the group 'ARC' to
  separate files given by the format name followed
  by '.fmt' in 2560 byte EBCDIC image with NO help
  text information
```

XDps pattern [outfile]

Exports a screen format to a file in DPS 100 style text. Any help text associated with the screen format will not be written to the file.

Wildcard specification is allowed to export multiple screen formats. The output files created will be indicated by the format name followed by ".dps".

For exporting single screen formats, an optional file name can be specified to indicate the output file. If an optional file name is not specified then the format name is used followed by ".dps". The command is used as follows:

```
msgar XDps myfmt
```

- Exports the screen format 'TIP\$Y\$/MYFMT' to the file 'myfmt.dps' in DPS 100 style text.
msgar XDps myfmt fmt.out
- Exports the screen format 'TIP\$Y\$/MYFMT' to the file 'fmt.out' in DPS 1100 style text.
msgar XDps arc/*
- Exports all screen formats in the group 'ARC' to separate files given by the format name followed by '.dps' in DPS 1100 style text.

XSFG pattern [outfile]

Exports a screen format to a file in SFG style text. Any help text associated with the screen format will also be copied to the file which can then be moved between any TIP/ix or TIP/30 system.

However, if the screen format contains any help text information, the help text information will be lost when moving the file to a TIP/30 system. This is due to the fact that there are no help text specifications provided on the host OS/3 computer.

Wildcard specification is allowed to export multiple screen formats. The output files created will be indicated by the format name followed by ".sfg".

For exporting single screen formats, an optional file name can be specified to indicate the output file. If an optional file name is not specified then the format name is used followed by ".sfg". The command is used as follows:

- ```
msgar XSFG myfmt
```
- Exports the screen format 'TIP\$Y\$/MYFMT' to the file 'myfmt.sfg' along with any associated help text information in SFG style text
- ```
msgar XSFG myfmt fmt.out
```
- Exports the screen format 'TIP\$Y\$/MYFMT' to the file 'fmt.out' along with any associated help text information in SFG style text
- ```
msgar XSFG arc/*
```
- Exports all screen formats in the group 'ARC' to separate files given by the format name followed by '.sfg' in SFG style text

## UXport pattern [outfile]

## UXCport pattern [outfile]

## UXSFG pattern [outfile]

These commands are similar to the Xport, XCport and XSFG described above except that the group is used on output. These commands unload multiple screen formats with the intent of moving all of them to another TIP/ix or system.

On output, a directory will be created using the group name, if one does not already exist, and the output file will be put into this directory using the format name followed by either ".fmt" or ".sfg".

The purpose for the group specification is to distinguish between screen formats of the same name in different groups. Examples follow:

```
msgar UX myfmt
```

- Unloads the format 'TIP\$\$/MYFMT' to the file 'myfmt.fmt' in the directory 'TIP\$\$'. The output file will contain all associated help text information.

```
msgar UXC myfmt
```

- Unloads the format 'TIP\$\$/MYFMT' to the file 'myfmt.fmt' in the directory 'TIP\$\$'. The output file will be in 2560 byte EBCDIC image and contain NO help text information.

```
msgar UXSFG myfmt
```

- Unloads the format 'TIP\$\$/MYFMT' to the file 'myfmt.sfg' in the directory 'TIP\$\$'. The output file will be in SFG style text and contain all associated help text information.

```
msgar UX */*
```

- Unloads all screen formats in all groups to files given by 'format.fmt' in the 'group' directory. The output files will contain all associated help text information.

```
msgar UXC */*
```

- Unloads all screen formats in all groups to files given by 'format.fmt' in the 'group' directory. The output files will be in 2560 byte EBCDIC image and contain NO help text information.

```
msgar UXSFG */*
```

- Unloads all screen formats in all groups to files given by 'format.sfg' in the 'group' directory. The output files will be in SFG style text and contain all associated help text information.



## msgshow - Screen Format Testing

This program (via either of two transaction codes) is used to test TIP/ix screen formats:

### **msgtst**

Prompts the user for test data and presents it on the terminal using the named screen format.

### **msgshow**

Displays a specified screen format (without data), waits for the user to enter test data, then displays the data as the data would be received by a program using the screen format.

In either case, the unformatted data screen expects the test data to be a continuous character string. A user can cycle back and forth between screens trying various data entry options.

When the user's formatted message is displayed, intentional errors may be introduced to check error field options. Entering a circumflex as the first character in a field and pressing the **XMIT** key causes the field to blink and an error message to be displayed.

**msgtst** and **msgshow** display the data received as it would appear in a user program MCS-DATA area.

The representation of negative values for signed numeric fields in DISPLAY Format is different for MBP and MF COBOL. MCS returns the negative values in the representation appropriate for the compiler used with the application program.

On UNIX platforms that support MBP COBOL, we compile the **msgshow** program with MBP. If your applications are compiled with Micro Focus COBOL then the data returned to **msgshow** (for negative fields) will not match the values that would be returned to your program(s).

No header information or communications characters are received, and that the number of characters sent is a function of cursor position.

Numeric fields are returned to the program right justified and zero filled. Data characters entered into a field which are incompatible with the field definition are replaced by blink characters (or are blinked) by the Message Control System. The errors may be corrected and data changed to try out various options available. Simply place the cursor after the data and press the **XMIT** key.

### **Syntax:**

```
msgtst[,grp] format [,fill][,func]
msgshow[,grp] format [,fill][,func]
```

**Where:**

**grp** The group name associated with the screen format to be tested.

Default: action is to search through all of user's groups.

**format**

The name of the screen format to be tested.

**fill** The fill character to be used while testing the screen format. Choices are: asterisk, underscore, the word SPACE or NO to indicate a space fill character.

Default: underscore is used as fill character.

**func** The MCS-FUNCTION code (see description of this field in the MCS section of the TIP/ix Programming Guide). If not specified, MCS-FUNCTION is set to space.

**Example:**

```
msgshow TF$FSE03
```

---

## newpass - Change Password

The **newpass** transaction lets you change a password.

**Syntax**

```
newpass [user id]
```

**Where:**

user id

If a user id is not specified, your user id is assumed.

If you have at least MAST level security (9), you can change the password of another user.

If you do not have at least MAST level security, the user id parameter is ignored.

There are two ways to specify the user id whose password is to be changed:

- supply it as a parameter to newpass

- change the user id field on the Change Password screen.

When changing your password (the password of the current user), you must enter the current password to confirm that you are indeed the current user.

You must enter the new password twice (once in each of two fields) for confirmation of the new password value.

---

## note - Display Informational Message

The **note** program sends a message to the terminal which invoked the **note** program. The note command is useful for displaying messages during the execution of a command file.

This is like using the ECHO command in a .bat file in MS-DOS.

### Syntax:

```
note [,W] text
```

### Where:

**W** A command line option character to cause the note program to pause and wait for any input (usually a function key or the MSG WAIT key).

This option is often used in TIP redirected input files (sometimes called "command files").

The text is displayed and the note program waits for any input before continuing.

**text** The message text (64 characters maximum) that is to be displayed on the terminal.

### Example:

```
note all user-ids have been catalogued
note,w press msg-wait to continue
```

---

## pingtip - Report TIP/ix Processes

Check if TIP/ix processes are still running.

```
pingtip
```

A report is displayed.

---

## precob - COBOL Conversion Aid

The **precob** utility examines the DATA DIVISION of a COBOL program, and reports possible data migration issues. Optionally, **precob** may:

- Produce a modified source file.
- Report the LINKAGE SECTION sizes for online TIP/30 and IMS transaction programs.
- Update the TIP/ix definition of the transaction.

Some options are provided for use with other products:

- TIP/dbi (Database Interface).
- The armcob utility (part of the **Heritage Support Package**).

### Syntax

```
precob [-options] [-L [n]] [-S schema] filename
```

### Where:

#### -options

Code a single hyphen at the start of the optionlist, followed by a combination of the following letters:

**d** Do create output file. If you don't specify any other options, the output file will have the same contents as the input file.

The name of the output file depends on the type of input file (as determined by its contents).

**.cbl** On-line transaction.  
**.bat** Batch program.  
**.ims** IMS transaction.

If the input filename is the same as the output filename, the original file is replaced.

**f** Quiet mode. No messages.

**F** Do not process FD SELECT in batch programs.

**l** Translate copybook names to lowercase.

**n** Do not produce an output file — just scan and report the LINKAGE SECTION sizes for online TIP/30 and IMS transaction programs.  
 This is the default.

**o** For use with **armcob**. Convert OS/3 application.

**O** For use with **armcob**. Convert IMS/1100 or OS/1100 application.

**q** Quiet mode. No messages.

**u** Use Data Division Linkage Section sizes to *conditionally* update values in the program record defined to TIP/ix.

The **precob** utility calculates the various area sizes (such as CDA size, WORK size, etc.) and compares them to the sizes in the associated program definition record.

If the actual sizes are *larger* than those defined to TIP/ix, **precob** updates the definition. The values

that are updated will be adjusted so that the value is a multiple of 256.

- U** Use Data Division Linkage Section sizes to *unconditionally* update values in the program record defined to TIP/ix. The **precob** utility calculates the various area sizes (such as CDA size, WORK size, etc.), then updates the sizes in the record definition. The values that are updated will be adjusted so that the value is a multiple of 256.
- V** Verbose; generate informational messages.
- w** Enable warnings.
- x** Translate COBOL 85 words used in an old COBOL 74 program.

**-L [n]** List DATA DIVISION level sizes. If *n* is specified, list sizes up to level *n*. Otherwise, only list level 01. However, this option does not include any of the DMS record layouts that will get copied into the program during pre-processing in the WORK-AREA size.

Also see -u option.

**Note:** Although using -L does not include any DMS record layouts that will get copied into the program during pre-processing in its listing of the WORK-AREA, the DMS record size values are listed as the last line of the output from "precob".

#### **-S schema**

For use with TIP/dbi.

#### **filename...**

The names of the files to scan. You may specify these using standard Unix file naming conventions.

#### **Example of Scanning and Report Production:**

```
precob myprog.cbl
```

#### **Additional considerations:**

The **precob** program recognizes the COBCPY environment variable that defines the search path of directories for copybooks. For example:

```
COBCPY=/dir1/hdr;/dir2/hdr; etc...
```

---

## **purge – Remove Process**

The **purge** utility forces an abnormal termination of a **user program** and then terminate the user's TIP/ix session (by forcing an abort of the TIP/ix shell).

To abort a user program without terminating the TIP/ix session, use the **die** command.

**Syntax:**

```
purge [-F|-T|-U] [*]value
purge S=segnum
```

**Where:**

- F** Use the *value* parameter to match files only.
- T** Use the *value* parameter to match terminals only.
- U** Use the *value* parameter to match users only.

**value**

If the meaning of *value* is not restricted (by specifying -U, -T or -F), **purge** attempts to match the value four ways (user id, terminal, file, and program).

- \* Match anything that starts with specified value.
- ! Match anything that does not start with specified value.

**user id**

Abort all programs running at each TIP/ix session being operated by this user id and then terminate the TIP/ix session(s).

**terminal**

Abort all programs running at each TIP/ix session being operated at this terminal and then terminate the TIP/ix session(s).

**file**

Abort all programs running at each TIP/ix session where this file has been assigned (see **aft** or **whoson**) and then terminate the TIP/ix session(s).

**program**

Abort all programs running at each TIP/ix session where this program is running at the highest stack level of the TIP/ix session (see **whoson**) and then terminate the TIP/ix session(s).

**S = segnum**

Specify the seg number of a single specific session to be aborted. You can get the seg number with the **whoson** command.

**Examples:**

Purge the program being executed by user "john".

```
purge -U john
```

Purge all user ids, terminal names, file names, and program names that begin with the prefix "ABC". From TIP/ix:

**purge \*abc**

From a UNIX shell, you have to protect the \* from expansion by Unix:

**purge \\*abc**

If you try to match a program name, a match only occurs if the program you want is active at the highest stack level of a TIP/ix session.

For example, run **tcn** then select "user id definition". In this case **tcn** is running at stack level 1 and **smuser** is running at stack level 2.

Therefore, "**purge smuser**" would match this session but "**purge tcn**" would not.

**Error Conditions:**

The **purge** utility cannot find a matching user, terminal, file, or program.

**Additional considerations:**

If more than one TIP/ix session would be purged, **purge** lists them, and prompts you for confirmation.

---

## readjrn - Read TIP/ix Journal File

The **readjrn** Unix program allows you to display the contents of a TIP/ix journal file or a TIP/ix Quick Before Look (QBL) file. The **readjrn** utility is provided primarily to assist debugging.

**Syntax:**

**readjrn [options]**

**Where:**

**-c** Validate and correct a corrupted journal or qbl file. See additional considerations.

**-h** Help. Display possible options.

By default, **readjrn** reads the primary journal file, as specified by the first **JRNFILE** parameter in **tipix.conf**, or the default journal file **\$TIPROOT/tipfiles/tipixjrn**.

The following options may be used to read other files:

**-a** Examine the alternate journal file (as specified in the second **JRNFILE=** parameter in **tipix.conf**).

**-q** Examine all the TIP/ix QBL files specified in **tipix.conf**, or examine the default QBL file **\$TIPROOT/tipfiles/tipix.qbl**.

**-f file** Specify the name of a journal or QBL file to read.

By default, readjrn displays some key fields of each record header and the first few bytes of data. The following options may be used to display more data:

**-d size**

Dump the specified number of bytes of record data area.

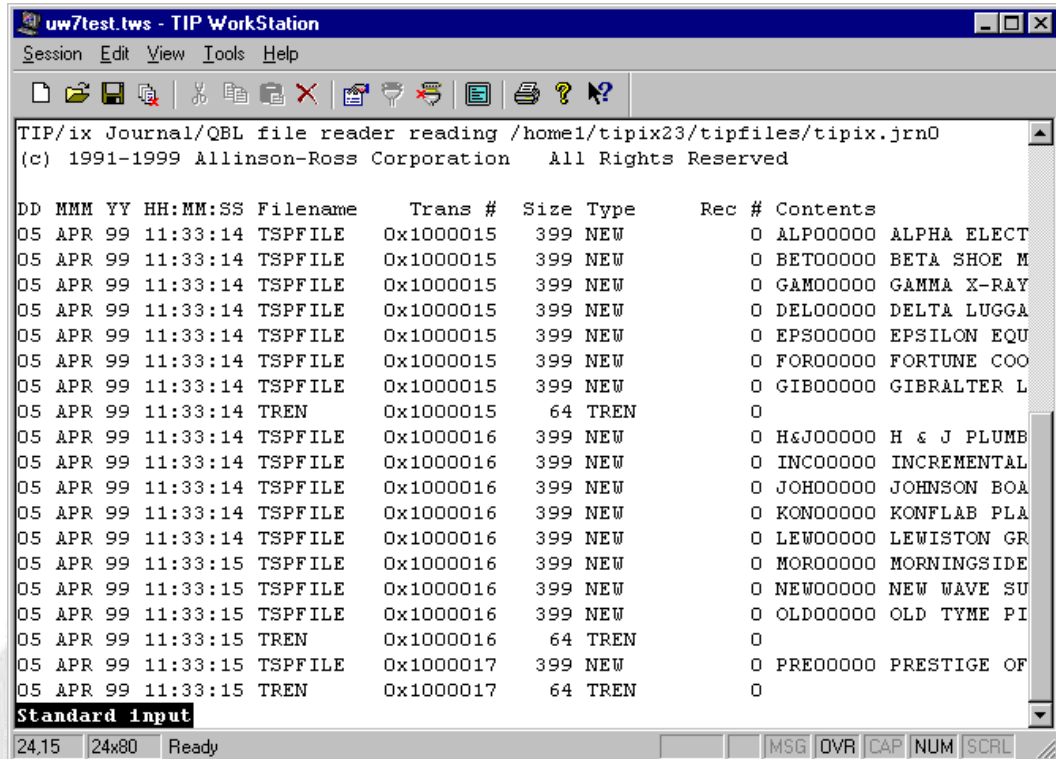
**-x**

Extended. Display all of the record header fields.

**Additional Considerations:**

The output may consist of many lines; suggested use is to pipe the output to the Unix "more" program. See example below.

Example of readjrn output:



```

TIP/ix Journal/QBL file reader reading /home1/tipix23/tipfiles/tipix.jrn0
(c) 1991-1999 Allinson-Ross Corporation All Rights Reserved

DD MMM YY HH:MM:SS Filename Trans # Size Type Rec # Contents
05 APR 99 11:33:14 TSPFILE 0x1000015 399 NEW 0 ALP00000 ALPHA ELECT
05 APR 99 11:33:14 TSPFILE 0x1000015 399 NEW 0 BET00000 BETA SHOE M
05 APR 99 11:33:14 TSPFILE 0x1000015 399 NEW 0 GAM00000 GAMMA X-RAY
05 APR 99 11:33:14 TSPFILE 0x1000015 399 NEW 0 DEL00000 DELTA LUGGA
05 APR 99 11:33:14 TSPFILE 0x1000015 399 NEW 0 EPS00000 EPSILON EQU
05 APR 99 11:33:14 TSPFILE 0x1000015 399 NEW 0 FOR00000 FORTUNE COO
05 APR 99 11:33:14 TSPFILE 0x1000015 399 NEW 0 GIB00000 GIBRALTER L
05 APR 99 11:33:14 TREN 0x1000015 64 TREN 0
05 APR 99 11:33:14 TSPFILE 0x1000016 399 NEW 0 H&J00000 H & J PLUMB
05 APR 99 11:33:14 TSPFILE 0x1000016 399 NEW 0 INC00000 INCREMENTAL
05 APR 99 11:33:14 TSPFILE 0x1000016 399 NEW 0 JOH00000 JOHNSON BOA
05 APR 99 11:33:14 TSPFILE 0x1000016 399 NEW 0 KON00000 KONFLAB PLA
05 APR 99 11:33:14 TSPFILE 0x1000016 399 NEW 0 LEW00000 LEWISTON GR
05 APR 99 11:33:15 TSPFILE 0x1000016 399 NEW 0 MOR00000 MORNINGSIDE
05 APR 99 11:33:15 TSPFILE 0x1000016 399 NEW 0 NEW00000 NEW WAVE SU
05 APR 99 11:33:15 TSPFILE 0x1000016 399 NEW 0 OLD00000 OLD TYME PI
05 APR 99 11:33:15 TREN 0x1000016 64 TREN 0
05 APR 99 11:33:15 TSPFILE 0x1000017 399 NEW 0 PRE00000 PRESTIGE OF
05 APR 99 11:33:15 TREN 0x1000017 64 TREN 0
Standard Input
24.15 24x80 Ready MSG OVR CAP NUM SCRL

```

Each record in the journal or QBL file has a specific header record containing a time stamp and a record length field. The record length is used to scan the file by record and verify each record's header values.

Under very rare circumstances it is possible for a journal or QBL file to become corrupted. The most common cause is for a disk full error during a write to the QBL or journal file, in which case a partial record may exist at the end of the file. If this does happen the readjrn utility can be used to correct the file by deleting the partial record. The **readjrn -c** option will scan the journal file and if it detects a discrepancy in the file it will prompt the user to proceed with correction. A backup of the file will be made.

If it is necessary to correct a QBL or journal file please proceed with caution. A serious disk error may result in a great deal of the file being



truncated. The **readjrn** utility may be used to dump the file contents beforehand so that the system administrator can examine record time stamps to determine how much of the file is valid. The number of bytes to be removed is displayed. This is important to verify that the file correction will do what is expected.

For information about Journal and QBL file usage, see **File Recovery** in the **TIP/ix Installation and Operation** manual.

For Journal and QBL file APIs and formats, see the **FCS-JOURNAL** and **Accessing TIP/ix Journal Files** sections in the **TIP/ix Programming Reference**.

---

## remove - Exit from Remote LOCAP

When TIP is operating with distributed transaction processing (DTP), connections to other TIP systems running on other LOCAPs can be established in a number of ways (see the description of TIP distributed transaction processing in the documentation of the TIP Program Control System - PCS, in our **TIP/ix Programming Reference** manual).

In some situations, an extended connection is required to allow the terminal user to execute a number of subsequent transactions at a remote LOCAP.

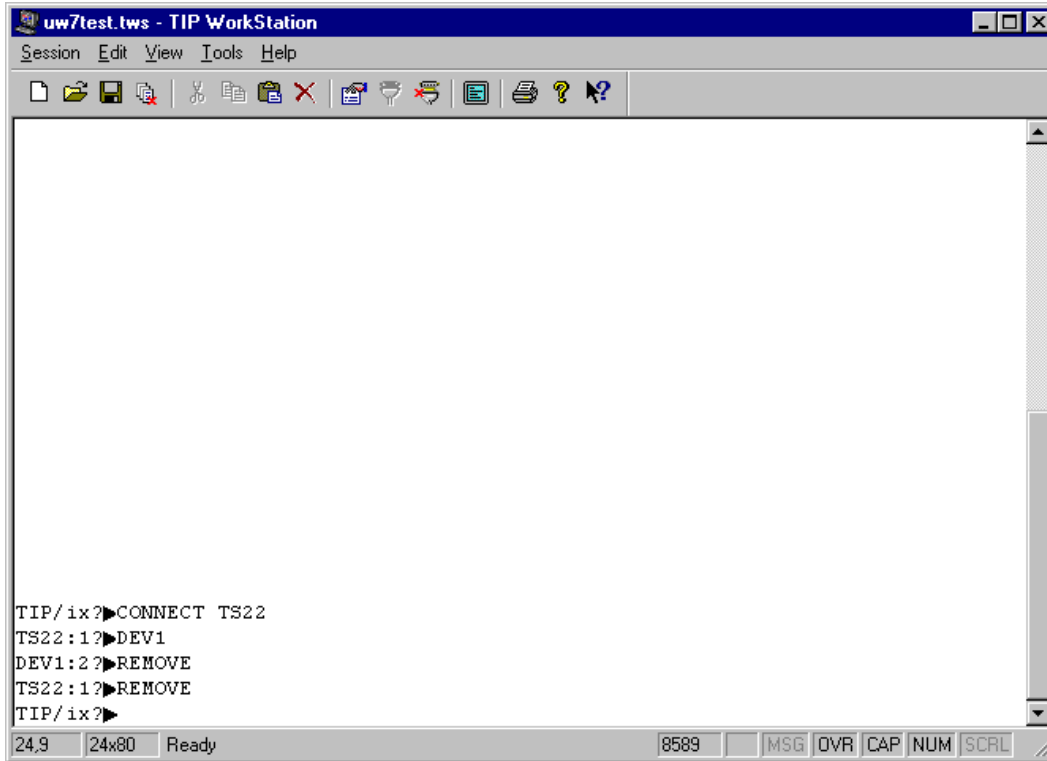
The REMOVE transaction is used to break a previously established connection to a remote LOCAP. The REMOVE transaction is the inverse of the [CONNECT](#) transaction (see separate documentation of that utility transaction).

### Syntax:

```
remove
```

### Example:

The following example illustrates a brief conversation with the LOCAPs named "TS22" and "DEV1".



```

uw7test.tws - TIP WorkStation
Session Edit View Tools Help
TIP/ix?>CONNECT TS22
TS22:1?>DEV1
DEV1:2?>REMOVE
TS22:1?>REMOVE
TIP/ix?>
24.9 24x80 Ready 8589 MSG OVR CAP NUM SCRL

```

The connection is also broken if the user logs off the (remote) LOCAP TIP system (using LOGOFF or FIN etc.) or if the physical connection between the two LOCAPs is broken.

## rfaxlt - Remote File Access Translation

The **rfaxlt** utility reads a *translation definition file* and generates a system file named **\$TIPROOT/conf/RFAXlate.tbl**.

At startup time, TIP/ix reads RFAXlate.tbl into memory to control conversion of data records between TIP/ix and TIP/30.

The following conversions are supported:

- BINARY
- PACKED
- signed numeric fields.

The following are not supported:

- KEYS which have non-character data
- REDEFINES clause.

**Syntax:**

```
rfaxlt filename
rfaxlt -d
```

**Where:**

filename      The translation definition file for this LOCAP. You create this file to specify: 1) a list of all the files that need to be translated, and 2) how to translate them.

-d      Dump the current RFAXlate.tbl for debugging purposes. The statements in the translation definition file are:

```
nnnnnn* Comments an have asterisk in col 7
nnnnnn COBOL [MICRO-FOCUS | MBP]
nnnnnn FILE filename
nnnnnn RAW fieldname
nnnnnn COPY copybook
```

The default COBOL is MICRO-FOCUS. You must specify MBP if this is being used. The declaration RAW indicates that the field is not to be translated at all.

The file names may be for local or remote files. TIP/ix does the data translation in all cases.

**Example:**

You have two files that need field translation:

TSPFILE      The record layout is in TC-TSP. Suppress translation on the field named CM-POSTAL.

INVEN The record layouts is in INREC.

Create the translation definition file and give it a name, say "filexlat".

```
000001* sample xlate
000002 COBOL MICRO-FOCUS
000003 FILE TSPFILE.
000004 RAW CM-POSTAL.
000005 COPY TC-TSP.
000006
000007 FILE INVEN.
000008 COPY COPY INREC.
000009
```

Run **rfaxlt** with the transaction definition file you just created.

```
rfaxlt filexlat
```

The **rfaxlt** program displays a report:

```
TIP/ix ver 95/07/05 2.1 Q3 - 0000 (c) 1991-
1995 Inglenet Business Solutions
TIP/ix Record definition compiler, 3.4
Reading `filexlat'
Applications compiled by Micro Focus COBOL
Writing remote file translation module:
/u/tipsrc/conf/RFAXlate.tbl
File: ISPFILe Size 325
Field Loc Len Type
..... . . PIC
..... . . PIC
File: INVEN Size 64
Field Loc Len Type
..... . . PIC
Completed
```

- Verify that \$TIPROOT/conf/RFAXlate.tbl was created.  
`ls -l $TIPROOT/conf/RFAXlate.tbl`
- Restart TIP/ix.  
`tipctl boot`

---

## rollfwd - Recover Lost Files

The **rollfwd** utility is an interactive program that allows you to recover lost or corrupted files that have been backed up to hard disk or magnetic (cartridge) tape. This is also referred to as Offline Recovery. This utility uses TIP/ix journal file(s) and command-line options to determine which files and which file changes should be rolled forward.

For an introduction to offline recovery (roll forward), see **File Recovery** in the *TIP/ix Installation and Operation* manual.

**rollfwd** is only able to recover files that you have specified you want journalized. Use [smfile](#) to request journalizing for a file by specifying “**Journal: Y**” in the file definition.

TIP/ix writes record changes to disk only. You must implement a tape backup procedure to protect the information on your hard disks.

The **rollfwd** utility is driven by the contents of at least one TIP/ix journal file. You must also specify which files to roll forward, and start and end date and time values. All changes made *after* the start date and time and *before* the end date and time will be re-applied. In general, the **rollfwd** utility prompts you for this information, providing default options, but it is possible to supply all this information at the command line. This is useful

for running the utility from a script, a technique that should be implemented with caution.

Once the files have been restored from the backup medium, you can invoke the Journal File Roll Forward program.

**Warning:**

When recovering files, either:

TIP/ix should not be running, or  
the files being recovered should be taken offline (with the [fclose](#) utility).

**Syntax:**

```
rollfwd [-h][-y][-i file] [-j journal file]
[-d startdate -t starttime]
[-e enddate -f endtime] [file1 [file2]&ldots;]
```

or

```
rollfwd [-h][-y][-i file] [-j journal file] -q
-d startdate -t starttime -e enddate -f
endtime file1 [file2&ldots;]
```

By default, **rollfwd** will use the journal files specified in *tipix.conf*, or the default journal file (\$TIPROOT/tipfiles/tipix.jrn).

**Where:**

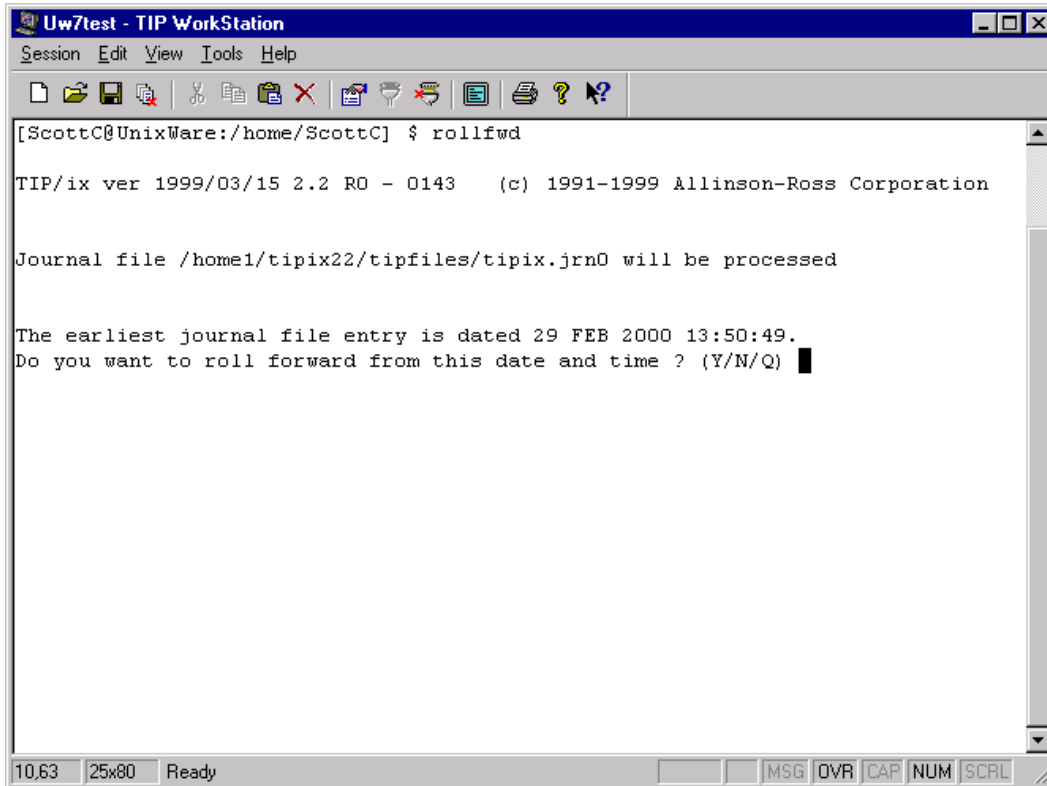
- h** This option will display the Help menu.
  - j** Specify the name of a journal file to process.
  - q** Quiet, used to disable rollfwd prompting.
  - y** Will auto answer Yes to prompts.
  - i** Will read a file where each line holds one TIP/ix file name to be rolled forward.
- If an invalid option is entered, the Help menu will be displayed.
- The quiet option makes the following options mandatory:
- d** Start date in the form DDMMMYYYY  
If either option d or t is supplied they must both be supplied
  - t** Start time in the form HH:MM or HH:MM:SS
  - e** End date in the form DDMMMYYYY  
If either option e or f is supplied they must both be supplied
  - f** End time in the form HH:MM or HH:MM:SS

**file&ldots**

a list of files on which to perform recovery

**Example:**

When you enter the **rollfwd** command, at your Unix shell prompt without any options, you will see something like this:



```
Uw7test - TIP WorkStation
Session Edit View Tools Help
[ScottC@UnixWare:/home/ScottC] $ rollfwd
TIP/ix ver 1999/03/15 2.2 RO - 0143 (c) 1991-1999 Allinson-Ross Corporation

Journal file /home1/tipix22/tipfiles/tipix.jrn0 will be processed

The earliest journal file entry is dated 29 FEB 2000 13:50:49.
Do you want to roll forward from this date and time ? (Y/N/Q) █
```

Type "**N**" if you know when the backup tape was made; **rollfwd** will prompt you to enter the date and time to roll forward from.

You must supply a time that is **PRIOR** to the time the backup was made!

Type "**Y**" if you are not certain when the backup tape was made. Roll forward will occur from the beginning of the journal file. If possible, verify that the date of the first entry in the journal file is earlier than your backup.

**rollfwd** will then prompt you:

```

Uw7test - TIP WorkStation
Session Edit View Tools Help
[ScottC@UnixWare:/home/ScottC] $ rollfwd

TIP/ix ver 1999/03/15 2.2 R0 - 0143 (c) 1991-1999 Allinson-Ross Corporation

Journal file /home1/tipix22/tipfiles/tipix.jrn0 will be processed

The earliest journal file entry is dated 29 FEB 2000 13:50:49.
Do you want to roll forward from this date and time ? (Y/N/Q) y

Do you want to roll forward to the end of selected journal file(s)? (Y/N/Q)

```

Type "N" if you want to roll forward to a specific point in time before the end of the journal file; **rollfwd** will prompt you to enter the date and time to roll forward to.

Next, **rollfwd** will prompt you:

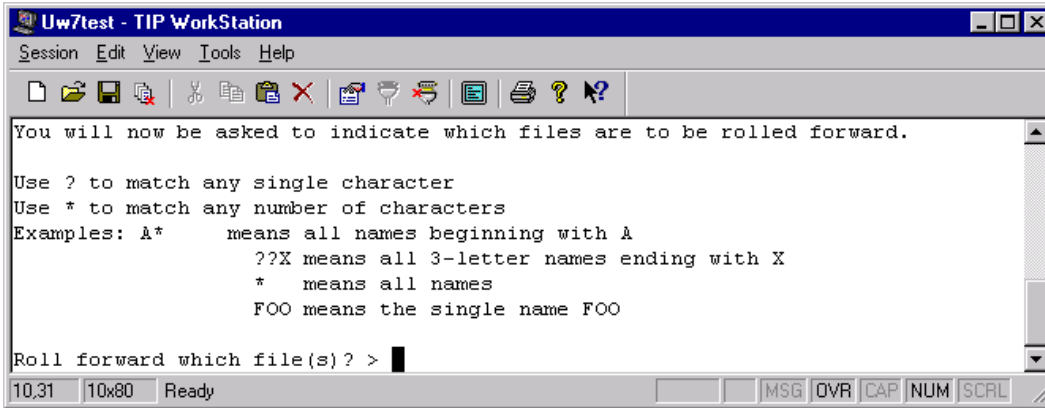
```

Uw7test - TIP WorkStation
Session Edit View Tools Help
Do you wish to roll forward all files? (Y/N/Q)

```

Type "Y" if you have restored *all* of your TIP/ix files from tape; type "N" if you have restored some but not all files.

If you type "N", you will receive the following prompts:



```

Uw7test - TIP WorkStation
Session Edit View Tools Help
[Icons]
You will now be asked to indicate which files are to be rolled forward.

Use ? to match any single character
Use * to match any number of characters
Examples: A* means all names beginning with A
 ??X means all 3-letter names ending with X
 * means all names
 FOO means the single name FOO

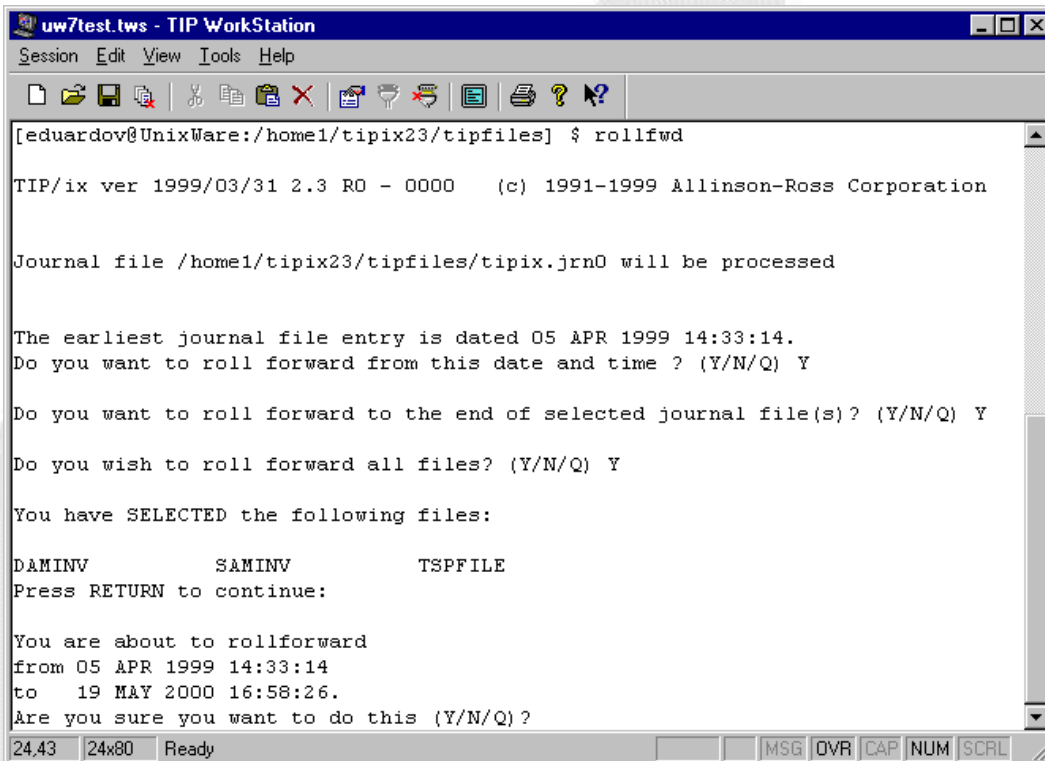
Roll forward which file(s)? >
10.31 10x80 Ready [MSG] [OVR] [CAP] [NUM] [SCRL]

```

Follow these instructions to indicate which files you want restored. Each time you type a logical file name (for example: "INVEN") or pattern (for example: "\*DATA") **rollfwd** will ask you whether you want to add to the list of names. When you have finished listing all the files you want rolled forward, type "N".

**rollfwd** warns you about files that cannot be rolled forward. These files are either cataloged as read-only or were not journalized.

**rollfwd** lists the files that you have requested and prompts you to confirm that you want to perform the Roll Forward procedure.



```

uw7test.tws - TIP WorkStation
Session Edit View Tools Help
[Icons]
[eduardov@UnixWare:/home1/tipix23/tipfiles] $ rollfwd

TIP/ix ver 1999/03/31 2.3 RO - 0000 (c) 1991-1999 Allinson-Ross Corporation

Journal file /home1/tipix23/tipfiles/tipix.jrn0 will be processed

The earliest journal file entry is dated 05 APR 1999 14:33:14.
Do you want to roll forward from this date and time ? (Y/N/Q) Y

Do you want to roll forward to the end of selected journal file(s)? (Y/N/Q) Y

Do you wish to roll forward all files? (Y/N/Q) Y

You have SELECTED the following files:

DAMINV SAMINV TSPFILE
Press RETURN to continue:

You are about to rollforward
from 05 APR 1999 14:33:14
to 19 MAY 2000 16:58:26.
Are you sure you want to do this (Y/N/Q)?
24.43 24x80 Ready [MSG] [OVR] [CAP] [NUM] [SCRL]

```

During the **rollfwd** operation, status and update information will be presented. Summary information will also be displayed for each journal file processed.



```

uw7test.tws - TIP WorkStation
Session Edit View Tools Help
[Icons]
Do you wish to roll forward all files? (Y/N/Q) Y

You have SELECTED the following files:

DAMINV SAMINV TSPFILE
Press RETURN to continue:

You are about to rollforward
from 05 APR 1999 14:33:14
to 19 MAY 2000 16:58:26.
Are you sure you want to do this (Y/N/Q)? Y

Reading journal file /home1/tipix23/tipfiles/tipix.jrn0
Processing updates...
Processing entry dated Fri May 19 15:37:06 2000: 88 records read... o
Summary:
 107 journal records read
 86 records processed
 16 NEW records
 70 AFTER records
 0 DELETE records

Rollforward complete
[eduardov@UnixWare:/home1/tipix23/tipfiles] $
24.47 24x80 Ready [MSG DVR CAP NUM SCRL]

```

Once the procedure is complete, you may safely start up the TIP/ix system.

If dual journalizing is configured (with two JRNFILE= parameters in the **tipix.conf** file), both journal files are used in the roll forward. The journal file that has the oldest internal time stamp is processed first.

---

## scratch - Erase Edit/Dynamic Files

The **scratch** program allows you to erase TIP/ix edit buffers or TIP/ix dynamic files.

### Syntax:

```

scratch [,q] [group] buffer
scratch [,q] dynamic

```

### Where:

,q The “q” option is used to request “quiet” mode. Scratch will not display any messages. This can be useful when calling scratch (via TIPSUB) from a user program.

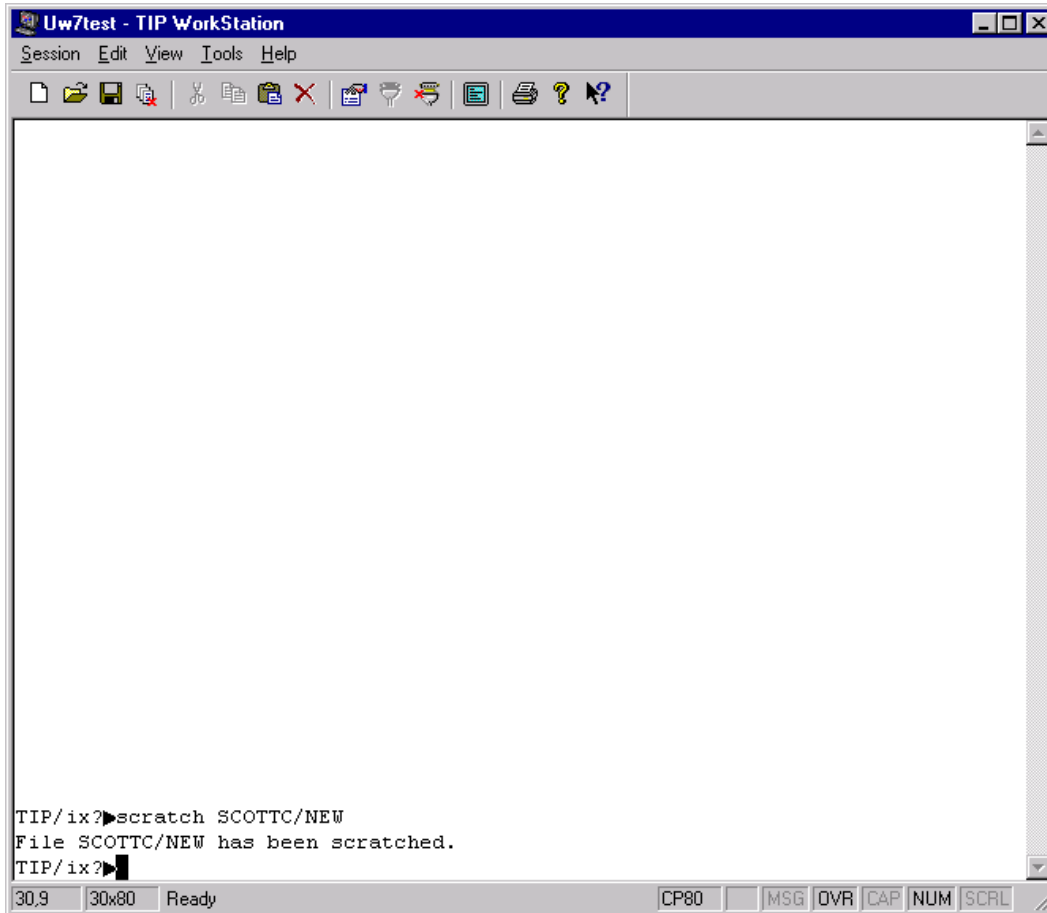
group Group that the edit buffer to be scratched belongs to. Use the “status e” command to obtain a list of edit buffers.

Default is the user’s 1st elective group.

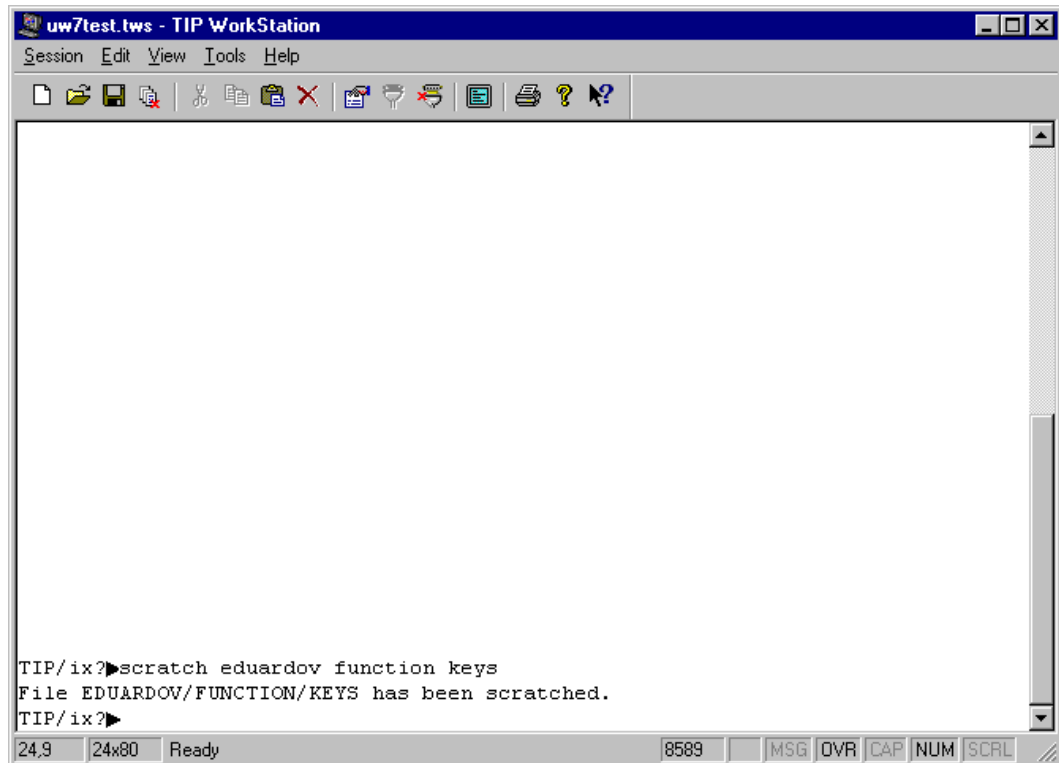
**buffer** The name of the edit buffer to be scratched.

**dynamic** The name of the dynamic file to be scratched. The name consists of three portions: user id, Catalog, and Filename. Use the “status d” command to obtain a list of dynamic files.

### Example of scratching an edit buffer:



### Example of scratching a dynamic file:



### Example of calling scratch from a user program:

```
MOVE " " TO CDA
MOVE "Q" TO CDA-OPTIONS
MOVE "EDP FUNCTION KEYS" TO CDA-TEXT
CALL "PARAM" USING CDA-PARAMETERS
CDA-TEXT
MOVE "SCRATCH" TO PIB-TRID
CALL "TIPSUB"
```

---

## script - execute a script from TIP/ix

This utility lets you execute a command file from the TIP/ix command line.

#### Syntax:

```
script name
```

#### Where:

**name** The ASCII file containing the commands to be executed.

The file can invoke any TIP/ix transactions you are authorized to run. If you are authorized to run UNIX

programs from TIP/ix, the file can also invoke UNIX programs and scripts.

TIP/ix does not require UNIX execute permission to the file. All that is required is read permission.

Normally, once all the transactions in the command file have finished, TIP/ix displays its command line prompt and awaits your next command. Of course, if the command file contains a FIN command, TIP/ix will terminate.

## seqreorg – Reorganize sequential files

This is a batch utility (for use from the Unix command shell) and can be used to copy and/or change the file format if sequential data files. The command usage information follows:

```
TIP/ix ver 2013/05/12 2.5 R0-0246 © 1991-2013 Inglenet Business Solutions
Re-Organize a Sequential/Relative file
 seqreorg [-opts] filename
where:
 -i inputfile File to copy
 -i @inputfile 2200 style filename; Checks for TIPECLQUAL env var
 -o outputfile Default: overwrite input
 -o @outputfile 2200 style filename to write over; Check for TIPECLQUAL
 -I format Input file type
 -O format Output file type
 MF = Micro Focus; OC = OpenCOBOL or COBOL-IT
Where 'format' is one of:
 rel - MF RELATIVE (fixed with LF)
 rec - MF/OC RECORD SEQUENTIAL (fixed no LF, FCSS)
 line - MF/OC LINE SEQUENTIAL (trailing spaces removed)
 vline - MF/OC LINE SEQUENTIAL (trailing spaces added)
 print - MF/OC LINE ADVANCING (Print report)
 linec - MF LINE SEQUENTIAL, BINARY data (spaces removed)
 vlinec - MF LINE SEQUENTIAL with BINARY data (spaces added)
 ftp - Ascii text file recently FTPed from 2200
 (Translate NULs to SPACES)
 ocdam - OC Direct Access Method (RELATIVE)
 varseq0 - OC Variable length Sequential format 0
 varseq1 - OC Variable length Sequential format 1
 varseq2 - OC Variable length Sequential format 2
 varseq3 - OC Variable length Sequential format 3
 -r n Input file Record size
 -R n Output file Record size; Default to input size read
 -T n Maximum Output Record size; (trunc if needed, implies -c)
 -c Verify data is valid ASCII printable
 -F name FCS file name to read (get -i, -r & -I from TIP/ix)
 -d Dump file to stdout in hex
 -m nn Start processing input at record number 'nn'
 -M nn Stop processing input at record number 'nn'
To convert 'line' into 'vline' format define record types and lengths
 -f pos,id,len pos is 1 relative position of 'id'
 id is any character string to look for
 len, if 'id' is found in 'pos' then pad to 'len' bytes
 -w Write output with CR LF (MSDOS) instead of LF (Unix)
```

The input and/or output file names may be a complete path name or just a file name if in current directory.

If the file name starts with an at-sign '@' then the name is taken to be a 2200 style file name. The TIPHOMEDATA environment variable defines the base directory where the data files are stored. You can use names like MYQUAL\*FOOBAR, MYQUAL\*FOOBAR(+1) for the next cycle of the file, or just FOOBAR or FOOBAR(-1). If the file qualifier is omitted then the environment variable TIPECLQUAL defines the default qualifier to use. Due to the \* and brackets using 2200 file names you will likely have to enclose the file name in quotes to avoid the Unix shell doing something you do not intend. Example -o '@ME\*MAST(+1)'

---

## SHUTDOWN - Shutdown Processing

The SHUTDOWN transaction is a mechanism whereby the system administrator can schedule one or more transactions that are to be run while the TIP/ix system is shutting down in an orderly manner (see description of the [EOJ](#) command).

The SHUTDOWN transaction is simply a TIP/ix script file named *SHUTDOWN* stored in the TIPROOT directory. This script file may contain several transactions to be scheduled for execution when the TIP/ix system is shutting down. Each line in the script file will be executed as if the line was a standard TIP/ix command line.

If the SHUTDOWN script exists, it will be executed at system [EOJ](#) time after all users have exited the TIP/ix system. At that point in time, no more user logins will be accepted.

All transaction programs executed in the script file will be running in the background via TIPFORK. Thus any transaction that the SHUTDOWN script performs *must* be capable of running in the background (that is, it should not solicit screen input). SHUTDOWN uses the user id SHUTDOWN.

---

## smfile - File Definitions

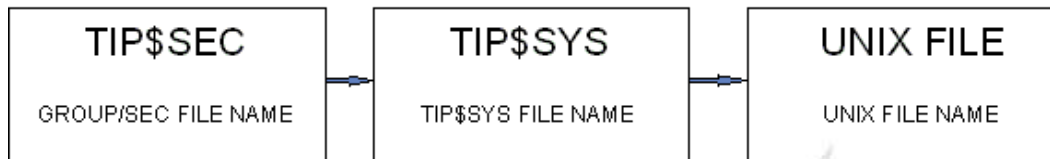
The **smfile** transaction is used to define data files that may be accessed by TIP/ix transaction programs (and IMS/90 and IMS/1100 transaction programs). Every file that is to be used by transaction programs must be defined to TIP/ix. These file definitions are kept in a file known to TIP/ix as TIP\$SYS.

TIP/ix transactions use the TIPFCS (TIP File Control System) interface to access files. In this interface file names are defined to be 1 to 8

uppercase characters. TIP/ix uses security and file definition information to relate the 8 character TIPFCS file name to the actual UNIX file to use.

When referencing a file with TIPFCS the name must match an entry in the TIP/ix security file (TIP\$SEC). Each file entry in the security file references an entry in the TIP/ix configuration file (TIP\$SYS) which in turn defines the UNIX file to use.

## ASSOCIATING TIPFCS FILE NAME TO UNIX FILE



The above association also applies to programs running under IMS emulation since requests to access files from IMS programs are converted into the appropriate TIPFCS request.

File look up in the security file is based on the user's access which is determined by the groups and security level(s) assigned to the user. Refer to [smsec](#) and [smuser](#) for more details.

**smfile** must be run from within the TIP/ix shell.

### Syntax:

```
smfile [function] [filename]
```

If a valid function is supplied then **smfile** will perform that function and then terminate when that function has been completed. Most functions require a file name to act upon.

If no function is supplied then **smfile** displays the following file maintenance menu screen.

From this menu you may perform all the traditional maintenance functions on the file definitions. You may add new file definitions; change, delete, show or print current file definitions and list a summary of all current file definitions.

In the preceding screen, the function to be performed can be abbreviated to the portion shown in upper case. For example the function *add* can be entered as "A".

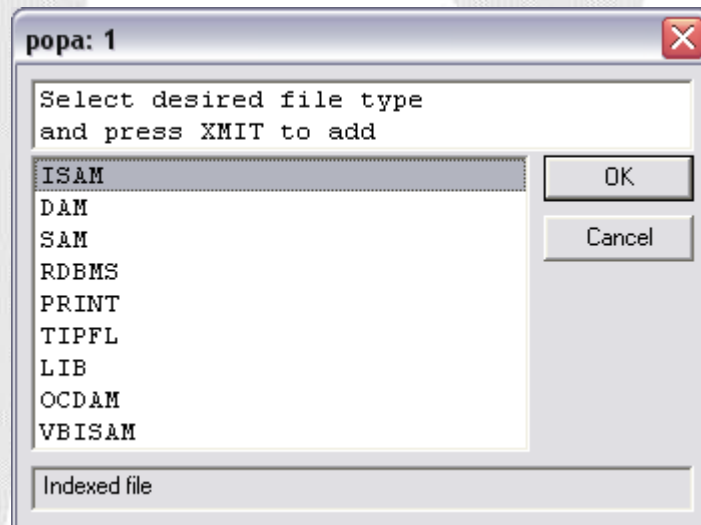
## Add a File Definition

To define a new file definition, enter “A” for the function to be performed (Add) and the name of the file (“NEWFILE” in our example). After pressing **XMIT**, you will be prompted to select the type of file to add from a list of valid file types. The file types available on TIP/ix are:

| FILE TYPE | DESCRIPTION                                                                                                                                                          |
|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ISAM      | Indexed file. Access records using record key(s).                                                                                                                    |
| DAM       | Direct access file. Access records by record number.                                                                                                                 |
| SAM       | Sequential access file. Records accessed in sequence from the beginning of file.                                                                                     |
| RDBMS     | Relational Database file. Requires TIP/dbi product to provide interface to database files.                                                                           |
| PRINT     | Print file. Defines routing and processing associated with TIPPRINT destinations and IMS print files. Refer to smprint for more information on defining print files. |
| TIPFL     | 2200 HVTIP data file. Defines a special type of direct access file used by the FCSS API. The file name will be just the FCSS file number.                            |
| OCDAM     | Direct access file used by OpenCOBOL and COBOL-IT.                                                                                                                   |
| VBISAM    | Indexed sequential file in VB-ISAM 2.1 format. This is the format used by COBOL-IT for batch programs.                                                               |

Move the cursor to the desired file type and press either of ENTER, XMIT or OK.

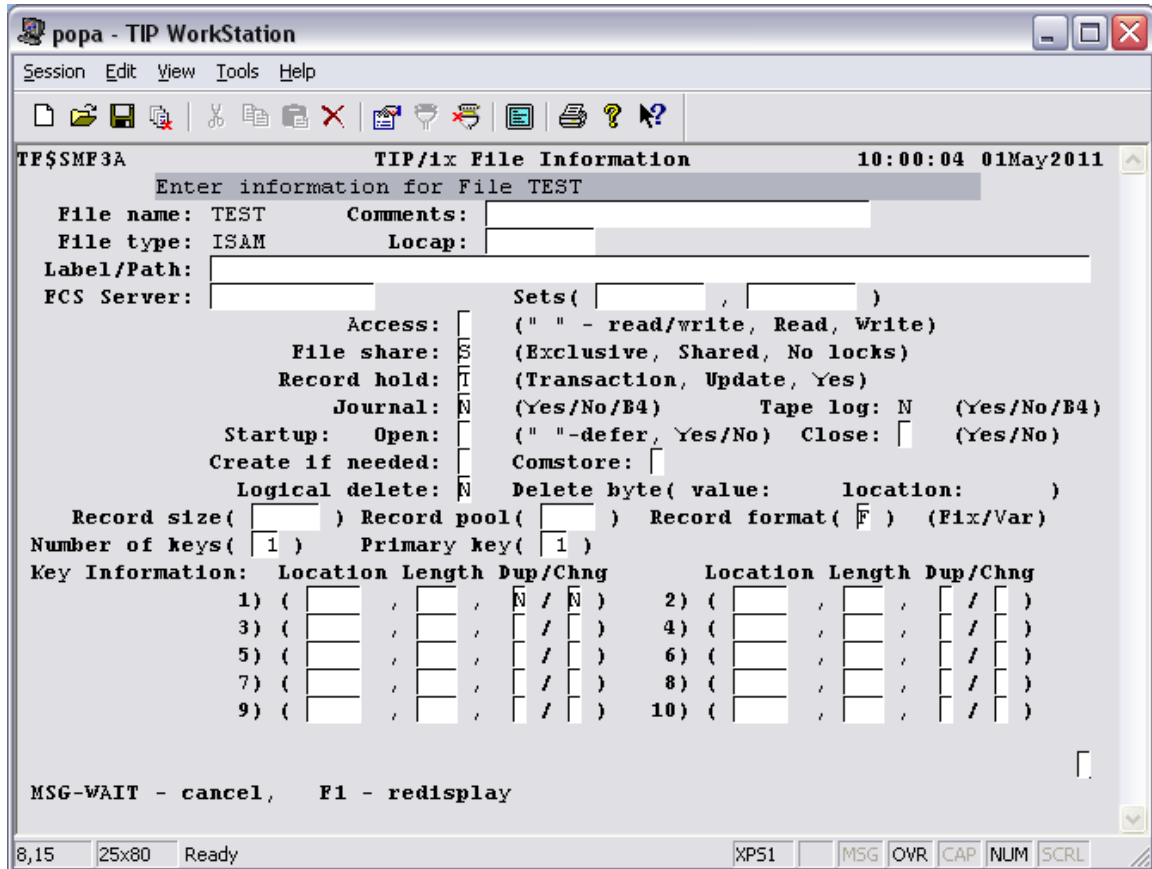
### Selecting a file type:



After you have selected the file type a screen will be displayed to allow you to enter the file definition information applicable to the file type you have selected.

### Defining an Indexed File

The following screen is displayed when a request is made to add, display, change, or delete an ISAM (indexed) file definition.



The screenshot shows a window titled 'popa - TIP WorkStation' with a menu bar (Session, Edit, View, Tools, Help) and a toolbar. The main area displays 'TIP/ix File Information' for file 'TEST' at 10:00:04 on 01May2011. The screen prompts the user to 'Enter information for File TEST' and lists various fields for configuration, including file name, type, label/path, FCS server, access permissions, file share, record hold, journal, startup, create if needed, logical delete, record size, record pool, record format, number of keys, and primary key. A 'Key Information' table is also present, showing 10 keys with their locations, lengths, and duplicate/change status. At the bottom, there are instructions: 'MSG-WAIT - cancel, F1 - redisplay'.

| Key Information: | Location            | Length | Dup/Chng  | Location | Length              | Dup/Chng  |
|------------------|---------------------|--------|-----------|----------|---------------------|-----------|
| 1)               | ( [ ] , [ ] , [ ] ) | [ ]    | [ ] / [ ] | 2)       | ( [ ] , [ ] , [ ] ) | [ ] / [ ] |
| 3)               | ( [ ] , [ ] , [ ] ) | [ ]    | [ ] / [ ] | 4)       | ( [ ] , [ ] , [ ] ) | [ ] / [ ] |
| 5)               | ( [ ] , [ ] , [ ] ) | [ ]    | [ ] / [ ] | 6)       | ( [ ] , [ ] , [ ] ) | [ ] / [ ] |
| 7)               | ( [ ] , [ ] , [ ] ) | [ ]    | [ ] / [ ] | 8)       | ( [ ] , [ ] , [ ] ) | [ ] / [ ] |
| 9)               | ( [ ] , [ ] , [ ] ) | [ ]    | [ ] / [ ] | 10)      | ( [ ] , [ ] , [ ] ) | [ ] / [ ] |

At the bottom of the screen is a summary of function keys and their actions.

At any time you may enter MSG WAIT to cancel the function and return to the main menu screen.

Pressing F1 causes the screen to be re-displayed (Caution! any data entered on the screen will be lost).

Fill in the information for the file definition and press XMIT to add the definition to the TIP\$SYS file.

If the file definition is added successfully and no security record exists for the file (in TIP\$SEC) then a screen will be displayed prompting for the security information. Refer to [smsec](#) for details about security records.

The following is a description of each field in the preceding screen.



**File name:**

The name of the file definition in the TIP\$SYS file. This is a logical name that will be referenced from a security record in the TIP\$SEC file. This name can be different from the physical file name.

**Comments:**

Optional information describing the file.

**File type:**

Supply one of the various types of files TIP/ix supports:

**ISAM** Indexed  
**DAM** Direct access  
**SAM** Sequential files  
**LIB** Library files  
**RDBMS** Database record or table definition

**LOCAP**

The LOCAP name of the TIP system where the file is to reside (see [smlocap](#) utility).

It is only necessary to fill in this field if the file is to be accessed from another TIP system. In this case the file is considered to be a remote file.

If the LOCAP name is left blank or is the LOCAP name for the current TIP system then the file is considered to be local.

If this field is filled in and you exit the field or press XMIT then if the LOCAP does not match the current TIP system then a remote file definition screen will be displayed.

**Label/Path**

Specifies where the physical file is located. Leaving this field blank causes TIP/ix to look for the file name in the directory *\$TIPROOT/tipfiles*. The file name used is the logical file name as specified in the File name field above.

If the physical file name is the same as the logical file name, except that it is in lower case, you must specify the name in lower case in the Label/Path field.

If the file exists in a directory other than *\$TIPROOT/tipfiles*, you must specify the full path and name of the file.

One method of specifying a full path is to use an environment variable to specify the UNIX directory and then append to it the UNIX file name. The environment variable must be defined in the *tipix.conf* configuration file in the *\$TIPROOT/conf* directory.

For example:

In *tipix.conf* define: **HR=/apps/files/hr.**

Enter Label as: \$HR/newfile

### FCS Server

Specify the appropriate file server (based on the ISAM format of the file) that TIP/ix will use to access the file.

The default file server is **tipfcs**.

**tipfcs** Using C-ISAM. The tipfcs file server supports C-ISAM file structures as implemented by the D-ISAM product of Byte Design Ltd. (also Micro Focus ISAM compatible).

**-N** TIP/ix handles the values of a negative sign followed by a number. i.e. -1, -2, -3...  
When TIP/ix encounters files with these values in the FCS Server field, TIP/ix will start a new TIPFCS process if it has not encountered the specific number in a previous file entry. If TIP/ix does encounter a number it already has seen it will add the file to the process started for the previous file. The system continues to add files to the process created for the specific number until it reaches the limit of number of files to be assigned to one process which is defined by the FCSFILES PARAM statement in the tipix.conf file. When the number of files exceed the limit TIP/ix will start another process.

### "string"

Users can enter any string that they want to (i.e. 'afiles', 'histfiles', 'newfile') and when they do this they must have a file in their path or TIP/ix's path called "string" but it must be linked to an ISAM file handler. (i.e. tipfcs, or mbpfcs) These are the only 2 ISAM file handlers that TIP/ix supports. This method may provide clarity for some users or system administrators but the first method is the recommended way. The problem with this method is that if the link to the ISAM file handler is lost or removed that the system may not boot.

**Sets** This entry is used to identify a set of records in the TIP\$SYS file that are in some way related. For example you could mark all TIP\$SYS records used in a payroll application by filling in the set name with "PAYROLL". The intention is that these records can be migrated as a set from one TIP/ix system to another using the "tippack" transaction.

### Access:

Defines the type of file access that TIP/ix programs will be allowed.

**<blank>**

TIP/ix transactions may read or write records from this file.

**R** TIP/ix transactions may only read records from this file. No updates will be allowed on this file.

**W** TIP/ix transactions may only write records to this file. Currently this access type does not restrict reading of records on most UNIX systems.

**File Share:**

Defines the type of access to the file at the D-ISAM or Visual COBOL ISAM level. This determines the type of concurrent access permitted by TIP/ix (online) programs and non-TIP/ix programs (batch).

**E** This provides TIP/ix with the "Exclusive" access to the file. Batch (non-TIP/ix) programs will not be able to access the file if it is open to TIP/ix. TIP/ix opens the file with the ISAM flag ISEXCLLOCK set.

**S** This allows access to the file from both TIP/ix and batch (non-TIP/ix) programs. In this case all record locking is coordinated by the ISAM (Either Visual COBOL ISAM or D-ISAM) file handler. File access may suffer due to the overhead of requesting locks through the ISAM file handler. Be aware that file recovery is unreliable if this mode is used to perform simultaneous updates to a file from both batch and TIP/ix programs. Since TIP/ix is not aware of any updates that occur in batch the use of TIP/ix rollback and recovery features may result in data integrity problems. If you are convinced that you need to share file access between batch and online programs, set this field to "Shared" and exercise caution.

**N** This is the same as Shared above except that TIP/ix handles all record locking internally and does not request record locks from the ISAM file handler.

**Record Hold****Transaction**

Lock records for the duration of the transaction. This is the recommended record holding specification and allows complete online rollback for the file. This prevents partial transaction updates from occurring.

**Update**

Allows multiple locks per file, but each lock is released after the corresponding record is updated. Online rollback is not supported for files with hold for Update.

**Yes** Single record lock per file; lock released upon update. No file updates are rolled back if a transaction aborts or requests rollback.

For a further discussion on TIP/ix record locking techniques, see the File Control System section of the TIP/ix Programming Reference manual.

**Journal:**

Allows each update to be recorded in a journal file. This can be useful to track changes to a file (see description of the [readirn](#) utility) or to perform off-line recovery (see description of the [rollfwd](#) utility). If journal records are kept for a file then when the file has been restored from a backup copy you have the ability to apply all updates to the file since the backup was created.

**Tape log:**

Not currently supported. TIP/30 allowed journal data to be written to a tape file (log tape).

**Open:**

**<blank>**

Defer opening the file until the first request to use the file.

**Yes** TIP/ix will attempt to open the file at system startup.

**No** The file is only opened when needed and closed when the application that requested the "Open" ends.

**Close:**

**Yes** The file is not available to TIP/ix applications.

**No** The file is available to TIP/ix applications. This is the **default** setting.

**Create if needed:**

**Y** Create the file if it does not already exist.

**N** Do not create the file.

**Comstore:**

**Y** The file is loaded into memory when it is first opened. Updates are only done in memory and nothing is written back to disk.

**N** This is a normal file.

**W** The file is loaded into memory, but updates are written back to disk.

If Y or W is specified then no records may be added to the file.

**Logical delete:**

- Yes** Requests to delete records (via TIPFCS) will result in TIP/ix updating the recording by setting a delete byte in the record.  
No delete request is issued to the ISAM handler.  
*Deleted* records remain valid records in the file so it is possible for both TIP/ix and batch programs to access deleted records
- No** Requests to delete records will be passed through to the ISAM handler for the file. Deleted records cannot be accessed by either TIP/ix or batch programs. This is the default setting.

**Delete byte value:**

This field can only be filled in if Logical delete is set to Yes. This is the value that TIPFCS will use to flag the record as deleted. Fill in either a single (displayable) character or 2 hexadecimal digits. The hexadecimal representation is useful when the delete byte is a non-displayable character.

Default is FF (HIGH-VALUES)

**Delete byte location:**

This field can only be filled in if Logical delete is set to Yes. The zero relative location of the delete flag byte in the record. This is the number of bytes that precede the delete flag byte in the record.

**Record Size:**

Specify (in bytes) the physical size of each record in the file. If the file is created by TIP/ix then this value is used to establish the record size. If the file is created outside of TIP/ix (batch COBOL for example) then this value is not used.

**Record Pool:**

This field is not currently used.

**Record Format:**

- F** Fixed-length records.  
**V** Variable length records.

**Number of keys:**

Specify the number of keys in the record. TIP/ix currently supports up to 10 keys.

**Prime key is:**

The number of key that is to be considered the primary key. Default value is 1 (1st key).

**Key Information:**

The description and attributes of all keys in the file. When

defining an existing ISAM file this should match the key information for the file.

For C-ISAM files the key information for a file can be displayed by running the UNIX command “dcheck -h unixfilename”. Note that location, length, and duplicate attributes are common to both the TIP/ix file definition and the ISAM file definition. The Chng attribute applies only to the TIP/ix environment.

**Location**

The zero relative location of the key. This value is the same as the number of bytes preceding the key.

**Length**

The length of the key in bytes.

**Dup** “Y” means that duplicates are allowed. This means multiple records can have the same value for this key.

The primary key must not allow duplicates (value must be “N”).

**Chng** “Y” means that key value can be changed on a record update.

The primary key must not allow changes (value must be “N”).

**Defining a Direct File**

The following screen is displayed when a request is made to add, display, change, or delete an DAM (direct) file definition.

**popa - TIP WorkStation**  
 Session Edit View Tools Help  
 TF\$SMP4A TIP/ix File Information 10:00:44 01May2011  
 Enter information for File TEST  
 File name: TEST Comments: \_\_\_\_\_  
 File type: DAM Locap: \_\_\_\_\_  
 Label/Path: \_\_\_\_\_  
 FCS Server: \_\_\_\_\_ Sets( \_\_\_\_\_ , \_\_\_\_\_ )  
 Access: [ ] (" " - read/write, Read, Write)  
 File share: [S] (Exclusive, Shared, No locks)  
 Record hold: [T] (Transaction, Update, Yes)  
 Journal: [N] (Yes/No) Tape log: [N] (Yes/No)  
 Startup: Open: [ ] (" "-defer, Yes/No) Close: [ ] (Yes/No)  
 Create if needed: [ ] Comstore: [ ] Max records: \_\_\_\_\_  
 Logical delete: [N] Delete byte( value: location: )  
 Record size( [ ] ) Record pool( [ ] ) Record format( [F] ) (Fix/Var)  
 MSG-WAIT - cancel, F1 - redisplay  
 20,49 25x80 Ready XP51 MSG OVR CAP NUM SCRL

At the bottom of the screen is a summary of function keys and their actions.

At any time you may enter MSG WAIT to cancel the function and return to the main menu screen.

Pressing F1 causes the screen to be re-displayed (Caution! any data entered on the screen will be lost).

Fill in the information for the file definition and press XMIT to add the definition to the TIP\$SYS file.

If the file definition is added successfully and no security record exists for the file (in TIP\$SEC) then a screen will be displayed prompting for the security information. Refer to [smsec](#) for details about security records.

### Max records

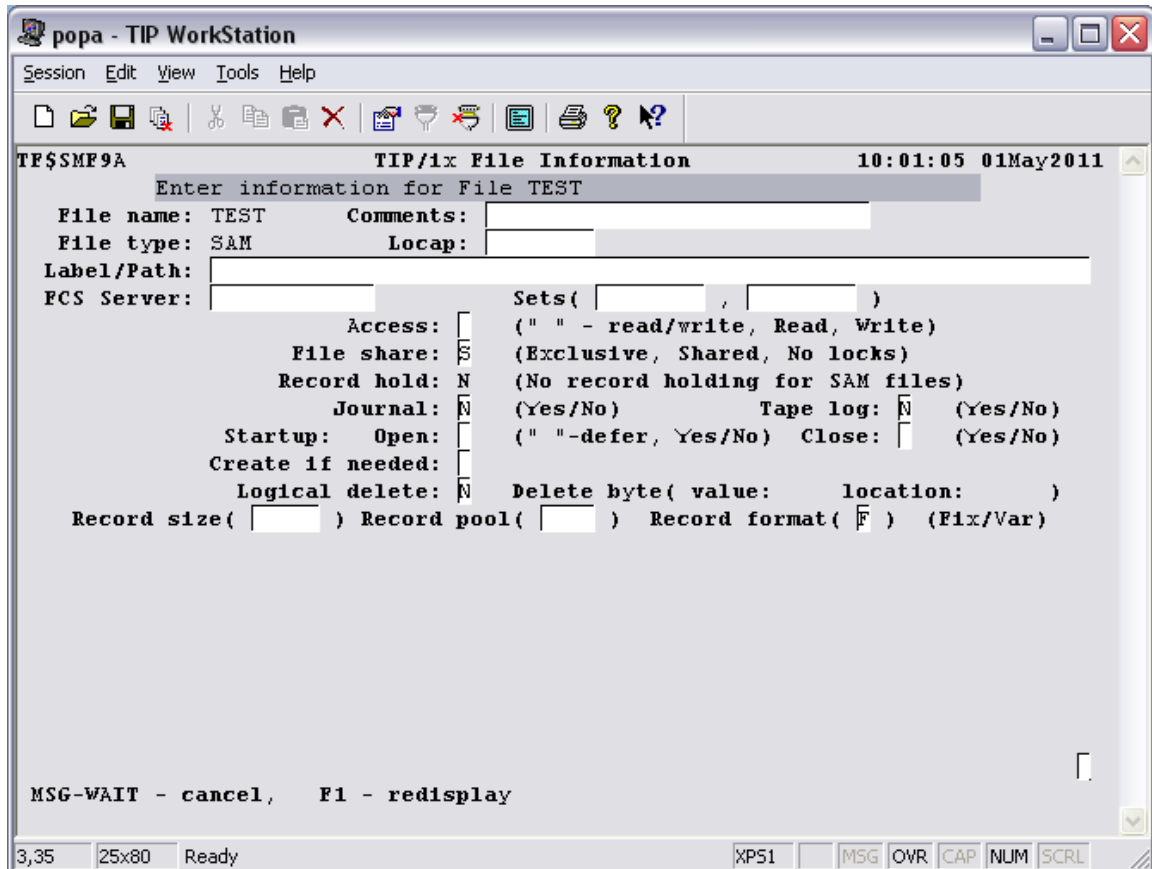
The maximum number of records which may exist in the file. This 10-digit number is an upper limit. The file could be smaller.

The other attributes for a DAM file definition are the same as those for an ISAM file except that there is no information relating to keys.

See the previous section "[Defining an Indexed File](#)" for a description of the fields on the screen.

## Defining a Sequential file

The following screen is displayed when a request is made to add, display, change, or delete an SAM (sequential) file definition.



The screenshot shows a window titled "popa - TIP WorkStation" with a menu bar (Session, Edit, View, Tools, Help) and a toolbar. The main area displays the following text:

```

TF$SMF9A TIP/ix File Information 10:01:05 01May2011
Enter information for File TEST
File name: TEST Comments: []
File type: SAM Locap: []
Label/Path: []
FCS Server: [] Sets([] , [])
 Access: [] (" " - read/write, Read, Write)
 File share: [S] (Exclusive, Shared, No locks)
 Record hold: [N] (No record holding for SAM files)
 Journal: [N] (Yes/No) Tape log: [N] (Yes/No)
Startup: Open: [] (" "-defer, Yes/No) Close: [] (Yes/No)
Create if needed: []
Logical delete: [N] Delete byte(value: location:)
Record size([]) Record pool([]) Record format([F]) (Fix/Var)

MSG-WAIT - cancel, F1 - redisplay

```

At the bottom of the window, there is a status bar showing "3,35 25x80 Ready" and several function key indicators: "XP51", "MSG", "OVR", "CAP", "NUM", "SCRL".

At the bottom of the screen is a summary of function keys and their actions.

At any time you may enter MSG WAIT to cancel the function and return to the main menu screen.

Pressing F1 causes the screen to be re-displayed (Caution! any data entered on the screen will be lost).

Fill in the information for the file definition and press XMIT to add the definition to the TIP\$SYS file.

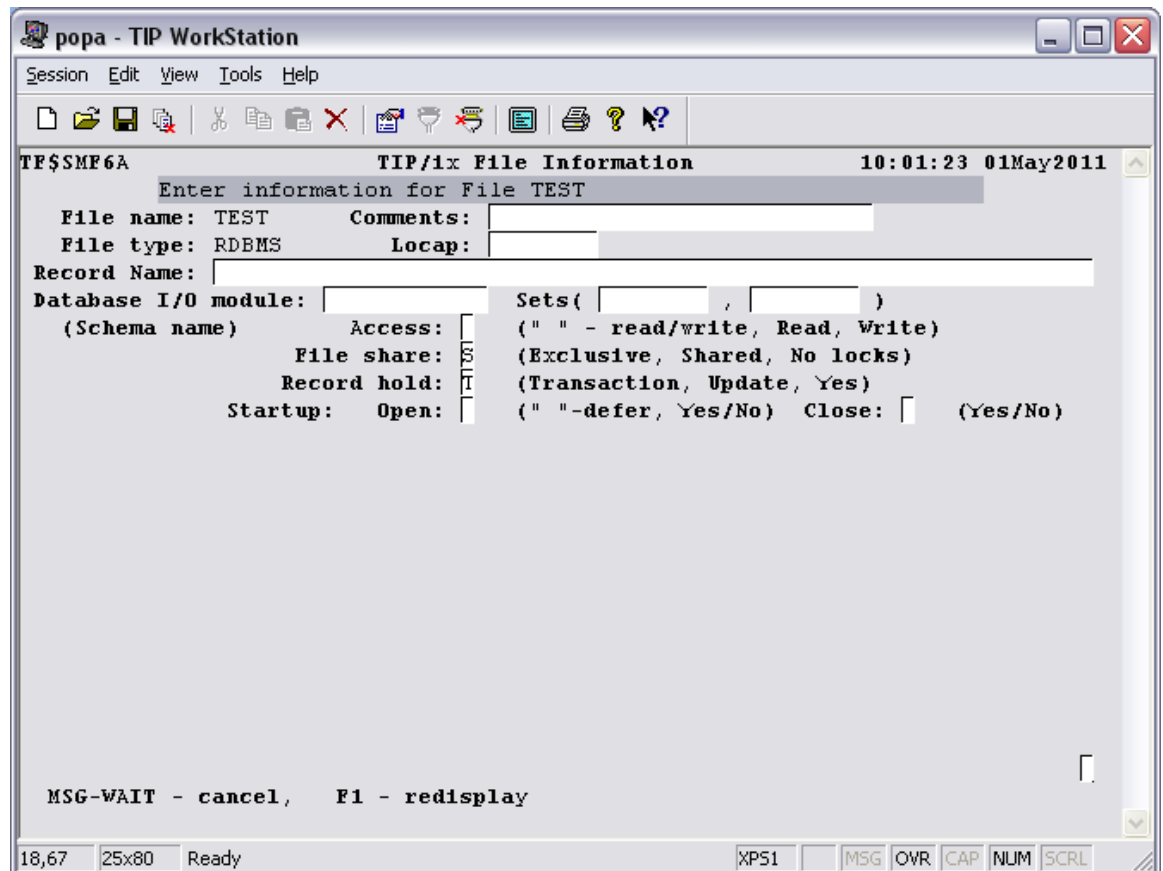
If the file definition is added successfully and no security record exists for the file (in TIP\$SEC) then a screen will be displayed prompting for the security information. Refer to [smsec](#) for details about security records.

The attributes for a SAM file definition are that same as those for an ISAM file except that there is no information relating to keys and there is no Record holding available to SAM files. Refer to a previous section "[Defining an Indexed File](#)" for a description of the fields on the screen.



## Defining a Database File

The following screen is displayed when a request is made to add, display, change, or delete an RDBMS (relational database) file definition.



At the bottom of the screen is a summary of function keys and their actions.

At any time you may enter MSG WAIT to cancel the function and return to the main menu screen.

Pressing F1 causes the screen to be re-displayed (Caution! any data entered on the screen will be lost).

Fill in the information for the file definition and press XMIT to add the definition to the TIP\$SYS file.

If the file definition is added successfully and no security record exists for the file (in TIP\$SEC) then a screen will be displayed prompting for the security information. Refer to [smsec](#) for details about security records.

There are only a few file definition attributes that apply to database files. Refer to the previous section "[Defining an Indexed File](#)" for a description of the common attributes.

The attributes that are unique to database files are:

**Record Name:**

The record name from the schema definition for the database

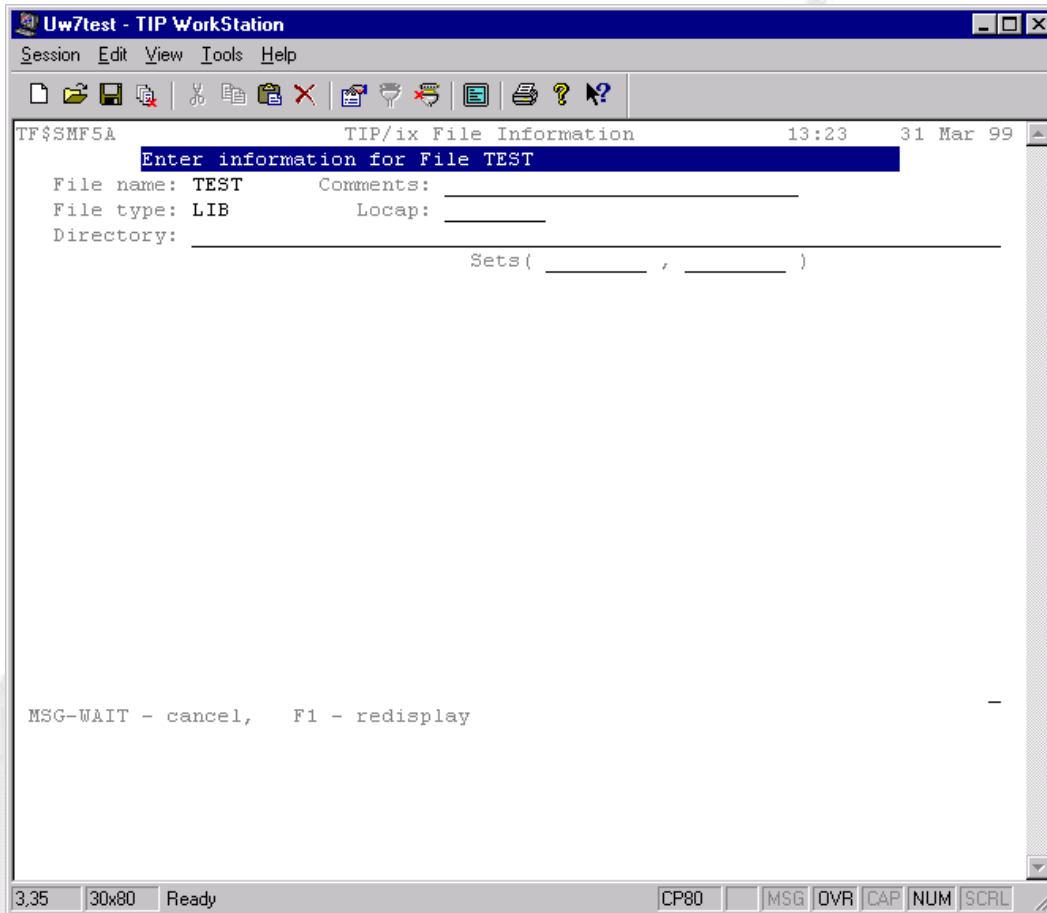
**Database I/O module:**

Normally this is the schema name in lower case followed by "io".

For example if schema is "ABC" then I/O module is "abcio".

**Defining a Library File**

The following screen is displayed when a request is made to add, display, change, or delete a Library file definition. Library files are implemented in TIP/ix as a UNIX directory. That is the library name is associated with a path which is a UNIX directory. Opening and manipulating a "library module" creates and manipulates a UNIX file in that directory.



At the bottom of the screen is a summary of function keys and their actions.

At any time you may enter MSG WAIT to cancel the function and return to the main menu screen.

Pressing F1 causes the screen to be re-displayed (Caution! any data entered on the screen will be lost).

Fill in the information for the file definition and press XMIT to add the definition to the TIP\$SYS file.

If the file definition is added successfully and no security record exists for the file (in TIP\$SEC) then a screen will be displayed prompting for the security information. Refer to [smsec](#) for details about security records.

There are only a few attributes that apply to library files. Refer to the previous section "[Defining an Indexed File](#)" for a description of the common attributes.

One difference with library files is that a UNIX directory is filled in instead of a UNIX file.

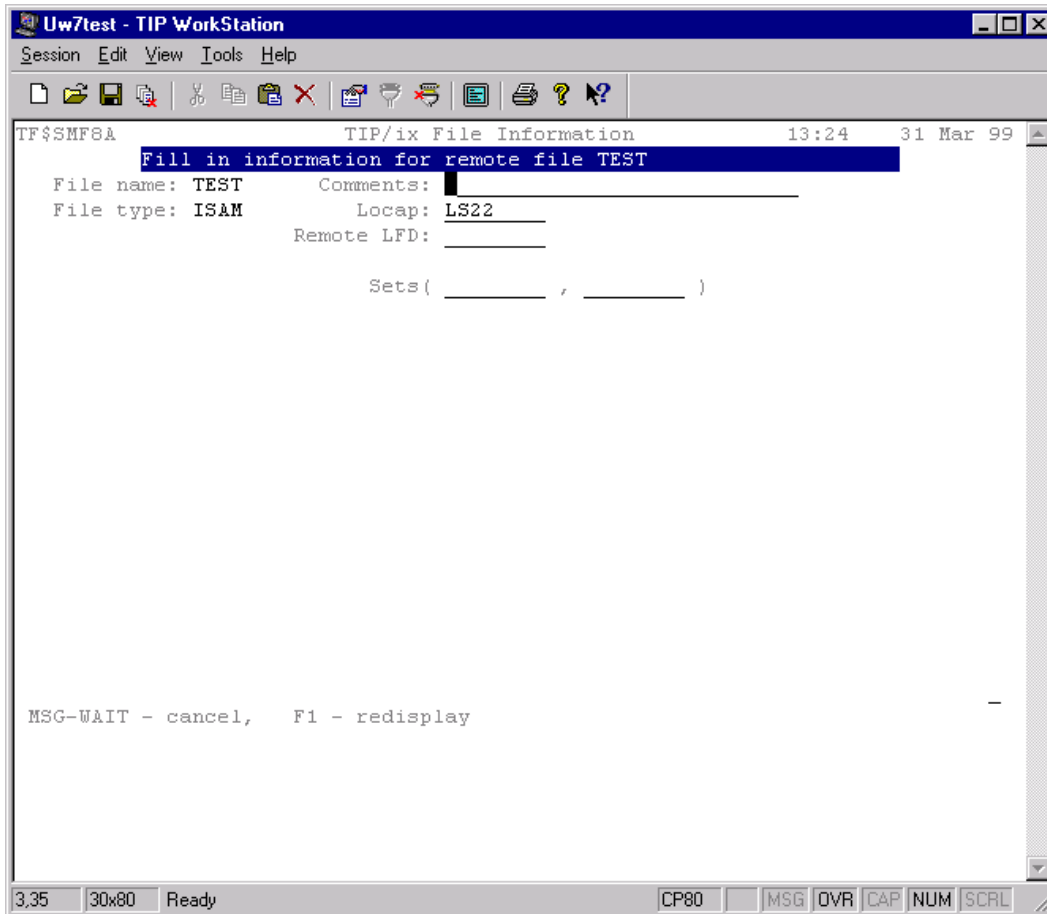
#### **Directory**

UNIX directory to use for this TIP/ix library. Modules referenced from this library will be read or written using this directory.

#### **Defining a Remote File**

A remote file is defined to be a file to be accessed from another TIP/ix or TIP/30 system. Each TIP system is identified by a LOCAP name. [Smlocap](#) is used to associate LOCAP names with TIP systems.

To define a remote file you 1st make a request to add or change a file record. Then on the file definition screen fill in the LOCAP name of the TIP system where the file is to reside. When a LOCAP name is filled in and it is not the LOCAP of the current TIP/ix system the following screen is displayed.



At the bottom of the screen is a summary of function keys and their actions.

At any time you may enter MSG WAIT to cancel the function and return to the main menu screen.

Pressing F1 causes the screen to be re-displayed (Caution! any data entered on the screen will be lost).

Fill in the information for the file definition and press XMIT to add the definition to the TIP\$SYS file.

If the file definition is added successfully and no security record exists for the file (in TIP\$SEC) then a screen will be displayed prompting for the security information. Refer to [smsec](#) for details about security records.

The only attribute that is unique to remote files is "Remote LFD".

#### Remote LFD:

The smfile entry on the remote LOCAP to be used to determine what UNIX file to access.

## List File Definitions

From the **smfile** menu enter “LI” for the function to be performed (List) and a name to use as the starting point for the list. This does not have to be a complete name. For example a name of “B” could be used to start listing file definitions with a name greater than (or equal to) “B”.

The list function displays a single line summary for each file definition showing the logical (**smfile**) file name, the file type, the file comments, and the UNIX file associated with the file definition.

The data displayed in the column titled “Unix File” on the list display varies with the file type. If the file is a remote file then the LOCAP is displayed instead of the UNIX file name. For print files (see [snsmprint](#)) instead of a UNIX file it will indicate the action (PIPE, FORK, or WRITE) associated with the print file definition and show either the print command or print directory.

When the UNIX file name is too long to fit on the list display it will be truncated so that the trailing portion (suffix) is displayed. If file name is an absolute path it will be truncated to `./.../<suffix>`. If the file name is a relative path it will be truncated to `.../<suffix>`.

The following screen is displayed in response to the list command.

```

Uw7test - TIP WorkStation
Session Edit View Tools Help
TF$SMF2A TIP/ix File Maintenance 13:25 31 Mar 99
Cmd File Type Description Unix File
- - - - -
DAMINV DAM Direct access Sample file daminv
INV ISAM TQL INV Example File inv
INVEN ISAM Sample Inventory Data file inven
ORD ISAM TQL ORD Example Data ord

PRNTR PRINT WRITE TO: /tmp
QAF1E1 SAM Created by ARMCAT /home1/tipix22/qa1111
QAF1E2 SAM Created by ARMCAT QAF1E2
QAF1E3 SAM Created by ARMCAT /tmp/qafile3

QAF1E4 SAM Created by ARMCAT QAF1E4
QAINVEN RDBMS QA INVEN TEST FILE gainven
QALIB LIB /usr/local/qatools/help
QATQD1E ISAM QATQD1E

<-- Enter 'D' to delete, 'C' to change, 'S' to show, 'G' to get security rec
Transmit to process command(s)

MSG-WAIT - return to menu, F1 - redisplay, F2 - next, F3 - previous

```

Press MSG WAIT to return to the main menu screen.

Press F2 to display the next screen of files.

Press F3 to display the next screen of files.

You can enter single character commands next to the files and when you press XMIT the commands will be processed sequentially (one at a time) from the top of the display. The commands that can be entered are:

- C** Change file definition
- D** Delete file definition
- S** Show file definition
- G** Get the security record(s) for this file definition.

If no commands are entered and you press XMIT next to a file on the list display then this is interpreted as a show (S) request for this file definition.

## Display a File Definition

**smfile** provides two ways to display a file definition.

From the **smfile** menu enter "S" for the function to be performed (show) and the name of the file definition to be displayed. ("TSPFILE" in our example).

Use the list function to list file definitions and type "S" next to the file definition(s) that you want to display.

The following screen is displayed in response to a request to display a file definition.

```

popa - TIP WorkStation
Session Edit View Tools Help
TIP$SMF3A TIP/ix File Information 10:02:29 01May2011

File name: TSPFILE Comments: Sample Data File
File type: ISAM Locap:
Label/Path: tspfile
FCS Server:

 Access: Sets(TIPY ,)
 File share: S (Exclusive, Shared, No locks)
 Record hold: T (Transaction, Update, Yes)
 Journal: Y (Yes/No/B4) Tape log: N (Yes/No/B4)
 Startup: Open: Y (" "-defer, Yes/No) Close: N (Yes/No)
 Create if needed: Y Comstore: N
 Logical delete: N Delete byte(value: location:)

Record size(335) Record pool(0) Record format(F) (Fix/Var)
Number of keys(3) Primary key(1)
Key Information: Location Length Dup/Chng Location Length Dup/Chng
1) (0 , 8 , N / N) 2) (9 , 25 , Y / Y)
3) (116 , 10 , Y / Y) 4) (, , /)
5) (, , /) 6) (, , /)
7) (, , /) 8) (, , /)
9) (, , /) 10) (, , /)

MSG-WAIT - exit, F2 - next, F3 - previous, F4 - update, F5 - security rec

```

Press MSG WAIT to return to the **smfile** screen where the show request was made.

Press F2 to display the next file definition.

Press F3 to display the previous file definition.

Press F4 to update the file definition. The screen will be redisplayed with the data fields padded with underscores.

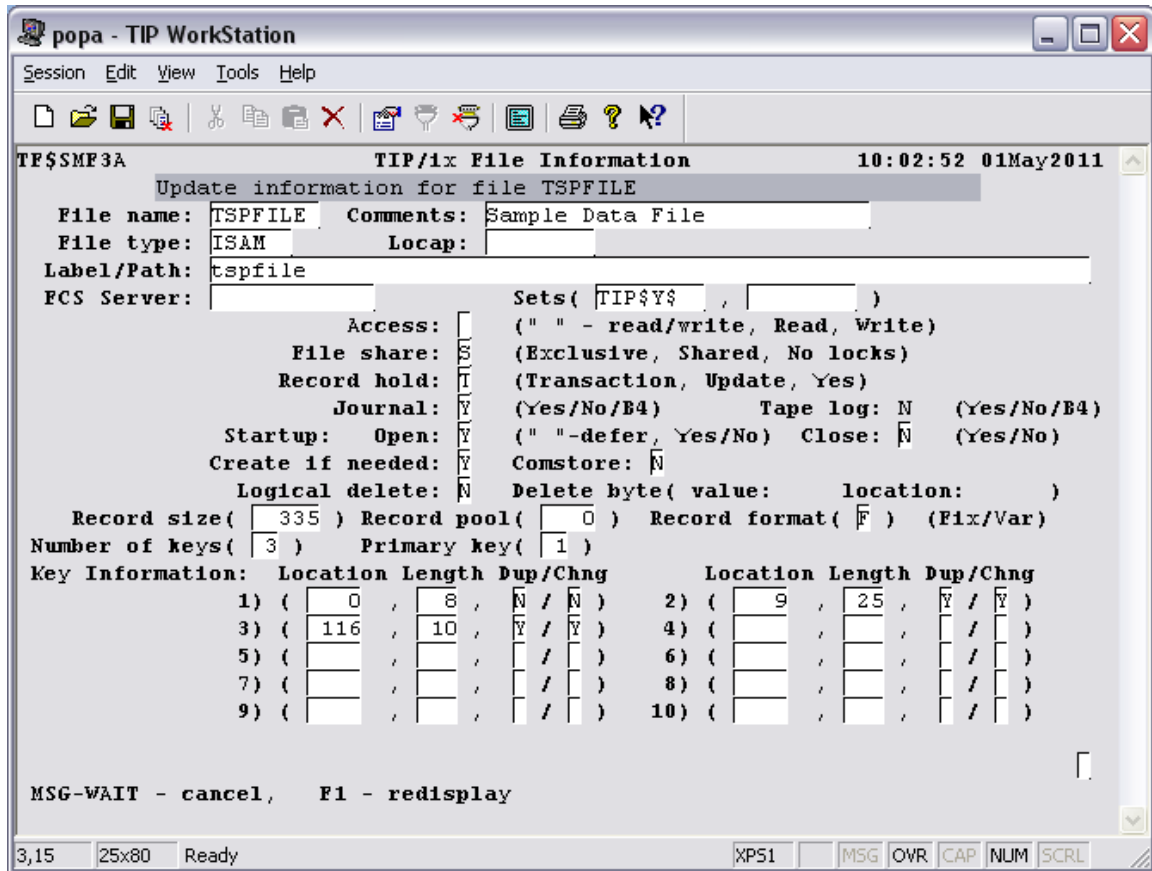
Press F5 to display the security information (from TIP\$SEC) for the file definition. This will show the file name(s) used to access the file via TIPFCS.

## Change a File Definition

**smfile** provides two ways to change (or update) a file definition.

1. From the **smfile** menu enter "CH" for the function to be performed (change) and the name of the file definition to be changed. ("TSPFILE" in our example).
2. Use the list function to list file definitions and type "C" next to the file definition(s) that you want to change.

The following screen is displayed in response to a request to change a file definition.



popa - TIP WorkStation

Session Edit View Tools Help

TF\$SMF3A TIP/ix File Information 10:02:52 01May2011

Update information for file TSPFILE

File name: TSPFILE Comments: Sample Data File  
 File type: ISAM Locap:  
 Label/Path: tspfile  
 FCS Server:

Access: ( " " - read/write, Read, Write)  
 File share: (Exclusive, Shared, No locks)  
 Record hold: (Transaction, Update, Yes)  
 Journal: (Yes/No/B4) Tape log: N (Yes/No/B4)  
 Startup: Open: ( " " -defer, Yes/No) Close: N (Yes/No)  
 Create if needed: Y Comstore: N  
 Logical delete: N Delete byte( value: location: )

Record size( 335 ) Record pool( 0 ) Record format( F ) (Fix/Var)  
 Number of keys( 3 ) Primary key( 1 )

| Key Information: |   |     |   | Location | Length | Dup/Chng | Location | Length | Dup/Chng |   |    |   |       |
|------------------|---|-----|---|----------|--------|----------|----------|--------|----------|---|----|---|-------|
| 1)               | ( | 0   | , | 8        | ,      | N / N    | 2)       | (      | 9        | , | 25 | , | Y / Y |
| 3)               | ( | 116 | , | 10       | ,      | Y / Y    | 4)       | (      |          | , |    | , | / /   |
| 5)               | ( |     | , |          | ,      | / /      | 6)       | (      |          | , |    | , | / /   |
| 7)               | ( |     | , |          | ,      | / /      | 8)       | (      |          | , |    | , | / /   |
| 9)               | ( |     | , |          | ,      | / /      | 10)      | (      |          | , |    | , | / /   |

MSG-WAIT - cancel, F1 - redisplay

3,15 25x80 Ready XPS1 MSG OVR CAP NUM SCRL

Press MSG WAIT to cancel the update and return to the **smfile** screen where the change request was made.

Fill in any desired changes and press XMIT to change the file definition. For a description of the fields on the file definition refer to the section on ["Adding a File Definition"](#).

If the file definition is changed successfully and no security record exists for the file (in TIP\$SEC) then a screen will be displayed prompting for the security information. Refer to [smsec](#) for details about security records.

If the file name is changed then the original file definition is left unchanged and a new file definition is added for the new name.

## Delete a File Definition

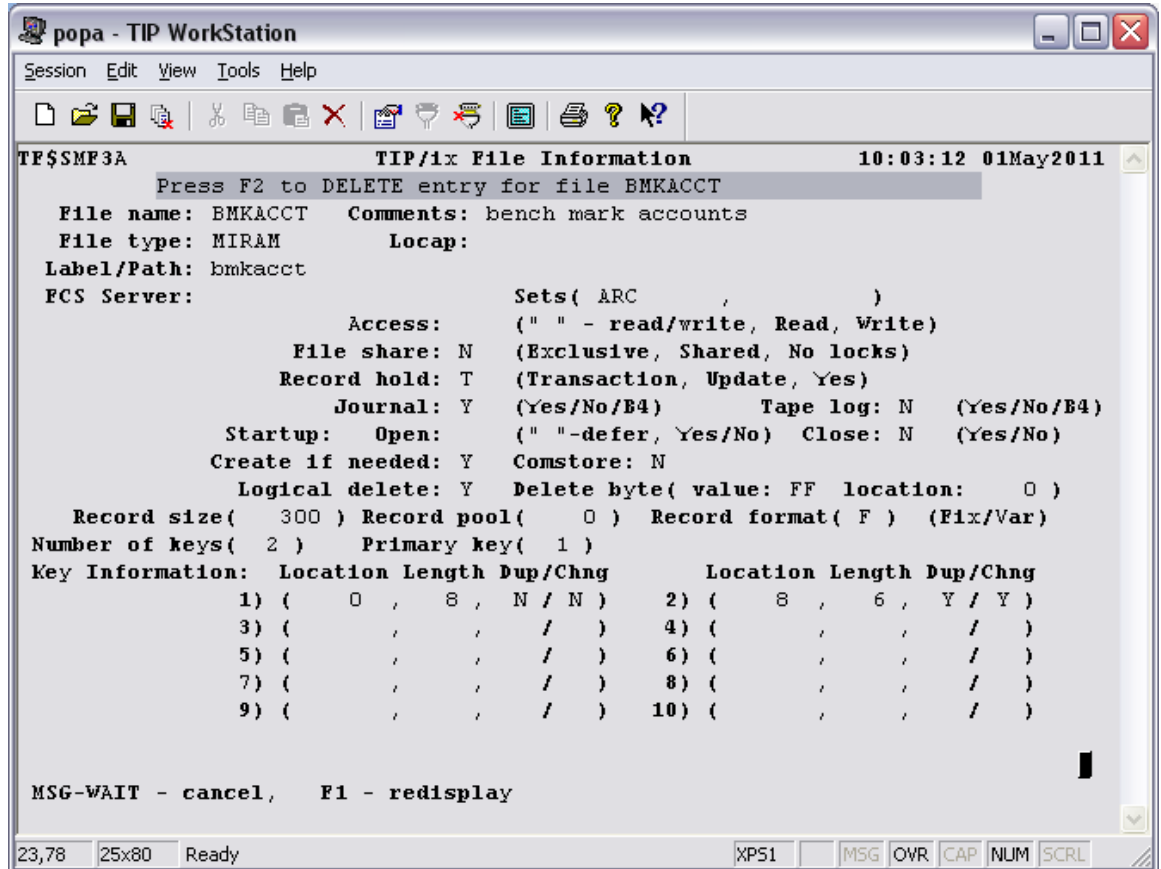
**smfile** provides two ways to delete a file definition.

1. From the **smfile** menu enter "DE" for the function to be performed (delete) and the name of the file definition to be deleted. ("BMKACCT" in our example).



- Use the list function to list file definitions and type "D" next to the file definition(s) that you want to delete.

The following screen is displayed in response to a request to delete a file definition.



Press MSG WAIT to cancel the delete request and return to the **smfile** screen where the delete request was made.

Press F2 to confirm that this is the file definition that you want to delete.

If you press F2 and the file definition is deleted then if there are any security records for this file definition you will be given the opportunity to delete them. If there is a single security record for the file it will be displayed and you are requested to press F2 to confirm that the security record is to be deleted. If there is more than one security record then they are displayed in a summary list (up to 12 will be shown) and a message indicating the number of security records that will be deleted is shown. Pressing F2 will delete all the security records for the file definition that was deleted.

## Get Security Records for File Definitions

**smfile** provides two ways to retrieve security records for file definitions.

1. Use the list function to list file definitions and type “G” next to the file definition(s) for which you want to display the corresponding security information.
2. Display a file definition (using either the show command or pressing XMIT next to a file on the list display) and then press F5.

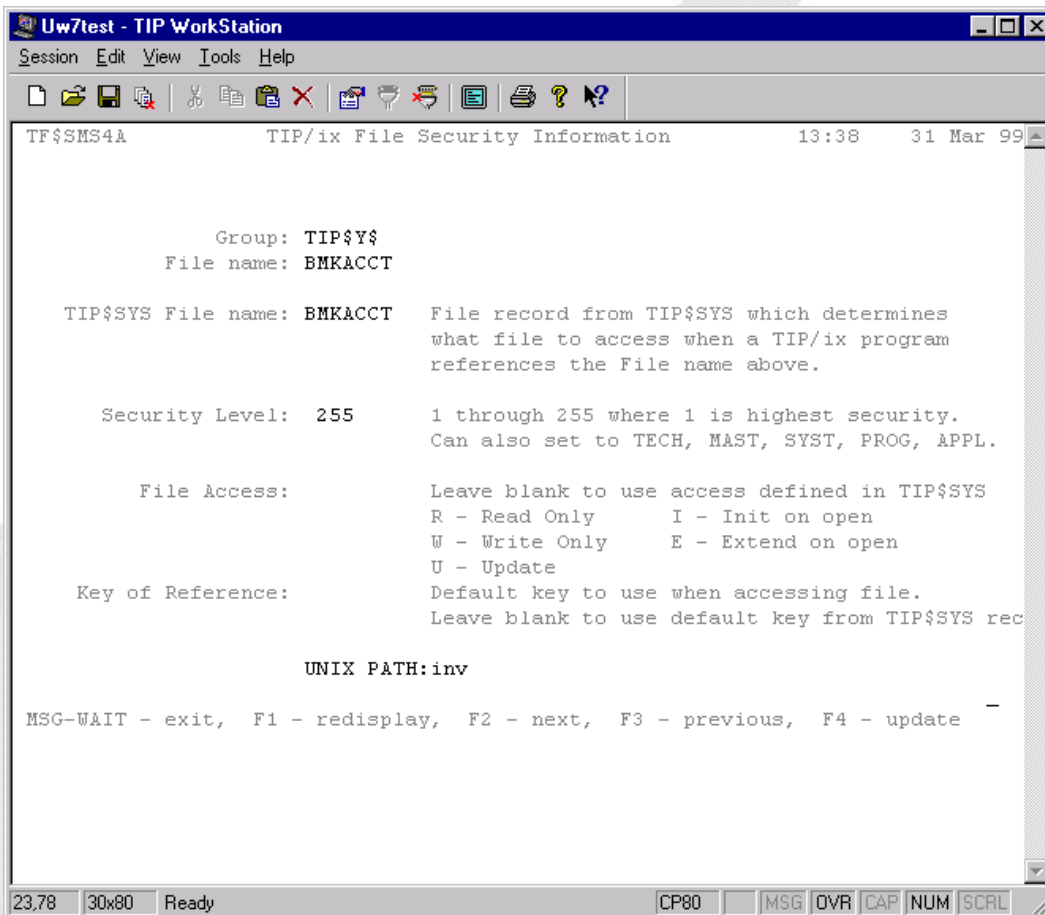
If there is only a single security record that references this file definition then that security record is displayed.

If there is more than one security record then a list of all security records referencing this file record will be displayed.

If no security record is defined then you will be asked if you wish to add one (via a Yes/No prompt).

For information about security records refer to the section on the [smsec](#) program

The following screen is displayed if there is a single security record referencing this file definition. (This example shows a request for the security record for the file definition for BMKACCT.)



```

Uw7test - TIP WorkStation
Session Edit View Tools Help
TF$SMS4A TIP/ix File Security Information 13:38 31 Mar 99

 Group: TIPY
 File name: BMKACCT

TIP$SYS File name: BMKACCT File record from TIP$SYS which determines
 what file to access when a TIP/ix program
 references the File name above.

 Security Level: 255 1 through 255 where 1 is highest security.
 Can also set to TECH, MAST, SYST, PROG, APPL.

 File Access: Leave blank to use access defined in TIP$SYS
 R - Read Only I - Init on open
 W - Write Only E - Extend on open
 U - Update

 Key of Reference: Default key to use when accessing file.
 Leave blank to use default key from TIP$SYS rec

 UNIX PATH:inv

MSG-WAIT - exit, F1 - redisplay, F2 - next, F3 - previous, F4 - update

```

Press MSG WAIT to return to the file definition screen where the request was made for the security information.

Press F2 to display the next security record in the TIP\$SEC file (navigating by the 3rd key which is the associated TIP\$SYS file name). The subsequent security records may or may not relate to the file definition for which security information was originally requested.

Press F3 to display the previous security record in the TIP\$SEC file (navigating by the 3rd key which is the associated TIP\$SYS file name). The subsequent security records may or may not relate to the file definition for which security information was originally requested.

Press F4 to update the security record. The screen will be redisplayed with the data fields padded with underscores.

The following screen is displayed in response to a request for security information if there are multiple security records referencing the file definition. (This example shows a request for the security record for the file definition for BMKACCT.)

```

uw7test.tws - TIP WorkStation
Session Edit View Tools Help
=====
TF$SMSBA TIP/ix Security Maintenance Utility 16:44 21 May 99
=====
End of List
Command Type TIP$SYS TIP$SEC Security TIP$SYS
 Name Name Group Level Unix Path
- FILE BMKACCT BMKACCT TEST 255 BMKACCT
- FILE BMKACCT BMKACCT TIPY 255 BMKACCT

<-- Enter 'D' to delete, 'C' to change, 'S' to show
 Transmit to process command(s)

MSG-WAIT - return to menu, F1 - redisplay, F2 - next, F3 - previous
6,5 24x80 Ready 2084 MSG OVR CAP NUM SCRL

```

Press MSG WAIT to return to the file definition screen where the request was made for the security information.

Press F2 to display the next screen of security records for the TIP\$SYS file name

Press F3 to display the previous screen of security records for the TIP\$SYS file name.

You can enter single character commands next to the security records and when you XMIT the commands will be processed sequentially (one at

a time) from the top of the display. The commands that can be entered are:

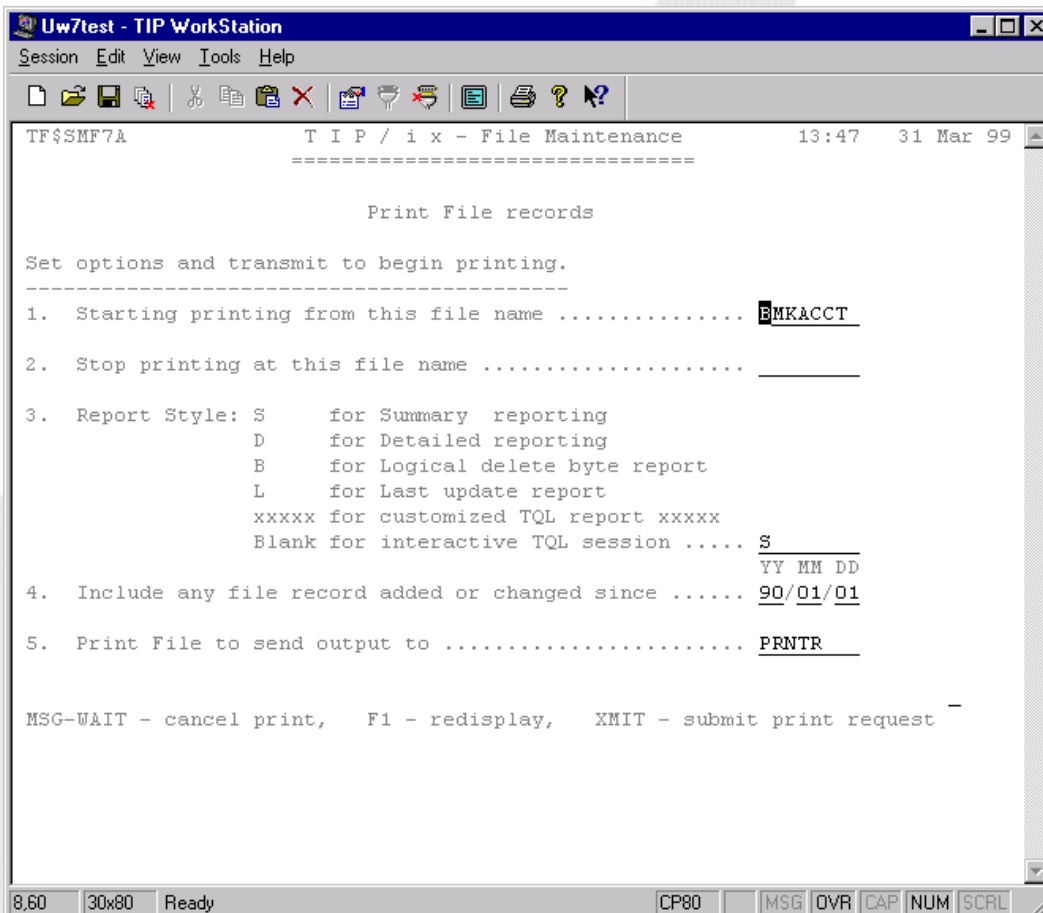
- C** Change security entry
- D** Delete security entry
- S** Show security entry

If no commands are entered and you press XMIT next to a security entry then this is interpreted as a show (S) request for that security entry.

## Print File Definitions

To print file definitions enter “PR” as the function to be performed (print) on the **smfile** menu display. A file name can also be supplied to be used as the point to start printing file definitions from. After pressing **XMIT**, you will be prompted to fill in the print options and report style.

When you select the Print function from **smfile** the following screen is displayed. Enter the desired print options and press transmit to print TIP/ix file definition information.



```

Uw7test - TIP WorkStation
Session Edit View Tools Help
TF$SMF7A T I P / i x - File Maintenance 13:47 31 Mar 99
=====
Print File records

Set options and transmit to begin printing.

1. Starting printing from this file name BMK&CCT
2. Stop printing at this file name _____
3. Report Style: S for Summary reporting
 D for Detailed reporting
 B for Logical delete byte report
 L for Last update report
 xxxxx for customized TQL report xxxxx
 Blank for interactive TQL session S
 YY MM DD
4. Include any file record added or changed since 90/01/01
5. Print File to send output to PRNTR

MSG-WAIT - cancel print, F1 - redisplay, XMIT - submit print request -
8,60 30x80 Ready CP80 MSG OVR CAP NUM SCRL

```

The following is a description of the fields in the preceding screen.

**Start at file name****Stop at file name**

Range of file definitions to include in report. For example if set to "K" and "R" respectively then all file names in the range K to R will be included in the report.

**Report style**

Fill in one of the valid report styles or leave blank.

- S** Summary report that prints a single line of information about each file.
- D** Detailed report that includes all information in the file definition as well as all security records for the file definitions.
- B** Report of files using logical record delete. Report includes value and location of the delete byte.
- L** Report who last updated each file definition and when each file definition was last updated.

**<report name>**

Any valid report name in the TQL program TIPSYSF. This allows you to modify the TQL program and select your own reports. Fill in LASTUPDT for a TQL report showing the user id, date, and time of the last update for each file definition.

**<blank>**

Leave this entry blank to initiate an interactive TQL session with the program TIPSYSF

**Change Date**

Only report file definitions that have changed since this date.

**Print File**

Print file to route report to. This must be a valid TIPPRINT destination. Print files are defined by [smprint](#) and of course aliases can be set up using [smsec](#). AUX0 can be used to direct report output to the terminal.

The print function is implemented by the supplied TQL program TIPSYSF. If this program is not compiled and available then the print function will not work.

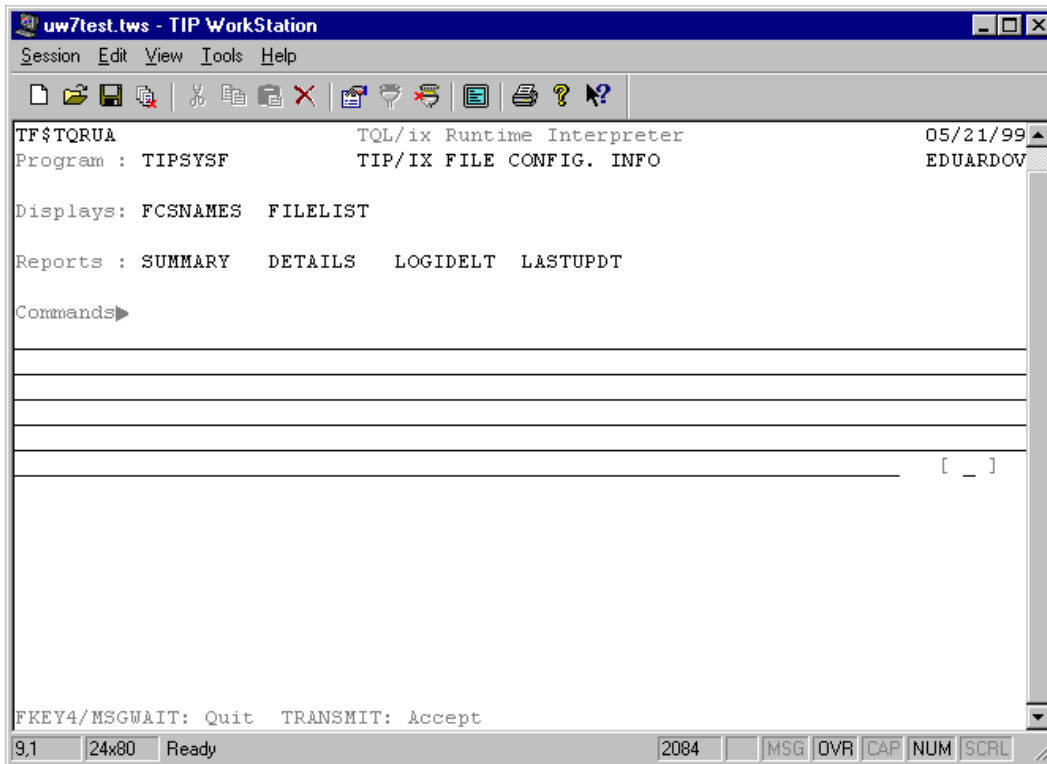
**Browse File Definitions with TQL**

The **TIPSYSF** TQL program enables you to interrogate file definitions with adhoc TQL queries. This TQL program can be run either:

- Directly from the TIP/ix command prompt by entering "TQL TIPSYSF"

- From the **smfile** menu enter "BR" for the function to be performed (browse).

The following screen is displayed when you run **TIPSYSF**.



At this point you can make any valid TQL request including:

- Request a predefined display.
- Request a predefined report.
- Make an ad hoc request to list (or print) fields from the file definitions.

See the **TIP/ix TQL Reference** manual for details on interacting with TQL.

The predefined displays include:

### **FCSNAMES**

Display all TIPFCS names that are associated with a file definition (via the smsec utility). The display consists of a few fields from the file definition (file name, comments, file type, LOCAP, and path) followed by a list of all security records that refer to the file definition.

### **FILELIST**

Display 16 file definitions at a time with a single line per file definition. This summary display shows the following attributes from the file definition (file name, file type, access, file share, record hold, journal, record size, number of keys, open, close, logical delete).

The predefined reports include:

### **SUMMARY**

Each file definition is summarized into a single line on the report. The report consists of the following fields from the file definition (file name, file type, comment, label/path).

### **DETAILS**

For each file definition the report includes all fields from the file definition and any security records referencing the file definition.

### **LOGIDELT**

A report listing only file definitions with the "logical delete" attribute set to "Y". Each file definition included in the report is summarized onto a single line consisting of (file name, file type, logical delete, delete byte value, delete byte location, record size, location of key1, and length of key1).

### **LASTUPDT**

A report showing when and who last updated file definitions. Each file definition included in the report is summarized onto a single line consisting of: file name, file type, comment, date and time the file definition was last updated, and user id that last updated the file definition (usually with smfile).

In addition to using the predefined displays and reports it is possible to perform ad hoc request and queries based on field settings in file definitions. The following is a list of field names and the corresponding file definition attribute:

| File Definition Attribute | TQL Field Names |
|---------------------------|-----------------|
| File Name                 | PF-NAME         |
| Comments                  | PF-CMT          |
| File Type                 | FILE-TYPE       |
| LOCAP                     | PF-LOCAP        |
| Label/Path                | PF-LBL          |
| FCS Server                | PF-TIPFCS       |
| Sets (1st one)            | PF-SET (1)      |
| Sets (2nd one)            | PF-SET (2)      |
| Access                    | PF-ACS          |
| File Share                | PF-SHARE        |
| Record Hold               | PF-RECHOLD      |

|                      |                |
|----------------------|----------------|
| Journal              | PF-RECJRN      |
| Open                 | PF-OPEN        |
| Close                | PF-CLOSE       |
| Create if needed     | PF-CREATE-FILE |
| Comstore             | PF-COMSTORE    |
| Logical Delete       | WK-LOGIDEL     |
| Delete Byte Value    | PF-DELBYTE     |
| Delete Byte Location | PF-DELOFFSET * |
| Record Size          | PF-RECSZ       |
| Record Pool          | PF-POOL        |
| Record Format        | PF-RCFM        |
| Number of Keys       | PF-NKEYS       |
| Primary Key          | PF-PKEY        |
| Key Location         | PF-KLOC (1-10) |
| Key Length           | PF-KLEN (1-10) |
| Key Dup              | PF-DUPS (1-10) |
| Key Chng             | PF-CHNG (1-10) |

\* You must subtract 1 from PF-DELOFFSET so that the value matches the zero relative value entered via SMFILE.

### Examples:

List files with more than 1 key.

```
FILELIST IF PF-NKEYS > 1
```

List files defined to use logical record delete.

```
SUMMARY IF WK-LOGIDEL = "Y"
```

or just use the predefined LOGIDELT report

```
LOGIDELT
```

List files defined to be of type "DAM".

```
SUMMARY IF FILE-TYPE = "DAM"
```

List files that have record size > 2000 and then sort the result in descending order by record size.

```
LIST (PF-NAME FILE-TYPE PF-RECSZ)
IF PF-RECSZ > 2000 SORT BY PF-RECSZ:D
```

Print file details report on auxiliary printer for file definitions with names in the range PUR to PZZ.



## DETAILS FROM "PUR" TO "PZZ" ON AUX1

---

## smgrpset - Group Set Definition

The **smgrpset** transaction is used to define group sets. A group set associates a name with a set of application groups. Each group set record can contain up to 32 groups and group set records can be chained from one to another. Therefore, a single group set name could effectively refer to many application groups (more than 32).

A group set name may be referred to by the [smuser](#) transaction to assign application groups that a user can access. [smuser](#) allows you to grant a user access to 2 elective (application) groups. To grant access to more than 2 groups it is necessary to use group sets. [smuser](#) allows group set names to be entered in the fields "Group set" and "Logon set".

TIP/ix permits a given user access to at most 16 application groups at any point in time. This can be thought of as the user's active groups. When a user starts a tipix session the active groups will consist of the user's 2 elective groups and the first *n* groups in the user's *Logon set* to a maximum of 16.

The active groups can be altered by using the **groups** transaction. Application programs may also alter the active groups by calling the **groups** transaction (via TIPSUB) and filling in the CDA with the desired group names. The **groups** transaction can only give access to the user's 2 elective groups and groups in the user's *Group set* and *Logon set*.

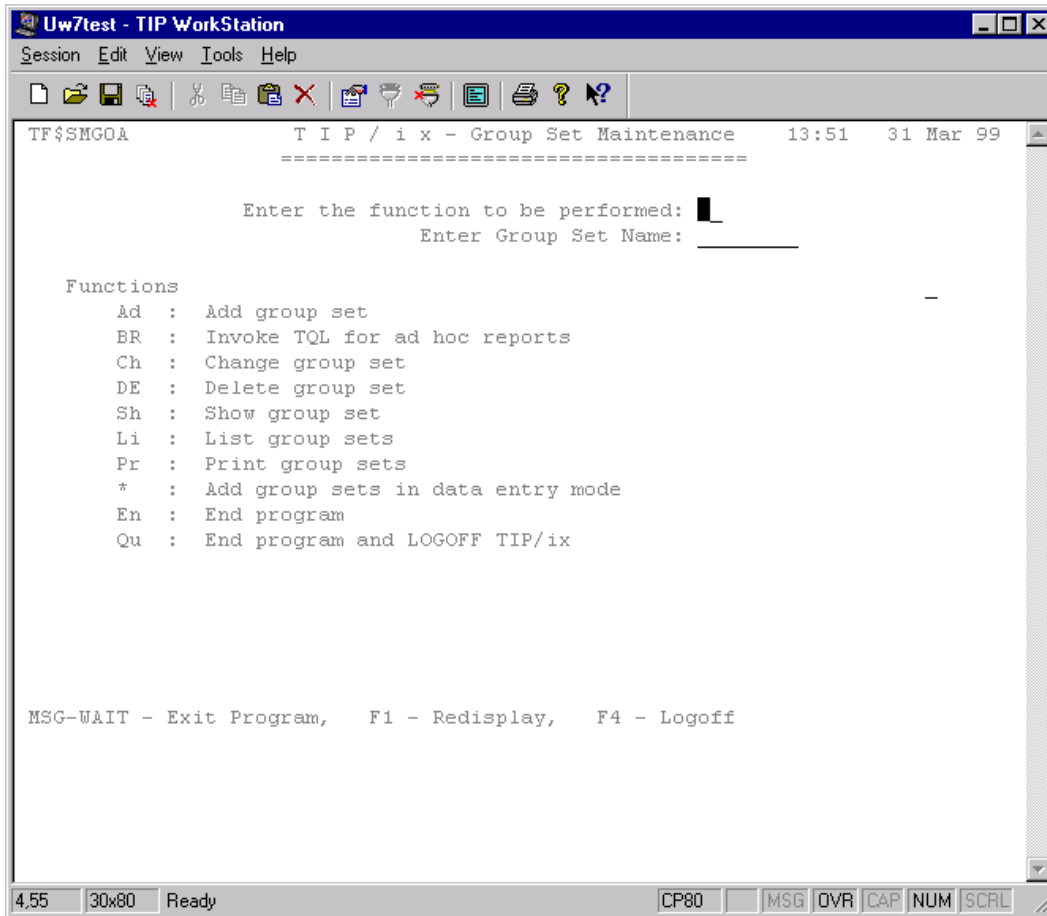
Therefore, a user can have *active* access to at most 16 application groups and *potential* access to any number of application groups (by chaining group set definitions).

### Syntax:

```
smgrpset [function] [groupset]
```

If a valid function is supplied then **smgrpset** will perform that function and then terminate when that function has been completed. Most functions require a group set name to act upon.

If no function is supplied then **smgrpset** displays the following group set maintenance menu screen.



```

Uw7test - TIP WorkStation
Session Edit View Tools Help
TF$SMGOA T I P / i x - Group Set Maintenance 13:51 31 Mar 99
=====
Enter the function to be performed: L
Enter Group Set Name: _____

Functions
Ad : Add group set
BR : Invoke TQL for ad hoc reports
Ch : Change group set
DE : Delete group set
Sh : Show group set
Li : List group sets
Pr : Print group sets
* : Add group sets in data entry mode
En : End program
Qu : End program and LOGOFF TIP/ix

MSG-WAIT - Exit Program, F1 - Redisplay, F4 - Logoff

4,55 30x80 Ready CP80 MSG OVR CAP NUM SCRL

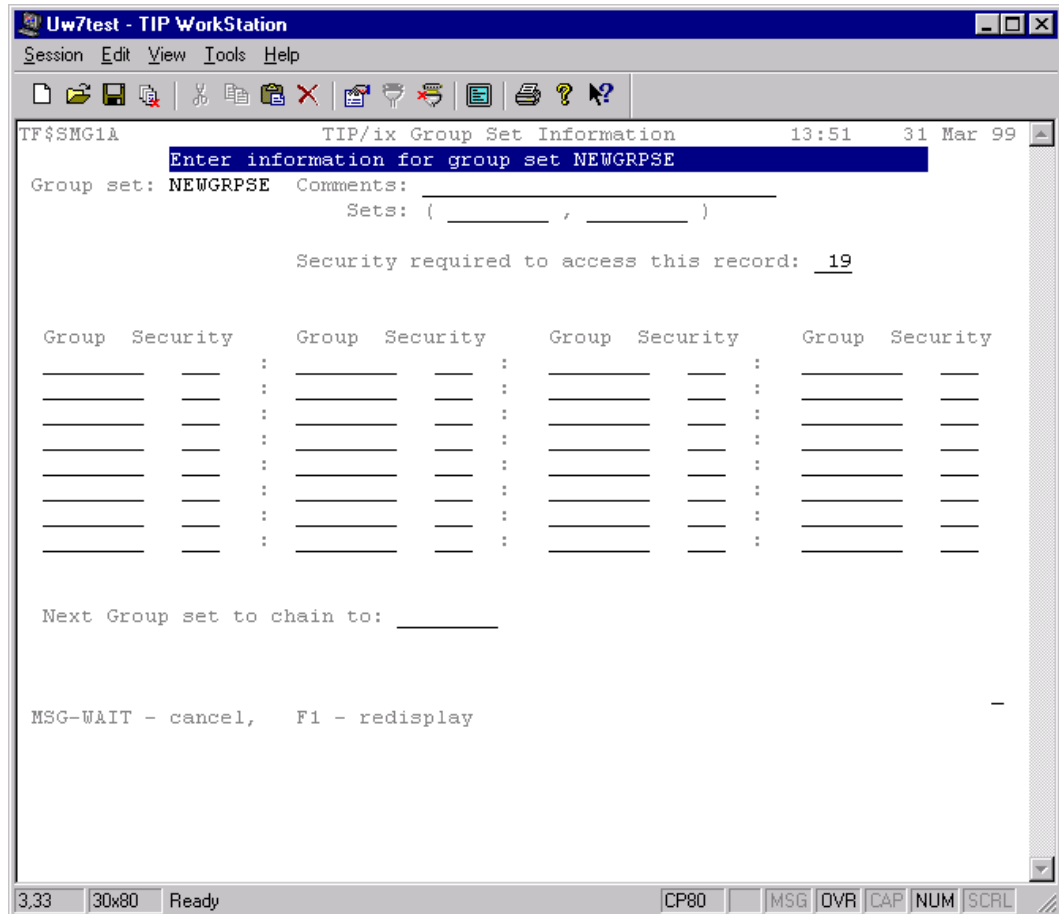
```

From this menu you may perform all the traditional maintenance functions on the group set definitions. You may add new group set definitions; change, delete, show, or print existing group set definitions and list a summary of all group set definitions.

In the preceding screen, the function to be performed can be abbreviated to the portion shown in upper case. For example the function *add* can be entered as “A”.

## Add a Group Set Definition

To define a new group set definition, enter “A” for the function to be performed (add) and the name of the group set (“NEWGRPST” in our example). After pressing **XMIT**, the following screen appears to allow you to enter the group set definition information.



At the bottom of the screen is a summary of function keys and their actions.

At any time you may enter MSG WAIT to cancel the function and return to the main menu screen.

Pressing F1 causes the screen to be re-displayed (Caution! any data entered on the screen will be lost).

Fill in the information for the group set definition and press XMIT to add the definition to the TIP\$SYS file.

The following is a description of each field in the preceding screen.

**Group set:**

The name of this group set definition. This name is used by the smuser transaction when assigning group access to user ids.

**Comments:**

Optional information describing the group set.

**Sets**

This entry is used to identify a set of records in the TIP\$SYS file that are in some way related. For example you could mark all TIP\$SYS records used in a payroll

application by filling in the set name with "PAYROLL". The intention is that these records can be migrated as a set from one TIP/ix system to another using the "tippack" transaction.

**Security required to access this record:**

This is a number between 1 and 255 which represents the level of security a user must have to access this group set definition via the **smgrpset** program. The highest level is level 1 and the lowest is 255. If the user's security value is numerically greater than this number then access will be denied.

This field only applies to maintenance of this group set record. This value has no affect on users granted access to this group set via the [smuser](#) program.

**Group** A group set can consist of up to 32 application groups. Duplicate names are removed from the list. Any user assigned access to this group set (via [smuser](#)) has potential access to the application groups in the group set definition. Note that each user can have active access to at most 16 application groups at any one time.

A user's active groups and security level(s) determine what files, programs, and queues a user can access. See [smsec](#) for a description of how to create security records for files, programs, and queues.

**Security**

This is a number between 1 and 255 which represents the level of security to be assigned to users for access to items in the corresponding application group (when access to the application group is granted via this group set record). The highest level is level 1 and the lowest is 255.

If a security value is assigned for a group (in a group set) then this takes precedence over the user's assigned security (for access in the associated group). A user's security level can be either increased or reduced by this technique.

For example, a group and security entry of "PAYROLL 19" implies that a user assigned to this group set via [smuser](#) (logon set or group set) will be able to access items (files, programs, queues) in the group PAYROLL with a security value in the range 19 to 255 (regardless of the security value assigned to the user id).

Group set records can be linked into "chains" to increase the number of groups available to users assigned to this group set.

Note: The maximum number of active groups for a given user is 16 but that there is no maximum number of potential groups.

## List Group Set Definitions

From the **smgrpset** menu enter “LI” for the function to be performed (List) and a name to use as the starting point for the list. This does not have to be a complete name. For example a name of “F” could be used to start listing group set definitions with a name greater than (or equal to) “F”.

The list function displays a single line summary for each group set definition showing the group set name, the group set description, the security required to access and/or the group set definition with **smgrpset**, the next group set in a chain of group sets, and the number of groups in the group set record.

The following screen is displayed in response to the list command.

```

Uw7test - TIP WorkStation
Session Edit View Tools Help
TF$SMG2A TIP/ix Group Set Maintenance 13:52 31 Mar 99
End of List
Cmd Group Set Description Group Set Security Chained to Group Set Groups in Group Set
_ NEWGRPSE New Group 19 1

<-- Enter 'D' to delete, 'C' to change, 'S' to show
Transmit to process command(s)

MSG-WAIT - return to menu, F1 - redisplay, F2 - next, F3 - previous
5,3 30x80 Ready CP80 MSG OVR CAP NUM SCRL

```

Press MSG WAIT to return to the main menu screen.

Press F2 to display the next screen-full of group sets.

Press F3 to display the previous screen-full of group sets.

You can enter single character commands next to the group sets and when you XMIT the commands will be processed sequentially (one at a time) from the top of the display. The commands that can be entered are:

- C** Change group set definition
- D** Delete group set definition
- S** Show group set definition

If no commands are entered and you press XMIT next to a group set on the list display then this is interpreted as a show (S) request for this group set definition.

## Display a Group Set Definition

**smgrpset** provides two ways to display a group set definition.

1. From the **smgrpset** menu enter "S" for the function to be performed (show) and the name of the group set definition to be displayed. ("NEWGRPSE" in our example).
2. Use the list function to list group set definitions and type "S" next to the group set definition(s) that you want to display.

The following screen is displayed in response to a request to display a group set definition.

```

Uw7test - TIP WorkStation
Session Edit View Tools Help
[Icons]
TF$SMG1A TIP/ix Group Set Information 13:55 31 Mar 99
Group set: NEWGRPSE Comments: New Group
Sets: (,)
Security required to access this record: 19
Last Update: 99/03/31 08:50 SCOTTC
Group Security Group Security Group Security Group Security
TECH : : : :
: : : :
: : : :
: : : :
: : : :
: : : :
: : : :
Next Group set to chain to:
MSG-WAIT - exit, F1 - redisplay, F2 - next, F3 - previous, F4 - update
23,78 30x80 Ready CP80 MSG OVR CAP NUM SCRL

```

Press MSG WAIT to return to the **smgrpset** screen where the show request was made.

Press F2 to display the next group set definition.

Press F3 to display the previous group set definition.

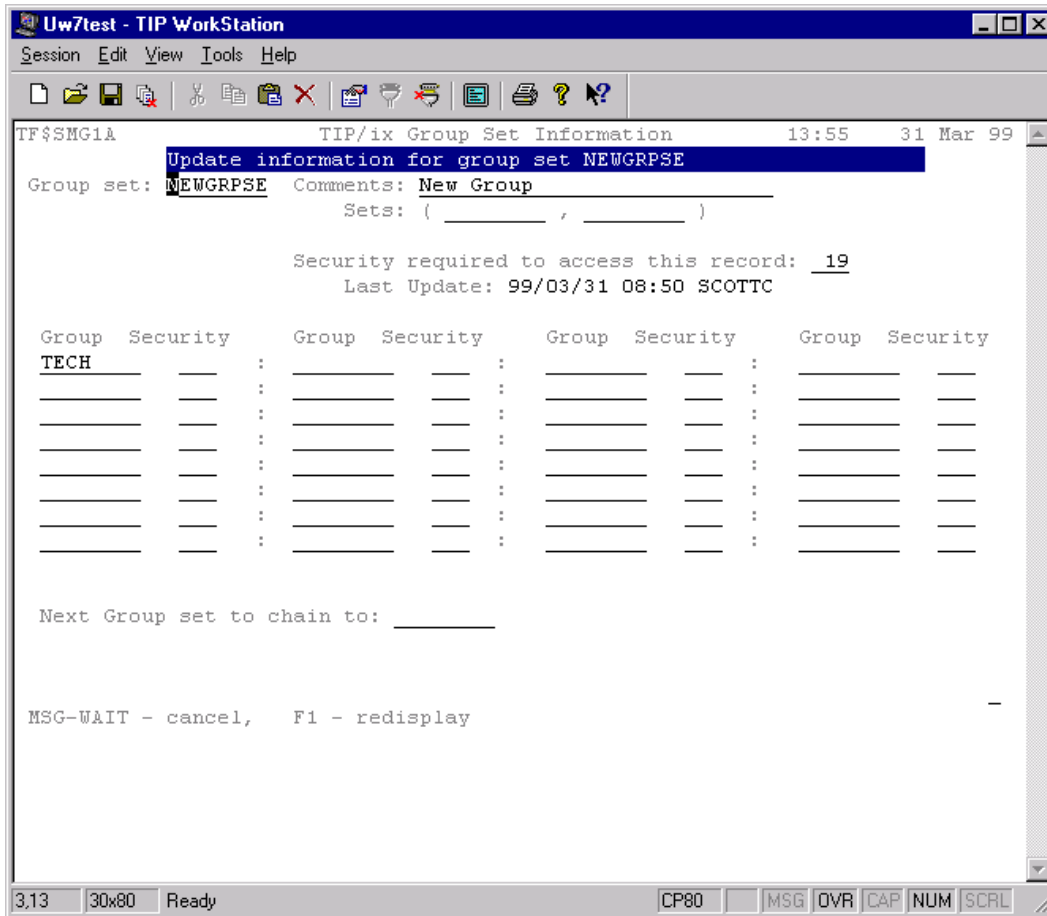
Press F4 to update the group set definition. The screen will be redisplayed with the data fields padded with underscores.

## Change a Group Set Definition

**smgrpset** provides two ways to change (or update) a group set definition.

1. From the **smgrpset** menu enter "CH" for the function to be performed (change) and the name of the group set definition to be changed. ("NEWGRPSE" in our example).
2. Use the list function to list group set definitions and type "C" next to the group set definition(s) that you want to change.

The following screen is displayed in response to a request to change a group set definition.



Press MSG WAIT to cancel the update and return to the **smgrpset** screen where the change request was made.

Fill in any desired changes and press XMIT to change the group set definition. For a description of the fields on the group set definition refer to the section on [Adding a Group Set Definition](#).

If the group set name is changed then the original group set definition is left unchanged and a new group set definition is added for the new name.

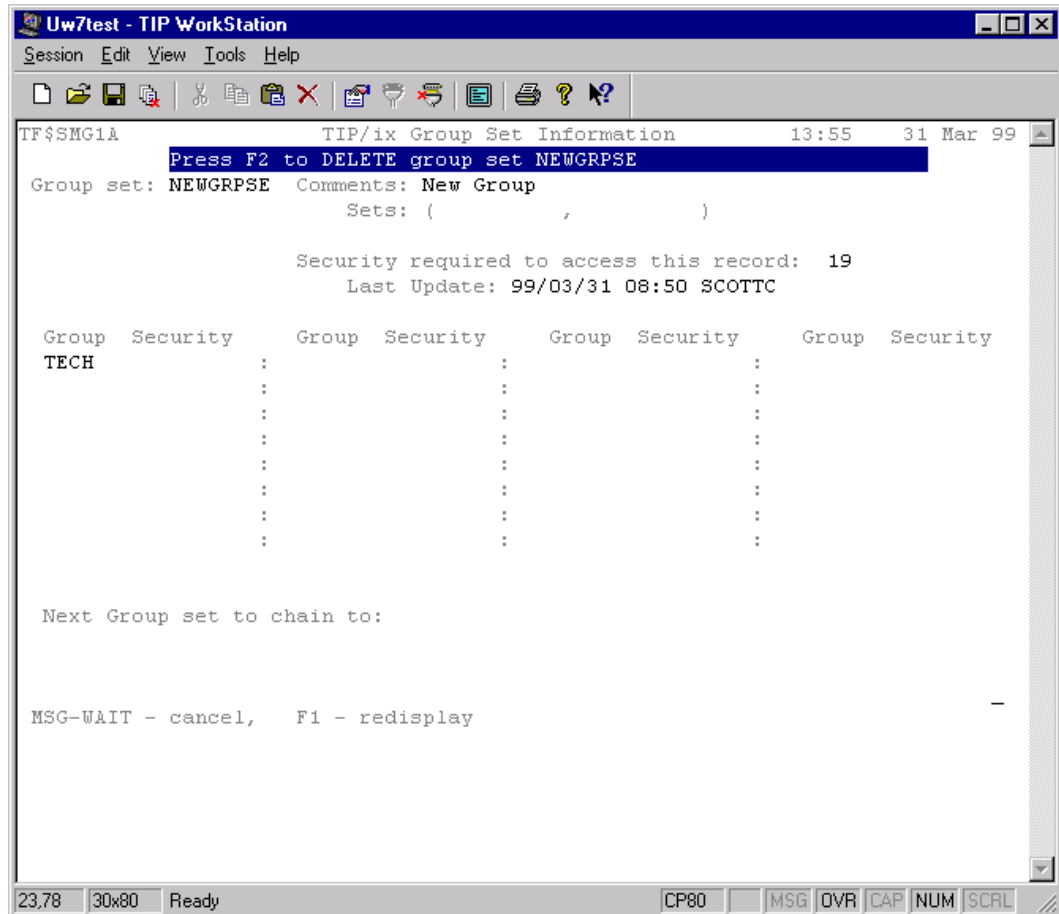
## Delete a Group Set Definition

**smgrpset** provides two ways to delete a group set definition.

1. From the **smgrpset** menu enter "DE" for the function to be performed (delete) and the name of the group set definition to be deleted. ("NEWGRPSE" in our example).
2. Use the list function to list group set definitions and type "D" next to the group set definition(s) that you want to delete.

The following screen is displayed in response to a request to delete a group set definition.





Press MSG WAIT to cancel the delete request and return to the **smgrpset** screen where the delete request was made.

Press F2 to confirm that this is the group set definition that you want to delete.

## Print Group set Definitions

To print group set definitions enter “PR” as the function to be performed (print) on the **smgrpset** menu display. A group set name can also be supplied to be used as the point to start printing group set definitions from. After pressing **XMIT**, you will be prompted to fill in the print options and report style.

When you select the Print function from **smgrpset** the following screen is displayed. Enter the desired print options and press transmit to print TIP/ix group set definition information.

```

Uw7test - TIP WorkStation
Session Edit View Tools Help
TF$SMG7A T I P / i x - Print Group Set records 13:56 31 Mar 99
=====
Set options and transmit to begin printing.

Enter number of desired report style 1
 1 - summary (1 line per group set)
 2 - detailed reporting (complete group set information)
 3 - cross reference report listing userid definitions
 that refer to each group set
 4 - date and time of last update to group set record

Starting printing from this group set..... NEWGRPSE
Stop printing at this group set YY MM DD
Include any group set record added or changed since 90/01/01
Print File to send output to PRNTR

MSG-WAIT - cancel print, F1 - redisplay, XMIT - submit print request

6,60 30x80 Ready CP80 MSG OVR CAP NUM SCRL

```

The following is a description of the fields in the preceding screen.

### Report style

- 1 Summary report that prints a single line of information about each group set.
- 2 Detailed report that includes all information in the group set definition.
- 3 Cross reference report that shows for each group set in the report all user ids that are assigned access (via Logon Set or Group Set fields of smuser) to the group set.
- 4 Report date, time, and user id performing last update of each group set record.

### <blank>

Leave this entry blank to initiate an interactive TQL session with the TQL program TIPSYSG

### Start printing from this Group Set

### Stop printing at this Group Set

Range of group set records to include in report. For example if set to "K" and "R" respectively then all group set names in the range K to R will be included in the report.

### Change Date

Only report group set definitions that have been added or changed (via smgrpset) since this date.

### Print File

Print file to route report to. This must be a valid TIPPRINT destination. Print files are defined by smprint and of course aliases can be set up using smsec. AUX0 can be used to direct report output to the terminal.

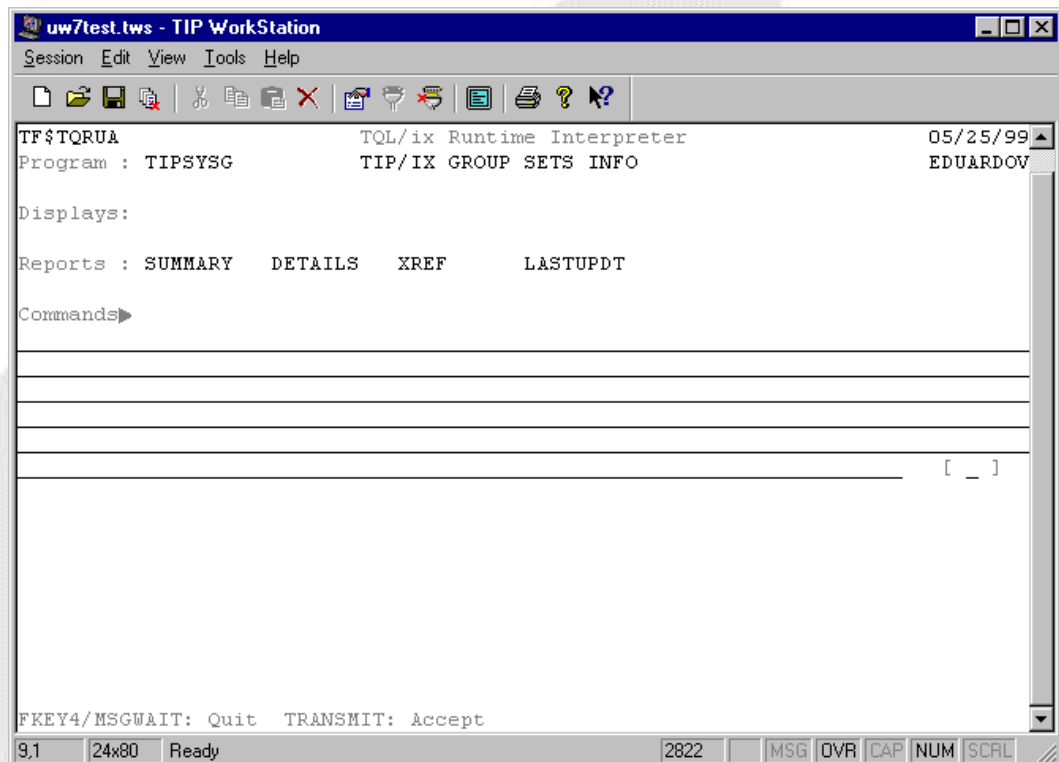
**Note:** The print function is implemented by the supplied TQL program TIPSYSG. If this program is not compiled and available then the print function will not work.

## Browse Group Set Definitions with TQL

The **TIPSYSG** TQL program enables you to interrogate group set definitions with ad hoc TQL queries. This TQL program can be run either:

- Directly from the TIP/ix command prompt by entering "TQL TIPSYSG"
- From the **smgrpset** menu enter "BR" for the function to be performed (browse).

The following screen is displayed when you run **TIPSYSG**.



At this point you can make any valid TQL request including:

- Request a predefined report.
- Make an ad hoc request to list (or print) fields from the group definitions.

See the **TIP/ix TQL Reference** for details on interacting with TQL.

The predefined reports include:

#### **SUMMARY**

Each group set definition is summarized into a single line on the report.

#### **DETAILS**

For each group set definition, show all fields from the group set definition. This can be used to determine what groups are used to comprise the individual group sets.

**XREF** For each group set list the users that have access to the group set. Users can access a group set if the user definition (smuser) defines the group set as the user's "Group set" or "Logon set" or if the group set is chained to the user's "Group set" or "Logon set" (smgrpset).

#### **LASTUPDT**

A report showing who last updated group set definitions (and when).

In addition to using the predefined reports it is possible to perform ad hoc queries based on field settings in group set definitions. The following is a list of group set definition attributes and the corresponding field names:

| Group Set Definition Attribute          | TQL Field Names    |
|-----------------------------------------|--------------------|
| Group Set Name                          | GRP-NAME           |
| Comments                                | GRP-CMT            |
| Sets (1st one)                          | GRP-SET (1)        |
| Sets (2nd one)                          | GRP-SET (2)        |
| Security required to access this record | GRP-SECURITY       |
| Group                                   | GRP-GROUP (1 - 32) |
| Security                                | GRP-SECUR (1 - 32) |
| Next Group set to chain to              | GRP-CHAIN          |

#### **Examples:**

See which user definitions make reference to which group set definitions (either directly through "Group set" or "Logon set" or indirectly through group set chaining).

#### **XREF**

List group set definitions that can only be viewed or updated by users with a security level less than 10.

**SUMMARY IF GRP-SECURITY < 10**

List group set definitions that contain at least 4 groups.

**SUMMARY IF GRP-GROUP (4) > " "**

Print group set details report on auxiliary printer for group definitions with names in the range A to M.

**DETAILS FROM "A" TO "M" ON AUX1**

---

## smlocap - Define A LOCAP

The **smlocap** transaction is used to define LOCAPs. A LOCAP is an 8 character name used to identify TIP systems when multiple TIP systems exist in a network. Distributed transaction processing facilities such as TIPPEER, TIPQUEUE, remote file access, and remote program execution direct service requests to a LOCAP name. TIP uses the LOCAP definitions created with **smlocap** to determine what TIP system to forward these requests to. The LOCAP definitions are kept in a file known to TIP/ix as TIP\$SYS.

If the LOCAP refers to a TIP/ix system then the system is identified by the combination of host name and IP Port Number.

If the LOCAP refers to a TIP/30 system then the system is identified by the network address of the Channel Gate device, gateway name from the Channel Gate configuration, and the computer name identifying the system/80 host (if there are multiple system/80 hosts attached to the Channel gate device). As this definition suggests it is necessary to have a Channel Gate device to establish a connection between TIP/30 and TIP/ix.

Information on how to use TIPPEER and TIPQUEUE facilities is found in the *TIP/ix Programming Reference*. The utility program [smqueue](#) is used to define queues for use with the TIPQUEUE program interface.

I/O requests for a file can be routed to another TIP system by using [smfile](#) to associate a LOCAP name with the file.

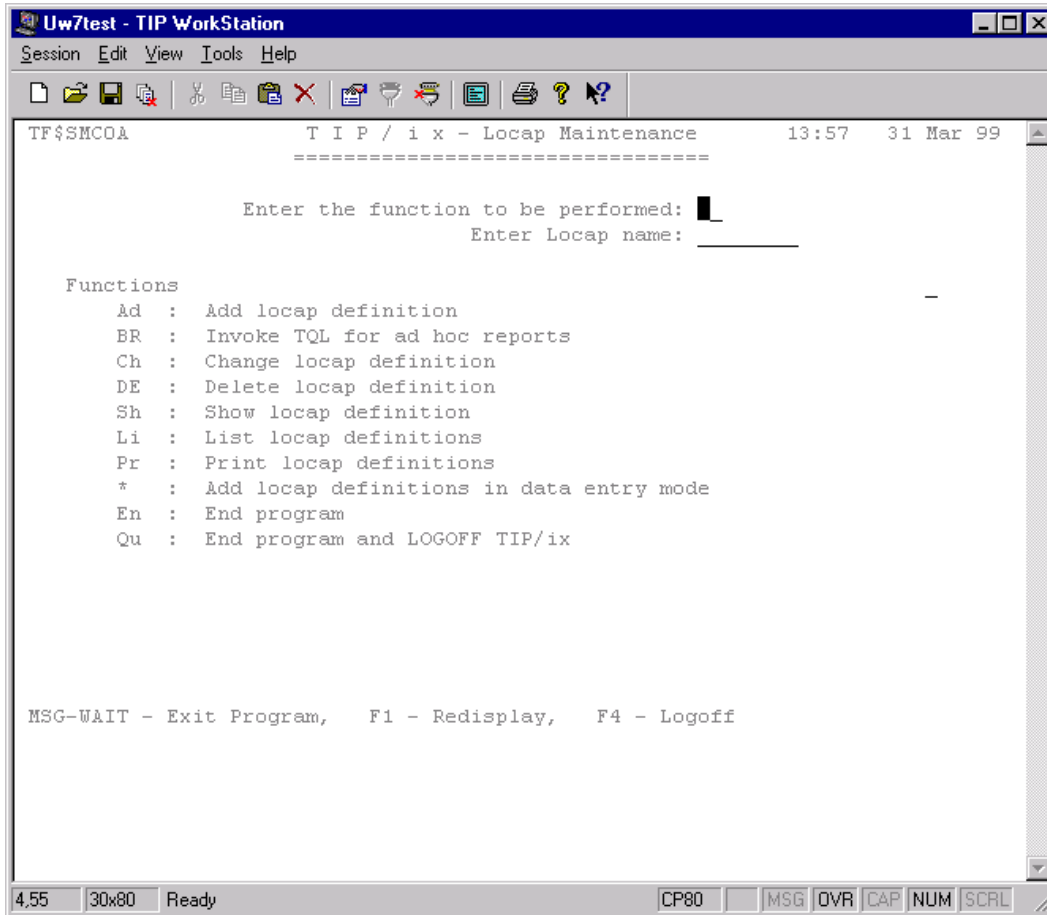
Programs can be set up to execute on another TIP system by filling in a LOCAP name when creating a program definition with [smprog](#).

### Syntax:

**smlocap [function] [locapname]**

If a valid function is supplied then **smlocap** will perform that function and then terminate when that function has been completed. Most functions require a LOCAP name to act upon.

If no function is supplied then **smlocap** displays the following program maintenance menu screen.



```

Uw7test - TIP WorkStation
Session Edit View Tools Help
TF$SMCOA T I P / i x - Locap Maintenance 13:57 31 Mar 99
=====
Enter the function to be performed: A
Enter Locap name: _____

Functions
Ad : Add locap definition
BR : Invoke TQL for ad hoc reports
Ch : Change locap definition
DE : Delete locap definition
Sh : Show locap definition
Li : List locap definitions
Pr : Print locap definitions
* : Add locap definitions in data entry mode
En : End program
Qu : End program and LOGOFF TIP/ix

MSG-WAIT - Exit Program, F1 - Redisplay, F4 - Logoff

4,55 30x80 Ready CP80 MSG OVR CAP NUM SCRL

```

From this menu you may perform all the traditional maintenance functions on the LOCAP definitions. You may add new LOCAP definitions; change, delete, show or print current LOCAP definitions and list a summary of all current LOCAP definitions.

In the preceding screen, the function to be performed can be abbreviated to the portion shown in upper case. For example the function *add* can be entered as “A”.

## Add a LOCAP Definition

To define a new LOCAP definition, enter “A” for the function to be performed (Add) and the name of the LOCAP (“NEWLOCAP” in our example). After pressing **XMIT**, you will be prompted to select the type of

LOCAP to add from a list of valid LOCAP types. The LOCAP types available are:

**TIP/IX**

The LOCAP identifies a TIP/IX system.

**TIP/30**

The LOCAP identifies a TIP/30 system.

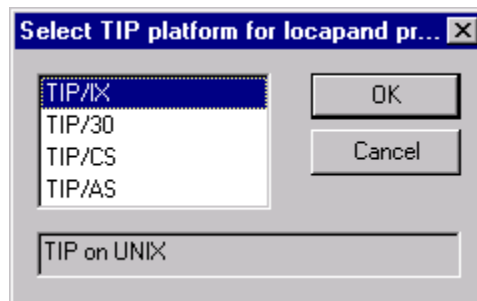
**TIP/AS**

The LOCAP identifies a TIP/AS system. TIP/AS (Application Server) is part of our TIP Studio product.

**TIP/CS**

The LOCAP identifies a TIP/CS system. TIP/CS (Connection Server) is part of our TIP Studio product.

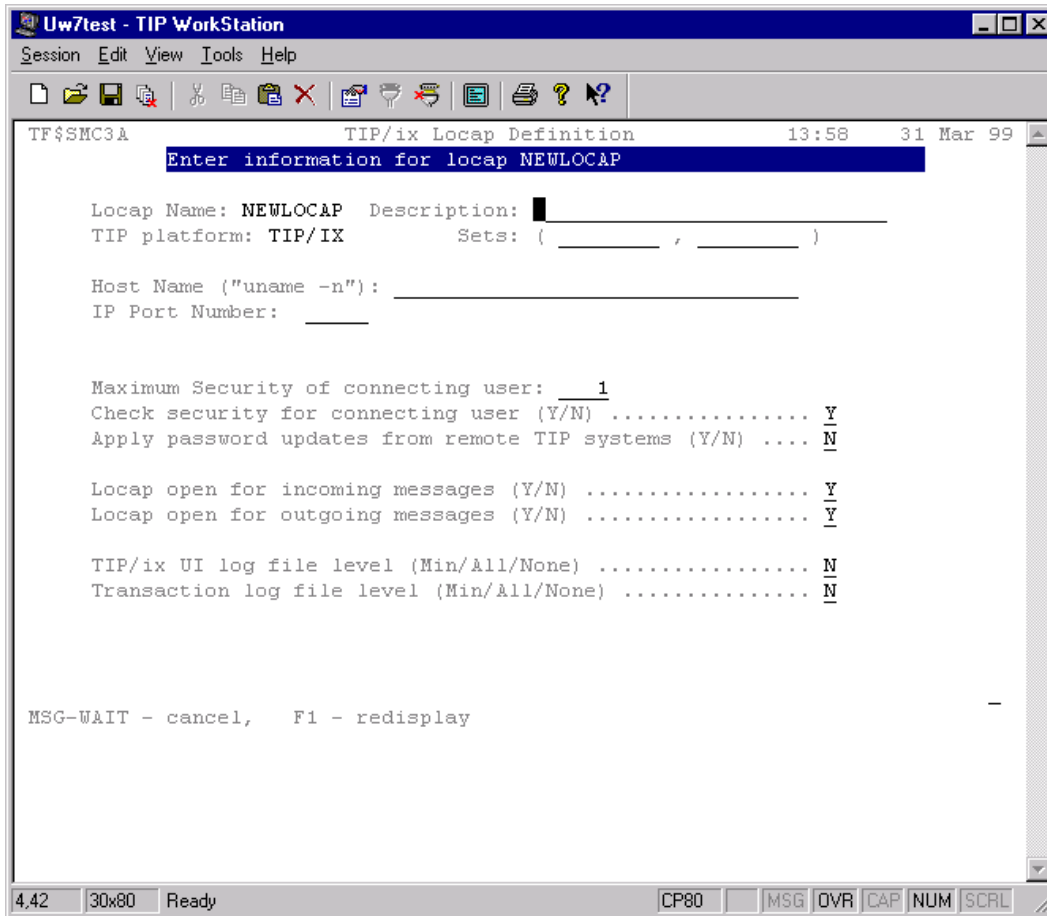
Move the cursor to the desired LOCAP type and press either of ENTER or XMIT.

**Selecting a LOCAP type:**

After you have selected the LOCAP type a screen will be displayed to allow you to enter the LOCAP definition information applicable to the LOCAP type you have selected.

**Defining a TIP/ix LOCAP**

The following screen is displayed when a request is made to add, display, change, or delete a TIP/ix LOCAP definition.



```

Uw7test - TIP WorkStation
Session Edit View Tools Help
TF$SMC3A TIP/ix Locap Definition 13:58 31 Mar 99
Enter information for locap NEWLOCAP

Locap Name: NEWLOCAP Description: []
TIP platform: TIP/IX Sets: ([] , [])

Host Name ("uname -n") : []
IP Port Number: []

Maximum Security of connecting user: 1
Check security for connecting user (Y/N) Y
Apply password updates from remote TIP systems (Y/N) N

Locap open for incoming messages (Y/N) Y
Locap open for outgoing messages (Y/N) Y

TIP/ix UI log file level (Min/All/None) N
Transaction log file level (Min/All/None) N

MSG-WAIT - cancel, F1 - redisplay

4.42 30x80 Ready CP80 MSG OVR CAP NUM SCRL

```

At the bottom of the screen is a summary of function keys and their actions.

At any time you may enter MSG WAIT to cancel the function and return to the main menu screen.

Pressing F1 causes the screen to be re-displayed (Caution! any data entered on the screen will be lost).

Fill in the information for the LOCAP definition and press XMIT to add the definition to the TIP\$SYS file.

The following is a description of each field in the preceding screen.

#### **LOCAP Name**

The name of the LOCAP. This is used to identify TIP systems in a network. Must be unique in the network.

#### **Description**

Optional information describing the TIP system (LOCAP).

#### **TIP platform**

Must be one of the valid LOCAP types.



**TIP/IX**

TIP system on UNIX

**TIP/30**

TIP system on OS/3

**TIP/AS**

TIP system on a Application Server, which is part of our TIP Studio product.

**TIP/CS**

TIP system on a Connection Server, which is part of our TIP Studio product.

**Sets** This entry is used to identify a set of records in the TIP\$SYS file that are in some way related. For example you could mark all TIP\$SYS records used in a payroll application by filling in the set name with "PAYROLL". The intention is that these records can be migrated as a set from one TIP/ix system to another using the "tippack" transaction.

**Host Name**

This is the node name of the computer where the TIP system to be associated with this LOCAP is found. This should match the string displayed by the UNIX command "uname -n" when this command is run on the UNIX system where the TIP system associated with this LOCAP is installed.

**IP Port Number**

Fill in a number between 5001 and 65535. All definitions in the network for a given LOCAP name must use the same value for the IP port number. Multiple TIP systems (LOCAPs) can be defined on a single host (computer) by assigning each LOCAP a unique IP Port Number.

**Maximum Security of connecting user**

All users connecting from the LOCAP named in this definition will have their security reduced to this level if their own security level is higher (numerically less than) this value.

This field can be a number between 1 and 255 or one of the following mnemonics which equate to the values shown.

**TECH** 1  
**MAST** 9  
**SYST** 19  
**PROG** 29  
**APPL** 32

**Check security for connecting user**

- Y** Only allow user ids that have been defined on the current TIP/ix system (via smuser) are permitted to connect from the LOCAP named in this definition. The security level and groups will be set according to the local definition of connecting user ids (subject to the Maximum Security restriction).
- N** No restriction on users connecting from the LOCAP named in this definition. The security level of connecting users is not altered unless it exceeds the Maximum Security setting in this LOCAP definition. The active groups for a connecting user are set to the first two active groups at the time the connection was made.

**Apply password updates**

Respond "Y" to allow password updates to be propagated from the LOCAP named in this definition. The local TIP system will accept password updates for a user id from a remote TIP system.

This feature is not currently implemented.

**Keep LOCAP open for incoming msgs**

- Y** Allow incoming connections from the LOCAP named in this definition to the current TIP system.
- N** Reject any attempts to connect to the current LOCAP from the LOCAP named in this definition.

**Keep LOCAP open for outgoing msgs**

- Y** Attempt to satisfy requests to connect to the LOCAP named in this definition from the current TIP system.
- N** Reject any attempts to connect to the LOCAP named in this definition.

**TIP/ix UI log file level**

- M** Record minimal logging information for the user interface process of any incoming distributed session from the LOCAP named in this definition.
- A** Record all possible logging information for the user interface process of any incoming distributed session from the LOCAP named in this definition.

For inbound DTP sessions, the log file is created in the **\$TIPROOT/tmpwrk/tiprsub.termname** (where termname is the name of the terminal being used).

Previously, these log files were created in **\$TIPROOT/log/tiprsub.pid**.

The log file name is the same as for option M above.

- N** No logging information is kept for the user interface process of any incoming distributed session from the LOCAP named in this definition..

#### Transaction log file level

This field controls transaction logging of in-bound distributed transaction processing (DTP) sessions.

- M** Minimal logging information is recorded in the transaction log files of inbound distributed sessions from the LOCAP named in this definition.
- A** All logging information is recorded in the transaction log files of inbound distributed sessions from the LOCAP named in this definition.
- N** No logging information is kept for transactions run as part inbound distributed sessions from the LOCAP named in this definition.

Normally, TIP/ix creates the log file in the user's home directory under the name `debug.termname` (where `termname` is the name of the terminal being used).

However, if the `DTPUNIXUSER` parameter in `tipix.conf` (which specifies which UNIX user id to use for in-bound DTP sessions) is set, the log file is created in the home directory of the `DTPUNIXUSER`.

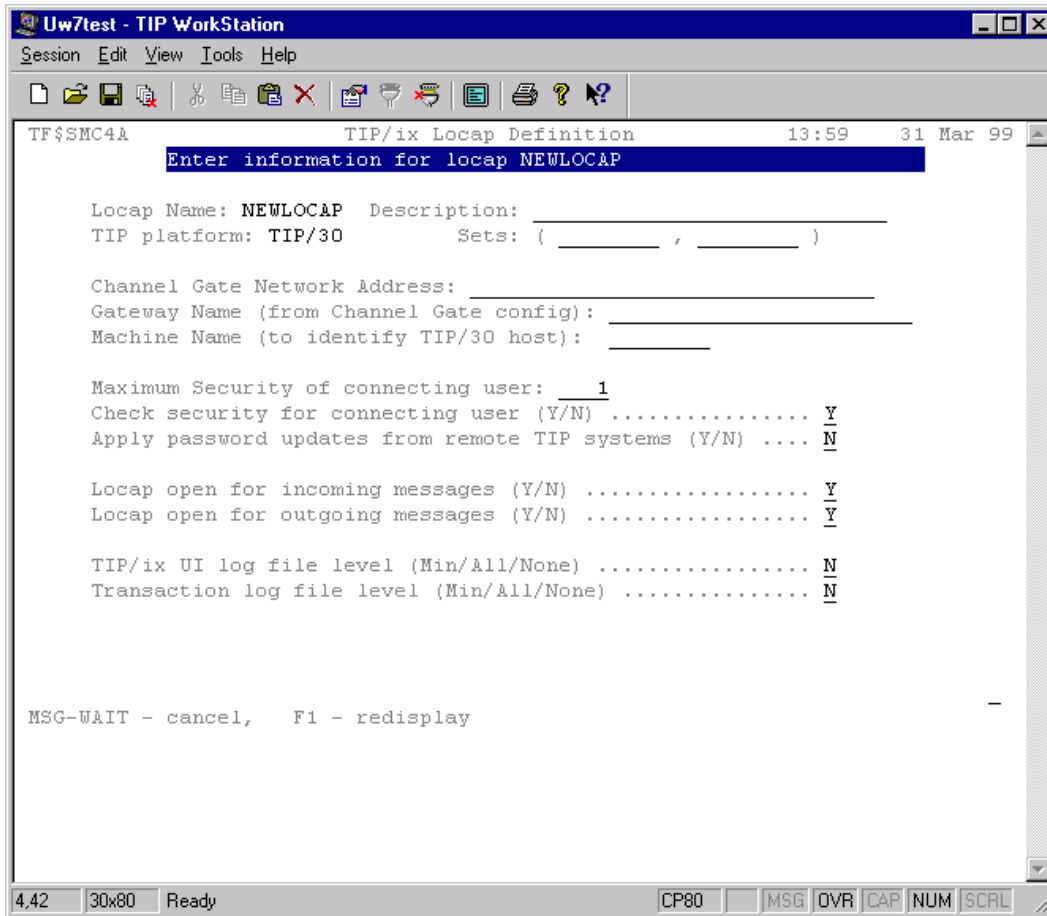
All transactions run during an in-bound session are logged to the same file.

This logging is equivalent to the logging for a local session started by running:

```
tipix -D debug.termname.
```

#### Defining a TIP/30 LOCAP

The following screen is displayed when a request is made to add, display, change, or delete a TIP/30 LOCAP definition.



```

Uw7test - TIP WorkStation
Session Edit View Tools Help
TF$SMC4A TIP/ix Locap Definition 13:59 31 Mar 99
Enter information for locap NEWLOCAP

Locap Name: NEWLOCAP Description: _____
TIP platform: TIP/30 Sets: (_____ , _____)

Channel Gate Network Address: _____
Gateway Name (from Channel Gate config): _____
Machine Name (to identify TIP/30 host): _____

Maximum Security of connecting user: 1
Check security for connecting user (Y/N) Y
Apply password updates from remote TIP systems (Y/N) N

Locap open for incoming messages (Y/N) Y
Locap open for outgoing messages (Y/N) Y

TIP/ix UI log file level (Min/All/None) N
Transaction log file level (Min/All/None) N

MSG-WAIT - cancel, F1 - redisplay

4.42 30x80 Ready CP80 MSG OVR CAP NUM SCRL

```

At the bottom of the screen is a summary of function keys and their actions.

At any time you may enter MSG WAIT to cancel the function and return to the main menu screen.

Pressing F1 causes the screen to be re-displayed (Caution! any data entered on the screen will be lost).

Fill in the information for the LOCAP definition and press XMIT to add the definition to the TIP\$SYS file.

The following is a description of each field in the preceding screen.

#### **LOCAP Name**

The name of the LOCAP. This is used to identify TIP systems in a network. Must be unique in the network.

#### **Description**

Optional information describing the TIP system (LOCAP).

#### **TIP platform**

Must be one of the valid LOCAP types.

**TIP/IX**

TIP system on UNIX

**TIP/30**

TIP system on OS/3

**TIP/AS**

TIP system on a Application Server.

**TIP/CS**

TIP system on a Connection Server.

**Sets** This entry is used to identify a set of records in the TIP\$SYS file that are in some way related. For example you could mark all TIP\$SYS records used in a payroll application by filling in the set name with "PAYROLL". The intention is that these records can be migrated as a set from one TIP/ix system to another using the "tippack" transaction.

**Channel Gate Network Address**

The network address of the Channel Gate device connected to the System/80 host where this TIP/30 system is located.

This value will match the NODE name for the channel gate device in the PEPGATE job stream on the System/80.

**Gateway Name**

This name must match the Gateway Name from the Channel Gate configuration. You must also make sure that the name is in UPPERCASE.

**Machine Name**

This field is required when multiple System/80's are connected to the Channel Gate device. This is used to determine which System/80 that the above Gateway Name refers to.

**Maximum Security of connecting user**

All users connecting from this LOCAP will have their security reduced to this level if their own security level is higher (numerically less than) this value.

This field can be a number between 1 and 255 or one of the following mnemonics which equate to the values shown.

**TECH** 1  
**MAST** 9  
**SYST** 19  
**PROG** 29  
**APPL** 32

**Check security for connecting user**

- Y** Only allow user ids that have been defined on the current TIP/ix system (via [smuser](#)) are permitted to connect from the LOCAP named in this definition. The security level and groups will be set according to the local definition of connecting user ids (subject to the Maximum Security restriction).
- N** No restriction on users connecting from the LOCAP named in this definition. The security level of connecting users is not altered unless it exceeds the Maximum Security setting in this LOCAP definition. The active groups for a connecting user are set to the first two active groups at the time the connection was made.

**Apply password updates**

Respond "Y" to allow password updates to be propagated from the LOCAP named in this definition. The local TIP system will accept password updates for a user id from a remote TIP system.

This feature is not currently implemented.

**Keep LOCAP open for incoming msgs**

- Y** Allow incoming connections from the LOCAP named in this definition to the current TIP system.
- N** Reject any attempts to connect to the current LOCAP from the LOCAP named in this definition.

**Keep LOCAP open for outgoing msgs**

- Y** Attempt to satisfy requests to connect to the LOCAP named in this definition from the current TIP system.
- N** Reject any attempts to connect to the LOCAP named in this definition.

**TIP/ix UI log file level**

- M** Record minimal logging information for the user interface process of any incoming distributed session from the LOCAP named in this definition.
- A** Record all possible logging information for the user interface process of any incoming distributed session from the LOCAP named in this definition.
- N** No logging information is kept for the user interface process of any incoming distributed session from the LOCAP named in this definition..

**Transaction log file level**

- M** Minimal logging information is recorded in the transaction log files of inbound distributed sessions from the LOCAP named in this definition.

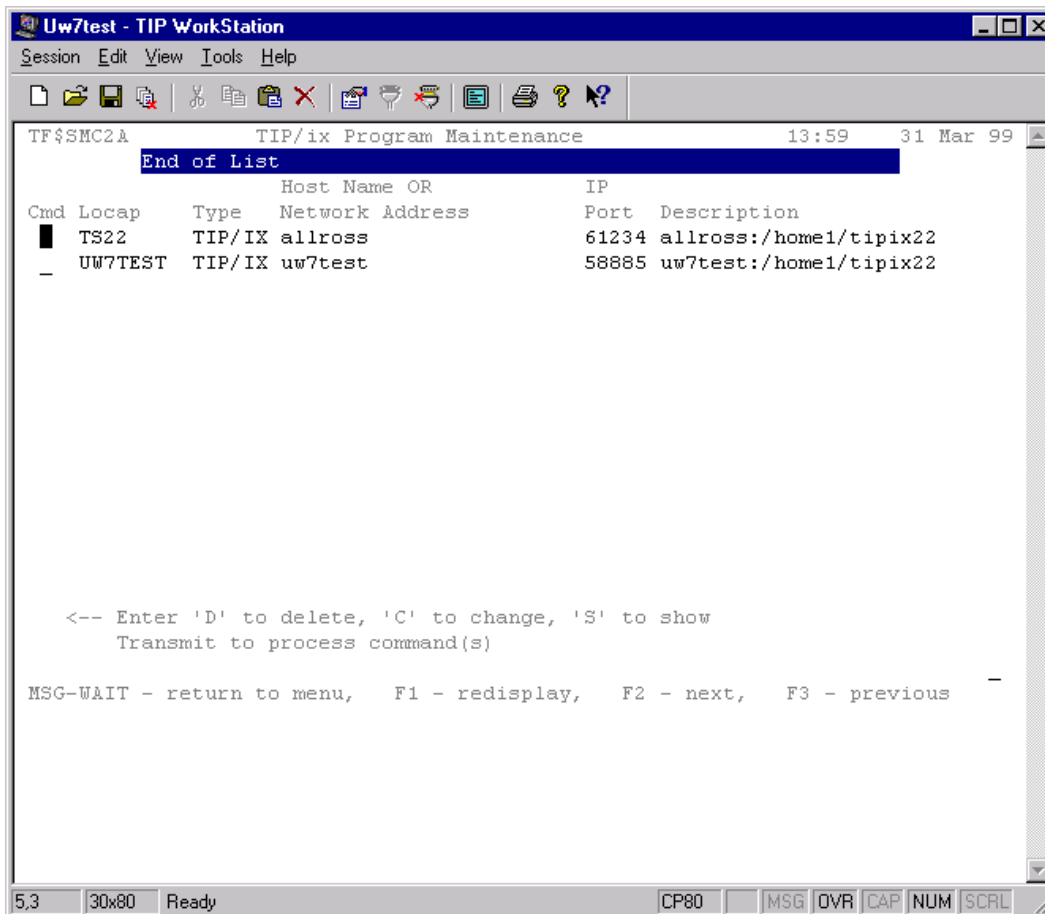
- A** All logging information is recorded in the transaction log files of inbound distributed sessions from the LOCAP named in this definition.
- N** No logging information is kept for transactions run as part inbound distributed sessions from the LOCAP named in this definition.

## List LOCAP Definitions

From the **smlocap** menu enter "LI" for the function to be performed (List) and a name to use as the starting point for the list. This does not have to be a complete name. For example a name of "F" could be used to start listing LOCAP definitions with a name greater than (or equal to) "F".

The list function displays a single line summary for each LOCAP definition showing the logical (**smlocap**) LOCAP name, the LOCAP description, and the LOCAP of the TIP system where the LOCAP is serviced, the server transaction, and some status information about the LOCAP. The status information indicates whether the LOCAP is open or closed, held or not held, and the number of items on the LOCAP at the current LOCAP.

The following screen is displayed in response to the list command.



```

Uw7test - TIP WorkStation
Session Edit View Tools Help
TF$SMC2A TIP/ix Program Maintenance 13:59 31 Mar 99
End of List
Host Name OR IP
Cmd Locap Type Network Address Port Description
TS22 TIP/IX allross 61234 allross:/home1/tipix22
UW7TEST TIP/IX uw7test 58885 uw7test:/home1/tipix22

<-- Enter 'D' to delete, 'C' to change, 'S' to show
Transmit to process command(s)

MSG-WAIT - return to menu, F1 - redisplay, F2 - next, F3 - previous
5,3 30x80 Ready CP80 MSG OVR CAP NUM SCRL

```

Press MSG WAIT to return to the main menu screen.

Press F2 to display the next screen-full of LOCAPs.

Press F3 to display the previous screen-full of LOCAPs.

You can enter single character commands next to the LOCAPs and when you XMIT the commands will be processed sequentially (one at a time) from the top of the display. The commands that can be entered are:

- C** Change LOCAP definition
- D** Delete LOCAP definition
- S** Show LOCAP definition

If no commands are entered and you press XMIT next to a LOCAP on the list display then this is interpreted as a show (S) request for this LOCAP definition.

## Display a LOCAP Definition

**smlocap** provides two ways to display a LOCAP definition.



From the **smlocap** menu enter “S” for the function to be performed (show) and the name of the LOCAP definition to be displayed. (“UW7TEST” in our example).

Use the list function to list LOCAP definitions and type “S” next to the LOCAP definition(s) that you want to display.

The following screen is displayed in response to a request to display a LOCAP definition.

```

Uw7test - TIP WorkStation
Session Edit View Tools Help
TF$SMC3A TIP/ix Locap Definition 14:00 31 Mar 99

Locap Name: UW7TEST Description: uw7test:/home1/tipix22
TIP platform: TIP/IX Sets: (,)

Host Name ("uname -n"): uw7test
IP Port Number: 58885

Maximum Security of connecting user: TECH
Check security for connecting user (Y/N) Y
Apply password updates from remote TIP systems (Y/N) N

Locap open for incoming messages (Y/N) Y
Locap open for outgoing messages (Y/N) Y

TIP/ix UI log file level (Min/All/None) N
Transaction log file level (Min/All/None) N

MSG-WAIT - exit, F1 - redisplay, F2 - next, F3 - previous, F4 - update

```

Press MSG WAIT to return to the **smlocap** screen where the show request was made.

Press F2 to display the next LOCAP definition.

Press F3 to display the previous LOCAP definition.

Press F4 to update the LOCAP definition. The screen will be redisplayed with the data fields padded with underscores.

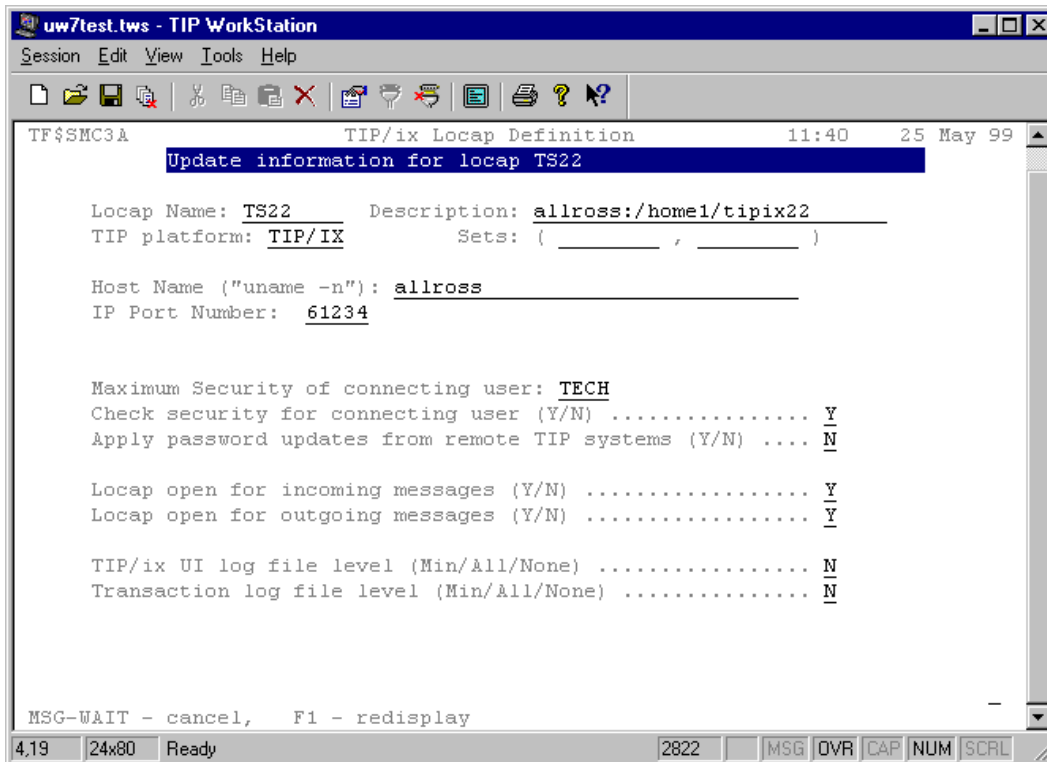
## Change a LOCAP Definition

**smlocap** provides two ways to change (or update) a LOCAP definition.

From the **smlocap** menu enter “CH” for the function to be performed (change) and the name of the LOCAP definition to be changed. (“TS22” in our example).

Use the list function to list LOCAP definitions and type “C” next to the LOCAP definition(s) that you want to change.

The following screen is displayed in response to a request to change a LOCAP definition.



```

uw7test.tws - TIP WorkStation
Session Edit View Tools Help
TF$SMC3A TIP/ix Locap Definition 11:40 25 May 99
Update information for locap TS22

Locap Name: TS22 Description: allross:/home1/tipix22
TIP platform: TIP/IX Sets: (,)

Host Name ("uname -n"): allross
IP Port Number: 61234

Maximum Security of connecting user: TECH
Check security for connecting user (Y/N) Y
Apply password updates from remote TIP systems (Y/N) N

Locap open for incoming messages (Y/N) Y
Locap open for outgoing messages (Y/N) Y

TIP/ix UI log file level (Min/All/None) N
Transaction log file level (Min/All/None) N

MSG-WAIT - cancel, F1 - redisplay
4,19 24x80 Ready 2822 MSG OVR CAP NUM SCRL

```

Press MSG WAIT to cancel the update and return to the **smlocap** screen where the change request was made.

Fill in any desired changes and press XMIT to change the LOCAP definition. For a description of the fields on the LOCAP definition refer to the section on [“Adding a LOCAP Definition”](#).

If the LOCAP name is changed then the original LOCAP definition is left unchanged and a new LOCAP definition is added for the new name.

## Delete a LOCAP Definition

**smlocap** provides two ways to delete a LOCAP definition.

1. From the **smlocap** menu enter “DE” for the function to be performed (delete) and the name of the LOCAP definition to be deleted. (“TS22” in our example).

- Use the list function to list LOCAP definitions and type “D” next to the LOCAP definition(s) that you want to delete.

The following screen is displayed in response to a request to delete a LOCAP definition.

```

Uw7test - TIP WorkStation
Session Edit View Tools Help
TF$SMC3A TIP/ix Locap Definition 14:00 31 Mar 99

Locap Name: TS22 Description: allross:/home1/tipix22
TIP platform: TIP/IX Sets: (,)

Host Name ("uname -n"): allross
IP Port Number: 61234

Maximum Security of connecting user: TECH
Check security for connecting user (Y/N) Y
Apply password updates from remote TIP systems (Y/N) N

Locap open for incoming messages (Y/N) Y
Locap open for outgoing messages (Y/N) Y

TIP/ix UI log file level (Min/All/None) N
Transaction log file level (Min/All/None) N

MSG-WAIT - exit, F1 - redisplay, F2 - next, F3 - previous, F4 - update

```

Press MSG WAIT to cancel the delete request and return to the **smlocap** screen where the delete request was made.

Press F2 to confirm that this is the LOCAP definition that you want to delete.

## Print LOCAP Definitions

To print LOCAP definitions enter “PR” as the function to be performed (print) on the **smlocap** menu display. A LOCAP name can also be supplied to be used as the point to start printing LOCAP definitions from. After pressing **XMIT**, you will be prompted to fill in the print options and report style.

When you select the Print function from **smlocap** the following screen is displayed. Enter the desired print options and press transmit to print TIP/ix LOCAP definition information.

```

Uw7test - TIP WorkStation
Session Edit View Tools Help
TF$SMC7A T I P / i x - Locap Maintenance 14:01 31 Mar 99
=====
 Print Locap records

Set options and transmit to begin printing.

1. Starting printing from this locap name UW7TEST
2. Stop printing at this locap name _____
3. Report Style: S for Summary reporting
 D for Detailed reporting
 L for Last update report
 xxxxx for customized TQL report xxxxx
 Blank for interactive TQL session S
 YY MM DD
4. Include any locap record added or changed since 90/01/01
5. Print File to send output to PRNTR

MSG-WAIT - cancel print, F1 - redisplay, XMIT - submit print request

8,60 30x80 Ready CP80 MSG OVR CAP NUM SCRL

```

The following is a description of the fields in the preceding screen.

**Start at LOCAP name**

**Stop at LOCAP name**

Range of LOCAP definitions to include in report. For example if set to "K" and "R" respectively then all LOCAP names in the range K to R will be included in the report.

**Report styles:**

- S** Summary report that prints a single line of information about each LOCAP.
- D** Detailed report that includes all information in the LOCAP definition.
- L** Report date, time, and user id performing last update to the LOCAP definitions.

**Report style <report name>**

Any valid report name in the TQL program TIPSYSL. This allows you to modify the TQL program and select your own reports.

Fill in LASTUPDT for a TQL report showing the user id, date, and time of the last update for each LOCAP definition.

**<blank>**

Leave this entry blank to initiate an interactive TQL session with the program TIPSYSL.

**Change Date**

Only report LOCAP definitions that have changed since this date.

**Print File**

Print file to route report to. This must be a valid TIPPRINT destination. Print files are defined by smprint and of course aliases can be set up using smsec. AUX0 can be used to direct report output to the terminal.

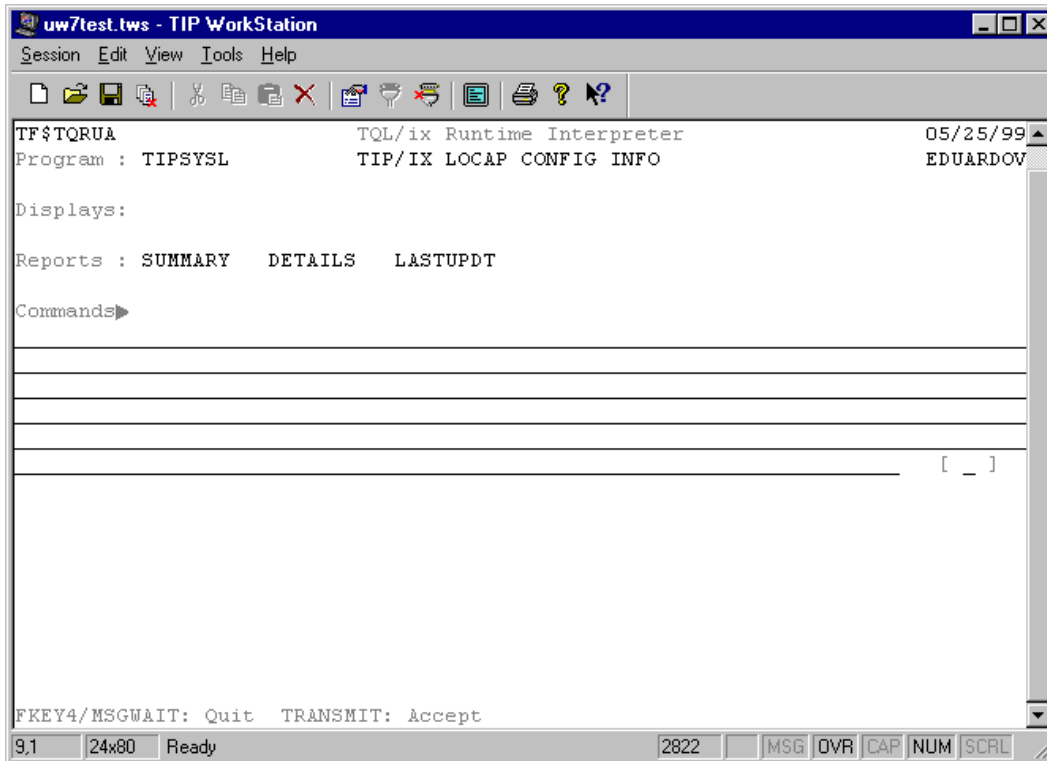
The print function is implemented by the supplied TQL program TIPSYSL. If this program is not compiled and available then the print function will not work.

## Browse LOCAP Definitions with TQL

The **TIPSYSL** TQL program enables you to interrogate LOCAP definitions with ad hoc TQL queries. This TQL program can be run either:

- Directly from the TIP/ix command prompt by entering "TQL TIPSYSL"
- From the smlocap menu enter "BR" for the function to be performed (browse).

The following screen is displayed when you run **TIPSYSL**.



At this point you can make any valid TQL request including:

- Request a predefined report.
- Make an ad hoc request to list (or print) fields from the LOCAP definitions.

See the **TIP/ix TQL Reference** for details on interacting with TQL.

The predefined reports include:

#### **SUMMARY**

Each LOCAP definition is summarized into a single line on the report.

#### **DETAILS**

For each LOCAP definition the report includes all fields from the LOCAP definition.

#### **LASTUPDT**

A report showing who last updated LOCAP definitions (and when).

In addition to using the predefined reports it is possible to perform ad hoc queries based on field settings in LOCAP definitions. The following is a list of LOCAP definition attributes and the corresponding field names:

| LOCAP Definition Attribute | TQL Field Names |
|----------------------------|-----------------|
| LOCAP Name                 | LO-NAME         |

|                                     |                  |
|-------------------------------------|------------------|
| Description                         | LO-CMT           |
| TIP platform                        | LO-PLATFORM      |
| Sets (1st one)                      | LO-SET (1)       |
| Sets (2nd one)                      | LO-SET (2)       |
| Host Name                           | LO-HOST-NAME     |
| IP Port Number                      | LO-IP-PORT       |
| Maximum Security of connecting user | LO-MAXSEC        |
| Check security for connecting user  | LO-CHK-SECUR     |
| Apply password updates              | LO-UPDT-PASS     |
| Keep LOCAP open for incoming msgs   | LO-OPEN-INCOMING |
| Keep LOCAP open for outgoing msgs   | LO-OPEN-OUTGOING |
| TIP/ix UI log file level            | LO-UILOG         |
| Transaction log file level          | LO-TRANLOG       |

### Examples:

List TIP/ix LOCAPs sorted by IP port number

```
SUMMARY IF LO-PLATFORM-IX SORT BY LO-IP-PORT
```

List LOCAPs that the current TIP/ix system will attempt to connect to.

```
LIST (LO-NAME LO-HOST-NAME LO-CMT)
IF LO-OPEN-OUTGOING = "Y"
```

Print LOCAP details report on auxiliary printer for LOCAP definitions with names in the range A to M.

```
DETAILS FROM "A" TO "M" ON AUX1
```

---

## smprint - Define A Print File

The **smprint** transaction is used to define print files that may be accessed by TIP/ix transaction programs (and IMS/90 and IMS/1100 transaction programs). Every print file that is to be used by transaction programs must be defined to TIP/ix. These print file definitions are kept in a file known to TIP/ix as TIP\$SYS.

You should use this program to define the TIP/ix default printer, **PRNTR**, and any other print destinations used by your programs.

A typical TIP/ix transaction uses the TIPPRINT interface to perform printing and TIPPRINT uses the TIPFCS interface to access print files. In the TIPFCS interface file names are defined to be 1 to 8 upper case

characters. TIP/ix uses security and file definition information to determine what processing to perform for a given print destination (file).

When referencing a file with TIPFCS (or TIPPRINT) the name must match an entry in the TIP/ix security file (TIP\$SEC). Each file entry in the security file references an entry in the TIP/ix configuration file (TIP\$SYS) which in turn defines the processing associated with the print destination.

## ASSOCIATING TIPFCS PRINT FILE NAME TO PROCESSING



The above association also applies to programs running under IMS emulation since requests to access files from IMS programs are converted into the appropriate TIPFCS request.

The possible processing options for print destinations are:

- direct (pipe) the output to a UNIX command
- write the output to a file in a UNIX directory
- write the output to a file in a UNIX directory and execute a UNIX command.

File look up in the security file is based on the user's access which is determined by the groups and security level(s) assigned to the user. Refer to [smsec](#) and [smuser](#) for more details.

**smprint** must be run from within the TIP/ix shell (tipix).

### Syntax:

```
smprint [function] [printfile]
```

If a valid function is supplied then **smprint** will perform that function and then terminate when that function has been completed. Most functions require a print file name to act upon.

If no function is supplied then **smprint** displays the following printer maintenance menu screen.



```

Uw7test - TIP WorkStation
Session Edit View Tools Help
TF$SMPOA T I P / i x - Printer Maintenance 14:01 31 Mar 99
=====
Enter the function to be performed: _
Enter Print File Name: _____

Functions
Ad : Add Printer definition
BR : Invoke TQL for ad hoc reports
Ch : Change Printer definition
DE : Delete Printer definition
Sh : Show Printer definition
Li : List Printer definitions
Pr : Print Printer definitions
* : Add Printer definitions in data entry mode
En : End program
Qu : End program and LOGOFF TIP/ix

MSG-WAIT - Exit Program, F1 - Redisplay, F4 - Logoff

4,55 30x80 Ready CP80 MSG OVR CAP NUM SCRL

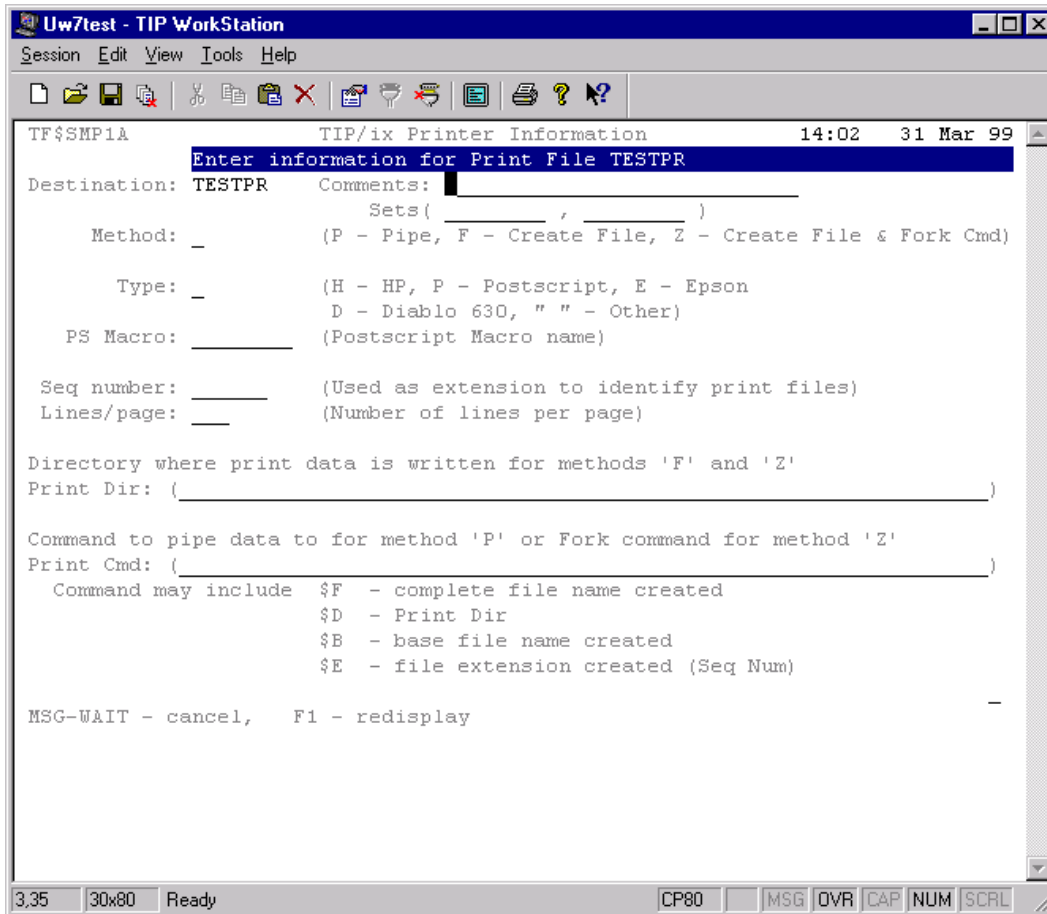
```

From this menu you may perform all the traditional maintenance functions on the printer definitions. You may add new printer definitions; change, delete, show or print current printer definitions and list a summary of all current printer definitions.

In the preceding screen, the function to be performed can be abbreviated to the portion shown in upper case. For example the function *add* can be entered as “A”.

## Add a Printer Definition

To define a new printer definition, enter “A” for the function to be performed (Add) and the name of the print destination (“TESTPR” in our example). After pressing **XMIT**, the following screen appears to allow you to enter the printer definition information.



```

Uw7test - TIP WorkStation
Session Edit View Tools Help
TF$SMP1A TIP/ix Printer Information 14:02 31 Mar 99
Enter information for Print File TESTPR
Destination: TESTPR Comments:
Sets(,)
Method: _ (P - Pipe, F - Create File, Z - Create File & Fork Cmd)
Type: _ (H - HP, P - Postscript, E - Epson
D - Diablo 630, " " - Other)
PS Macro: (Postscript Macro name)
Seq number: (Used as extension to identify print files)
Lines/page: (Number of lines per page)
Directory where print data is written for methods 'F' and 'Z'
Print Dir: ()
Command to pipe data to for method 'P' or Fork command for method 'Z'
Print Cmd: ()
Command may include $F - complete file name created
$D - Print Dir
$B - base file name created
$E - file extension created (Seq Num)
MSG-WAIT - cancel, F1 - redisplay
3,35 30x80 Ready CP80 MSG OVR CAP NUM SCRL

```

At the bottom of the screen is a summary of function keys and their actions.

At any time you may enter MSG WAIT to cancel the function and return to the main menu screen.

Pressing F1 causes the screen to be re-displayed (Caution! any data entered on the screen will be lost).

Fill in the information for the printer definition and press XMIT to add the definition to the TIP\$SYS file.

If the printer definition is added successfully and no security record exists for the print destination (in TIP\$SEC) then a screen will be displayed prompting for the security information. Refer to [smsec](#) for details about security records.

The following is a description of each field in the preceding screen.

**Destination:**

The name of the printer definition in the TIP\$SYS file. This is the name that will be referenced from a security record in the TIP\$SEC file. This name can be different from the TIPFCS name used to access the printer. See chart at the beginning of the section on smprint.

**Comments:**

Optional information describing the print file definition.

**Sets:** This entry is used to identify a set of records in the TIP\$SYS file that are in some way related. For example you could mark all TIP\$SYS records used in a payroll application by filling in the set name with "PAYROLL". The intention is that these records can be migrated as a set from one TIP/ix system to another using the "tippack" transaction.

**Method:**

The type of action to be taken when TIP/ix sends output to this print destination.

- P** Output is to be directed ("piped") to the command specified in the "Print Cmd" field.
- F** Output is to be sent to a UNIX file. The filename will be composed of the logical printer name, a period, and the sequence number (for example, "MYOUTPUT.1"). The UNIX file will be created in the directory specified by the "Print Dir" field.
- Z** Output is to be sent to a UNIX file (as with method "F" above) and then when an FCS-CLOSE is issued to the print file the UNIX command given in the "Print Cmd" field will be run.

**Type:** Fill in the value for one of the supported printer types to have TIP/ix to send a printer setup sequence when an FCS-OPEN is issued to the print destination.

Leave this field blank if no print setup is desired. This might be the case directing the output to a file or a when directing the output to a command that does not do printing.

- H** Use printer setup sequence for an HP Laserjet printer.
- P** Use printer setup sequence for a Postscript printer. This option requires that the "PS Macro" field be filled in with the name of the file containing Postscript to download to the printer prior to the print data.
- E** Use printer setup sequence for an Epson printer.
- D** Use printer setup sequence for a Diablo 630 printer.

**<blank>**

No printer setup sequence is sent ahead of the print data.

**PS Macro:**

The name of the file containing the Postscript code to download to the printer when the printer is opened (prior to

sending any print data). This field must be filled in if the Type field is set to "P" (Postscript).

The file name can be specified as a full path and file name or as a relative file name. If the file name is a relative file name (no leading slash) then TIP/ix expects the file to be in the *\$TIPROOT/include* directory and will automatically add an extension of ".ps". If the PS Macro was specified as "arcland" then TIP/ix will use the file "\$TIPROOT/include/arcland.ps".

Ingenet Business Solutions supplies several *sample* postscript code files ("macros") in the *\$TIPROOT/include* directory. The supplied postscript files all have the extension ".ps".

Each of these files is a postscript program to control the printing of the data; for example: arcland.ps is a landscape mode format.

Understanding these postscript programs requires knowledge of Postscript! If you have a postscript printer, but do not understand postscript very well, we suggest you try "arcland".

**Next Seq Num:**

The sequence number. This number is automatically incremented each time the printer is opened (FCS-OPEN which results in a reference to this printer definition). It is used as part of the file name when printer output is sent to a file (method "F" or "Z").

**Lines per page:**

The number of lines per page for this particular printer.

**Default** value is 56 lines per page (if this field is left blank).

**Print Dir:**

The UNIX directory where print files should be created when method "F" or "Z" is specified in the printer definition. You should ensure that this directory exists and is writable by users who intend to use this printer definition. (Consult the entries for the commands "mkdir" and "chmod" in your UNIX manual for information on how to do this).

**Print Cmd:**

The UNIX command to be run for methods "P" or "Z". If you are using method "P", this should be a command which reads its data from its "*standard input*". If you are using method "Z", you will want to use one or more of the following meta characters to indicate to the UNIX command what file is to be printed.

- \$F** The UNIX pathname of the output file. This is equivalent to "\$D/\$B.\$E"
- \$D** The UNIX directory where the print file is placed (as specified in the "Print Dir" field)
- \$B** The UNIX name of the output file without the print directory or the sequence number. This is name specified in the "Destination" field.
- \$E** The sequence number of the output file.

## Defining PRNTR - An Example

The TIP/ix system uses the logical printer name "PRNTR" as a standard default print destination. The printer name "PRNTR" must be defined using **smprint**.

The following example shows how to set up a dumb printer as the PRNTR destination. The example assumes that the UNIX system has been properly set to use the UNIX "lp" spooler (not all UNIX systems have an "lp" spooler!)

```

uw7test.tws - TIP WorkStation
Session Edit View Tools Help
TF$SMP1A TIP/ix Printer Information 15:08 26 May 99
Destination: PRNTR Comments: Default Printer
 Sets(
Method: P (P - Pipe, F - Create File, Z - Create File & Fork Cmd)
Type: (H - HP, P - Postscript, E - Epson
 D - Diablo 630, " " - Other)
PS Macro: (Postscript Macro name)
Seq number: 1 (Used as extension to identify print files)
Lines/page: 0 (Number of lines per page)
Directory where print data is written for methods 'F' and 'Z'
Print Dir: (
Command to pipe data to for method 'P' or Fork command for method 'Z'
Print Cmd: (lp 2> /dev/null
 Command may include $F - complete file name created
 $D - Print Dir
 $B - base file name created
 $E - file extension created (Seq Num)
MSG-WAIT - exit, F2 - next, F3 - previous, F4 - update, F5 - security rec
23.78 24x80 Ready P315 MSG QVR CAP NUM SCRL

```

The printer data will be "Piped" to the UNIX command "lp" (the desired syntax is defined in the "Print Cmd" field). In this example, we have been careful to redirect standard error ("2>") to the null device. This avoids the situation where printer error messages suddenly appear on the screen (where TIP/ix may well be expecting to use MCS!).

Some implementations of the "lp" command may support simpler methods of suppressing error messages (for example, -s is a popular, but by no means universally accepted, switch for that purpose).

## List Printer Definitions

From the **smprint** menu enter "LI" for the function to be performed (List) and a name to use as the starting point for the list. This does not have to be a complete name. For example a name of "B" could be used to start listing printer definitions with a name greater than (or equal to) "B".

The list function displays a single line summary for each printer definition showing the logical (**smprint**) file name, the printer description, and the print destination.

The data displayed in the column titled "Print Destination" on the list display varies with the print method specified. If the method is specified as "P" then the command the heading "PIPE TO" is used and the field "*Print Cmd*" is displayed. If the method is specified as "F" then the heading "WRITE TO" is used and the field "*Print Dir*" is displayed. If the method is specified as "Z" then the command the heading "FORK TO" is used and the field "*Print Cmd*" is displayed.

The following screen is displayed in response to the list command.

```

Uw7test - TIP WorkStation
Session Edit View Tools Help
TF$SMP3A TIP/ix Print File Maintenance 14:06 31 Mar 99
End of List
Print
Cmd File Description Print Destination
_ PRNTR Default Printer PIPE TO: lp 2> /dev/null

<-- Enter 'D' to delete, 'C' to change, 'S' to show, 'G' to get security rec
Transmit to process command(s)

MSG-WAIT - return to menu, F1 - redisplay, F2 - next, F3 - previous

```

Press MSG WAIT to return to the main menu screen.

Press F2 to display the next screen of printer definitions.

Press F3 to display the previous screen of printer definitions.

You can enter single character commands next to the print files and when you press XMIT the commands will be processed sequentially (one at a time) from the top of the display. The commands that can be entered are:

- C** Change printer definition
- D** Delete printer definition
- S** Show printer definition
- G** Get the security record(s) for this printer definition.

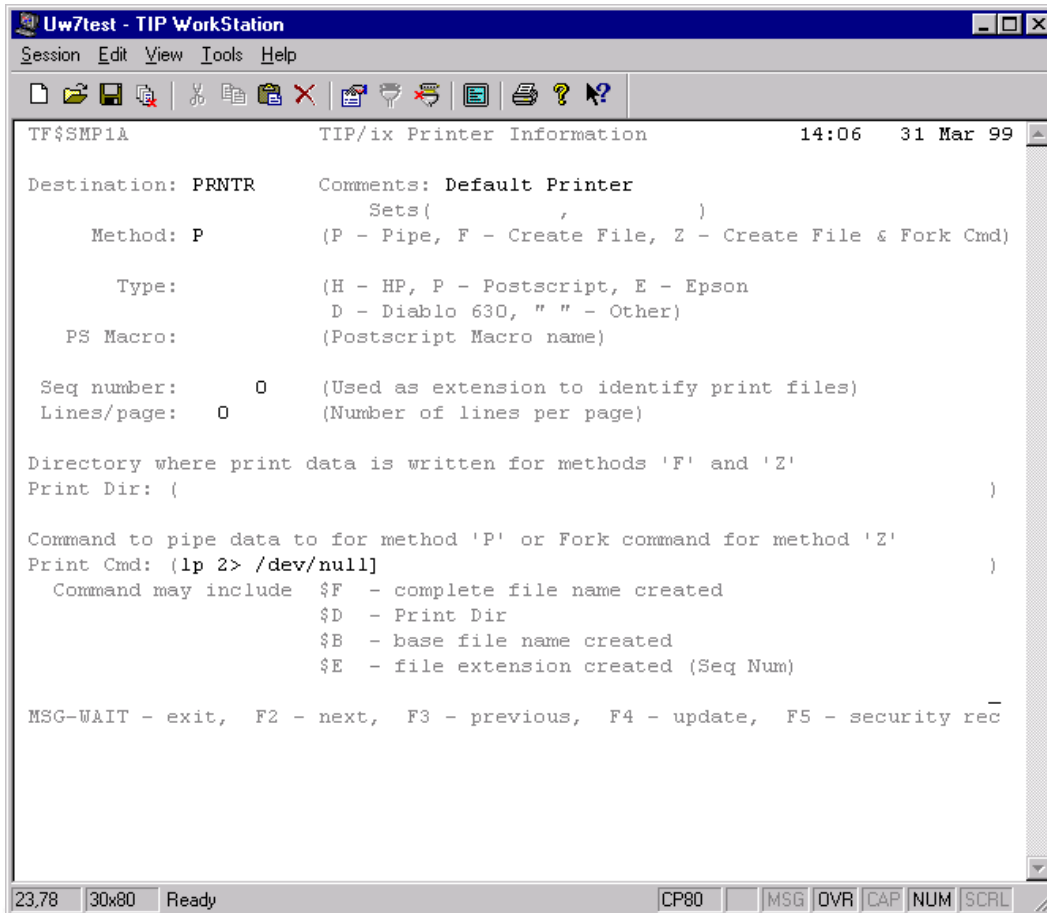
If no commands are entered and you press XMIT next to a printer on the list display then this is interpreted as a show (S) request for this printer definition.

## Display a Printer Definition

**smprint** provides two ways to display a printer definition.

1. From the **smprint** menu enter “S” for the function to be performed (show) and the name of the printer definition to be displayed. (“PRNTR” in our example).
2. Use the list function to list printer definitions and type “S” next to the printer definition(s) that you want to display.

The following screen is displayed in response to a request to display a printer definition.



```

Uw7test - TIP WorkStation
Session Edit View Tools Help
TF$SMP1A TIP/ix Printer Information 14:06 31 Mar 99
Destination: PRNTR Comments: Default Printer
 Sets(
Method: P (P - Pipe, F - Create File, Z - Create File & Fork Cmd)
Type: (H - HP, P - Postscript, E - Epson
 D - Diablo 630, " " - Other)
PS Macro: (Postscript Macro name)
Seq number: 0 (Used as extension to identify print files)
Lines/page: 0 (Number of lines per page)
Directory where print data is written for methods 'F' and 'Z'
Print Dir: (
Command to pipe data to for method 'P' or Fork command for method 'Z'
Print Cmd: (lp 2> /dev/null)
Command may include $F - complete file name created
 $D - Print Dir
 $B - base file name created
 $E - file extension created (Seq Num)
MSG-WAIT - exit, F2 - next, F3 - previous, F4 - update, F5 - security rec
23,78 30x80 Ready CP80 MSG OVR CAP NUM SCRL

```

Press MSG WAIT to return to the **smprint** screen where the show request was made.

Press F2 to display the next printer definition.

Press F3 to display the previous printer definition.

Press F4 to update the printer definition. The screen will be redisplayed with the data fields padded with underscores.

## Change a Printer Definition

**smprint** provides two ways to change (or update) a printer definition.



1. From the **smprint** menu enter “CH” for the function to be performed (change) and the name of the printer definition to be changed. (“PRNTR” in our example).
2. Use the list function to list printer definitions and type “C” next to the printer definition(s) that you want to change.

The following screen is displayed in response to a request to change a printer definition.

```

Uw7test - TIP WorkStation
Session Edit View Tools Help
TIP/ix Printer Information 14:07 31 Mar 99
Update information for Print file PRNTR
Destination: PRNTR Comments: Default Printer
 Sets(_____ , _____)
Method: P (P - Pipe, F - Create File, Z - Create File & Fork Cmd)
Type: _ (H - HP, P - Postscript, E - Epson
 D - Diablo 630, " " - Other)
PS Macro: _____ (Postscript Macro name)
Seq number: _____ 0 (Used as extension to identify print files)
Lines/page: _____ 0 (Number of lines per page)
Directory where print data is written for methods 'F' and 'Z'
Print Dir: (_____)
Command to pipe data to for method 'P' or Fork command for method 'Z'
Print Cmd: (lp 2> /dev/null)
 Command may include $F - complete file name created
 $D - Print Dir
 $B - base file name created
 $E - file extension created (Seq Num)
MSG-WAIT - cancel, F1 - redisplay

```

Press MSG WAIT to cancel the update and return to the **smprint** screen where the change request was made.

Fill in any desired changes and press XMIT to change the printer definition. For a description of the fields on the printer definition refer to the section on [“Adding a Printer Definition”](#).

If the printer definition is changed successfully and no security record exists for the print file (in TIP\$SEC) then a screen will be displayed prompting for the security information. Refer to [smsec](#) for details about security records.

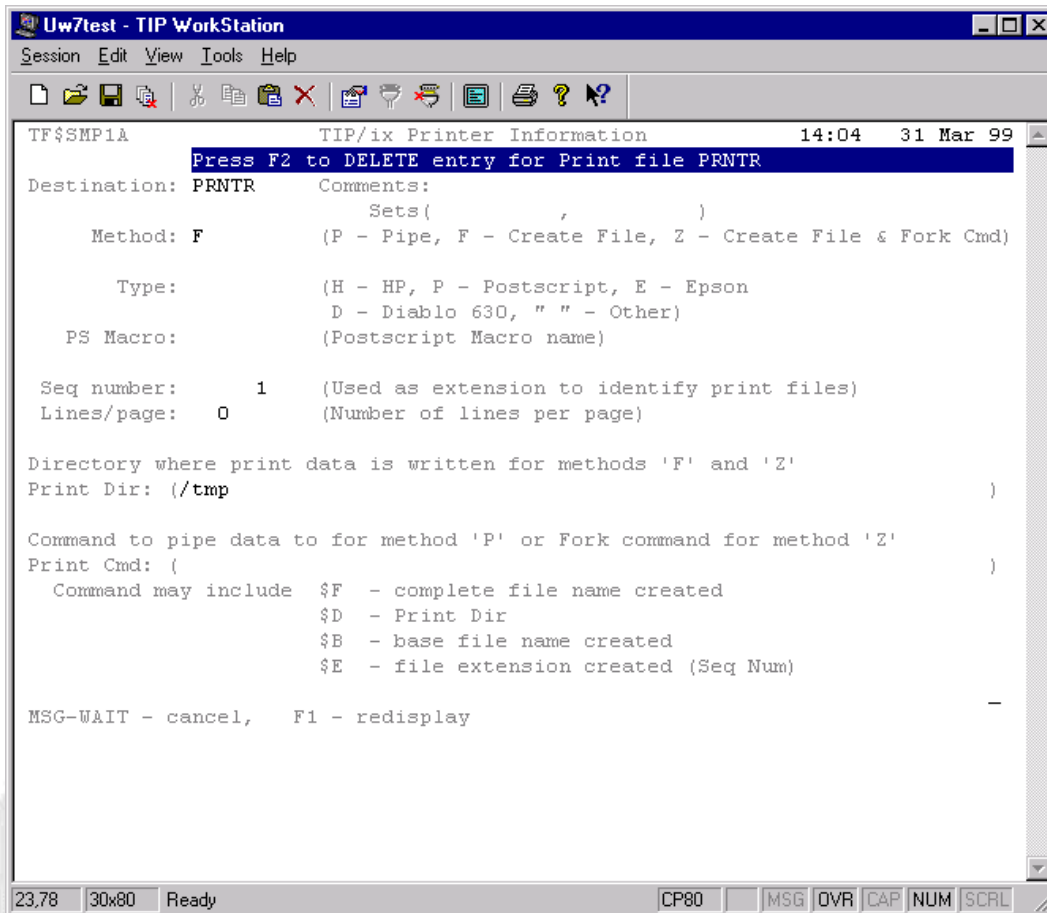
If the printer name is changed then the original printer definition is left unchanged and a new printer definition is added for the new name.

## Delete a Printer Definition

**smprint** provides two ways to delete a printer definition.

1. From the **smprint** menu enter “DE” for the function to be performed (delete) and the name of the printer definition to be deleted. (“PRNTR” in our example).
2. Use the list function to list printer definitions and type “D” next to the printer definition(s) that you want to delete.

The following screen is displayed in response to a request to delete a printer definition.



```

Uw7test - TIP WorkStation
Session Edit View Tools Help
TIP/ix Printer Information 14:04 31 Mar 99
Press F2 to DELETE entry for Print file PRNTR
Destination: PRNTR Comments:
 Sets(
Method: F (P - Pipe, F - Create File, Z - Create File & Fork Cmd)
Type: (H - HP, P - Postscript, E - Epson
 D - Diablo 630, " " - Other)
PS Macro: (Postscript Macro name)
Seq number: 1 (Used as extension to identify print files)
Lines/page: 0 (Number of lines per page)
Directory where print data is written for methods 'F' and 'Z'
Print Dir: (/tmp)
Command to pipe data to for method 'P' or Fork command for method 'Z'
Print Cmd: (
 Command may include $F - complete file name created
 $D - Print Dir
 $B - base file name created
 $E - file extension created (Seq Num)
MSG-WAIT - cancel, F1 - redisplay
23,78 30x80 Ready CP80 MSG OVR CAP NUM SCRL

```

Press MSG WAIT to cancel the delete request and return to the **smprint** screen where the delete request was made.

Press F2 to confirm that this is the printer definition that you want to delete.

If you press F2 and the printer definition is deleted then if there are any security records for this print file definition you will be given the opportunity to delete them. If there is a single security record for the print file it will be displayed and you are requested to press F2 to confirm that the security record is to be deleted. If there is more than one security

record then they are displayed in a summary list (up to 12 will be shown) and a message indicating the number of security records that will be deleted is shown. Pressing F2 will delete all the security records for the printer definition that was deleted.

## Get Security Records for Printer Definitions

**smprint** provides two ways to retrieve security records for printer definitions.

1. Use the list function to list printer definitions and type "G" next to the printer definition(s) for which you want to display the corresponding security information.
2. Display a printer definition (using either the show command or pressing XMIT next to a printer on the list display) and then press F5.

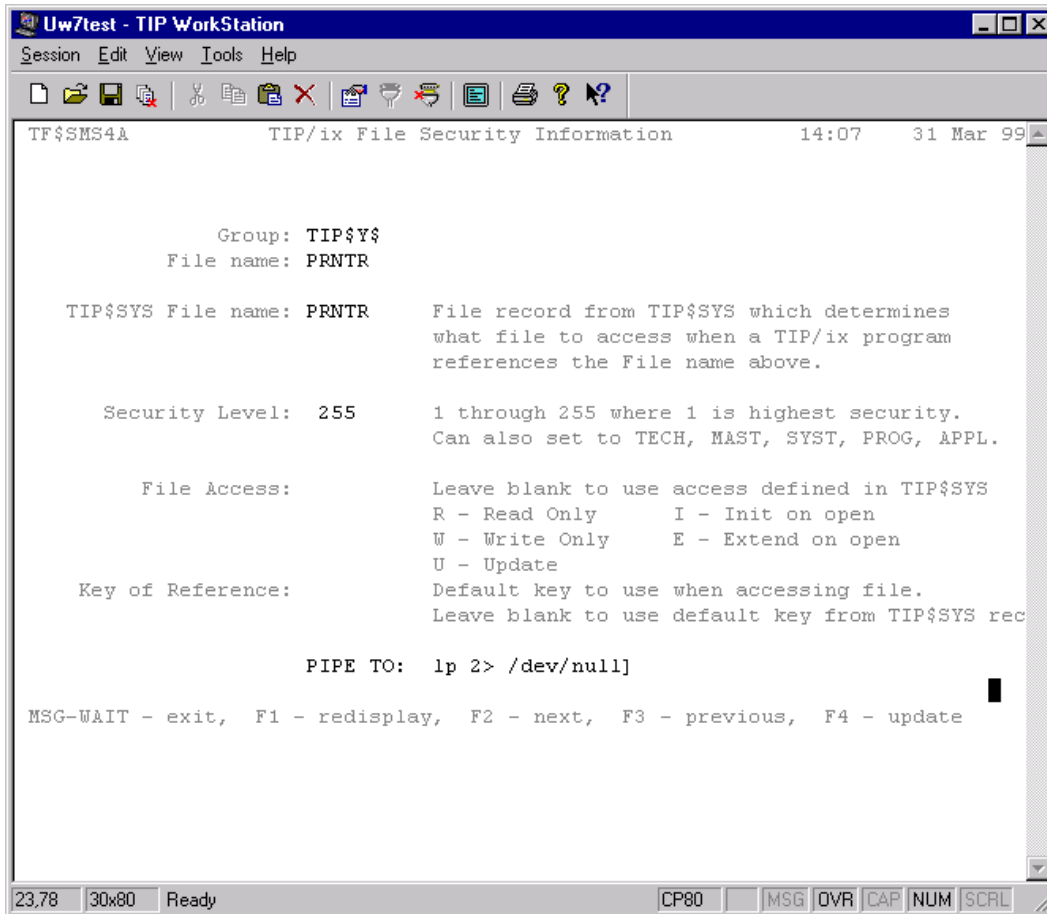
If there is only a single security record that references this printer definition then that security record is displayed.

If there is more than one security record then a list of all security records referencing this printer record will be displayed.

If no security record is defined then you will be asked if you wish to add one (via a Yes/No prompt).

For information about security records refer to the section on the [smsec](#) program

The following screen is displayed if there is a single security record referencing this printer definition. (This example shows a request for the security record for the printer definition for PRNTR.)



```

Uw7test - TIP WorkStation
Session Edit View Tools Help
TF$SMS4A TIP/ix File Security Information 14:07 31 Mar 99

Group: TIPY
File name: PRNTR

TIP$SYS File name: PRNTR File record from TIP$SYS which determines
what file to access when a TIP/ix program
references the File name above.

Security Level: 255 1 through 255 where 1 is highest security.
Can also set to TECH, MAST, SYST, PROG, APPL.

File Access: Leave blank to use access defined in TIP$SYS
R - Read Only I - Init on open
W - Write Only E - Extend on open
U - Update

Key of Reference: Default key to use when accessing file.
Leave blank to use default key from TIP$SYS rec

PIPE TO: lp 2> /dev/null]

MSG-WAIT - exit, F1 - redisplay, F2 - next, F3 - previous, F4 - update
23,78 30x80 Ready CP80 MSG OVR CAP NUM SCRL

```

Press MSG WAIT to return to the printer definition screen where the request was made for the security information.

Press F2 to display the next security record in the TIP\$SEC file (navigating by the 3rd key which is the associated TIP\$SYS file name). The subsequent security records may or may not relate to the printer definition for which security information was originally requested.

Press F3 to display the previous security record in the TIP\$SEC file (navigating by the 3rd key which is the associated TIP\$SYS file name). The subsequent security records may or may not relate to the printer definition for which security information was originally requested.

Press F4 to update the security record. The screen will be redisplayed with the data fields padded with underscores.

The following screen is displayed in response to a request for security information if there are multiple security records referencing the printer definition. (This example shows a request for the security record for the printer definition for PRNTR.)

```

uw7test.tws - TIP WorkStation
Session Edit View Tools Help
=====
TF$SMSBA TIP/ix Security Maintenance Utility 15:41 26 May 99
=====
End of List
=====
Command Type Name Group Level Unix Path

FILE PRNTR PRNTR TIPY 255 PIPE TO:lp -s -d postie
FILE PRNTR PRINTER TIPY 255 PIPE TO:lp -s -d postie
=====
<-- Enter 'D' to delete, 'C' to change, 'S' to show
Transmit to process command(s)

MSG-WAIT - return to menu, F1 - redisplay, F2 - next, F3 - previous
6,5 24x80 Ready P315 MSG OVR CAP NUM SCRL

```

Press MSG WAIT to return to the printer definition screen where the request was made for the security information.

Press F2 to display the next screen of security records for the TIP\$SYS (printer) file name

Press F3 to display the previous screen of security records for the TIP\$SYS (printer) file name.

You can enter single character commands next to the security records and when you XMIT the commands will be processed sequentially (one at a time) from the top of the display. The commands that can be entered are:

- C** Change security entry
- D** Delete security entry
- S** Show security entry

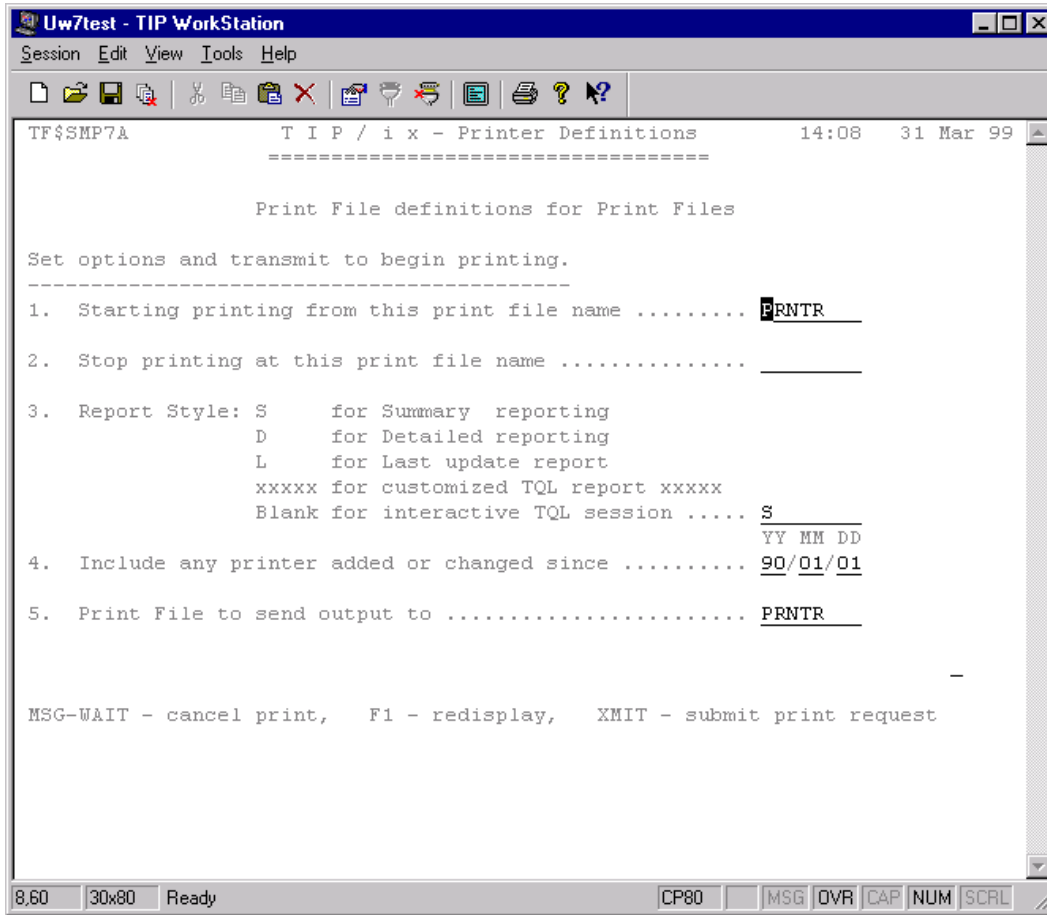
If no commands are entered and you press XMIT next to a security entry then this is interpreted as a show (S) request for that security entry.

## Print Printer Definitions

To print printer definitions enter "PR" as the function to be performed (print) on the **smprint** menu display. A printer name can also be supplied to be used as the point to start printing printer definitions from. After

pressing **XMIT**, you will be prompted to fill in the print options and report style.

When you select the Print function from **smprint** the following screen is displayed. Enter the desired print options and press transmit to print TIP/ix printer definition information.



```

Uw7test - TIP WorkStation
Session Edit View Tools Help
TF$SMP7A T I P / i x - Printer Definitions 14:08 31 Mar 99
=====
Print File definitions for Print Files

Set options and transmit to begin printing.

1. Starting printing from this print file name PRNTR
2. Stop printing at this print file name
3. Report Style: S for Summary reporting
 D for Detailed reporting
 L for Last update report
 xxxxx for customized TQL report xxxxx
 Blank for interactive TQL session S
 YY MM DD
4. Include any printer added or changed since 90/01/01
5. Print File to send output to PRNTR

MSG-WAIT - cancel print, F1 - redisplay, XMIT - submit print request
8,60 30x80 Ready CP80 MSG OVR CAP NUM SCRL

```

The following is a description of the fields in the preceding screen.

#### Start at print file

#### Stop at print file

Range of printer (print file) definitions to include in report. For example if set to “K” and “R” respectively then all printer definitions in the range K to R will be included in the report.

#### Report style

Fill in one of the valid report styles or leave blank.

- S** Summary report that prints a single line of information about each printer definition.
- D** Detailed report that includes all information in the printer definition as well as all security records for the printer definitions.

**L** Report date, time, and user id performing last update to print definitions.

**<report name>**

Any valid report name in the TQL program TIPSYSYSP. This allows you to modify the TQL program and select your own reports. Fill in LASTUPDT for a TQL report showing the user id, date, and time of the last update for each printer definition.

**<blank>**

Leave this entry blank to initiate an interactive TQL session with the program TIPSYSYSP

**Change Date**

Only report printer definitions that have changed since this date.

**Print File**

Print file to route report to. This must be a valid TIPPRINT destination. Print files are defined by smprint and of course aliases can be set up using smsec. AUX0 can be used to direct report output to the terminal.

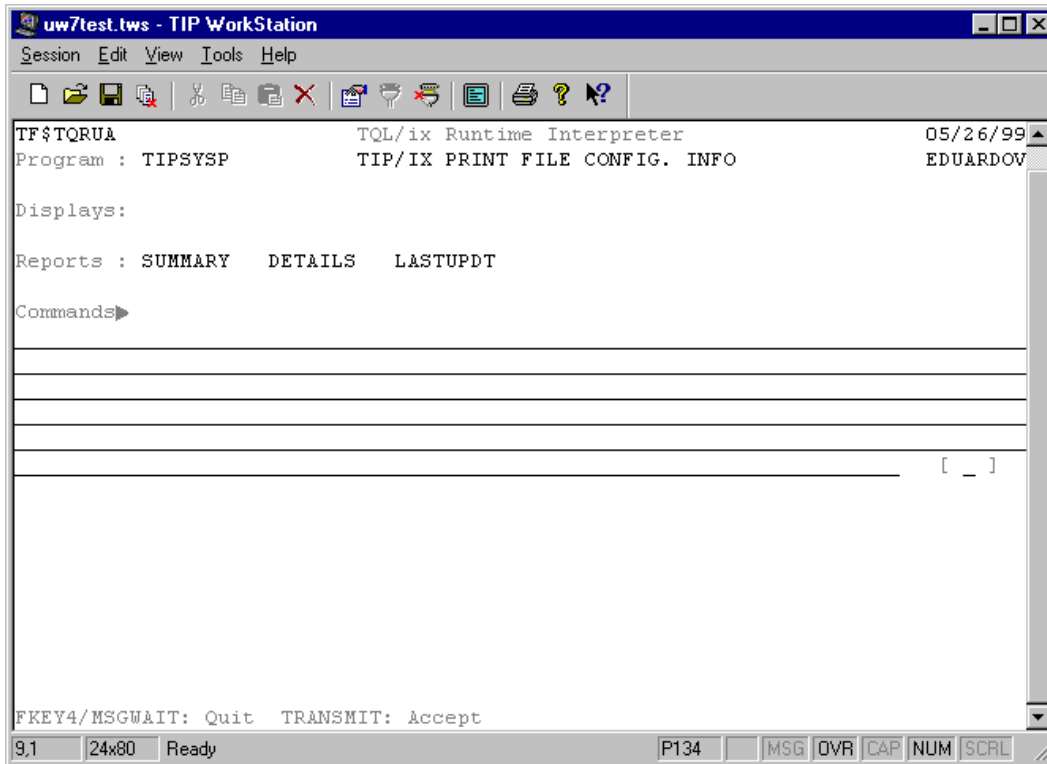
The print function is implemented by the supplied TQL program TIPSYSYSP. If this program is not compiled and available then the print function will not work.

## Browse Print Definitions with TQL

The **TIPSYSYSP** TQL program enables you to interrogate Print definitions with ad hoc TQL queries. This TQL program can be run either:

- Directly from the TIP/ix command prompt by entering "TQL TIPSYSYSP"
- From the smprint menu enter "BR" for the function to be performed (browse).

The following screen is displayed when you run **TIPSYSYSP**.



At this point you can make any valid TQL request including:

- Request a predefined report.
- Make an ad hoc request to list (or print) fields from the print definitions.

See the *TIP/ix TQL Reference* for details on interacting with TQL.

The predefined reports include:

#### **SUMMARY**

Each print definition is summarized into a single line on the report.

#### **DETAILS**

For each print definition the report includes all fields from the definition and any security records referencing it.

#### **LASTUPDT**

A report showing who last updated print definitions (and when).

In addition to using the predefined reports it is possible to perform ad hoc queries based on field settings in print definitions. When formulating queries it is useful to know that print file definitions are a special case of file definition where the 88 level PF-PRNTR is true.

The following is a list of print definition attributes and the corresponding field names:



|                            |                 |
|----------------------------|-----------------|
| Print Definition Attribute | TQL Field Names |
| Destination                | PF-NAME         |
| Comments                   | PF-CMT          |
| Sets (1st one)             | PF-SET (1)      |
| Sets (2nd one)             | PF-SET (2)      |
| Method                     | PF-PRINT-METHOD |
| Type                       | PF-PRINT-TYPE   |
| PS Macro                   | PF-PS-MACRO     |
| Next Seq Number            | PF-PRINT-SEQ    |
| Lines per page             | PF-PRINT-LPP    |
| Print Dir                  | PF-LBL          |
| Print Cmd                  | PF-PRINT-CMD    |

### Examples:

List print file definitions that direct the output to a file and fork a command.

```
SUMMARY IF PF-PRINT-METHOD = "Z"
```

List some fields from print file definitions

```
LIST (PF-NAME PF-PRINT-METHOD PF-CMT) IF
PF-PRNTR
```

Print details report on auxiliary printer for print file definitions with names in the range A to M.

```
DETAILS FROM "A" TO "M" ON AUX1
```

---

## smprog - Define a Program

The **smprog** transaction defines the operating characteristics of transaction programs that run under TIP/ix. Every transaction program must be defined to TIP/ix. These transaction program definitions are kept in a file known to TIP/ix as TIP\$SYS.

TIP/ix transaction names can be from 1 to 8 uppercase characters. TIP/ix uses security and program definition information to relate the 8 character transaction name to a UNIX executable module.

When a user makes a request to run a transaction TIP/ix looks for an entry for that transaction in the TIP/ix security file (TIP\$SEC). Each program entry in the security file references a program entry in the TIP/ix

configuration file (TIP\$SYS) which in turn defines the UNIX executable to use.

In general, there should only be one TIP\$SYS record for each binary. However, there can be multiple TIP\$SEC records pointing to the same TIP\$SYS record to permit you to assign different security levels and transaction names to any particular binary.

## ASSOCIATING TRANSACTION NAME TO UNIX BINARY



Transaction look up in the security file is based on the user's access which is determined by the groups and security level(s) assigned to the user. Refer to [smsec](#) and [smuser](#) for more details.

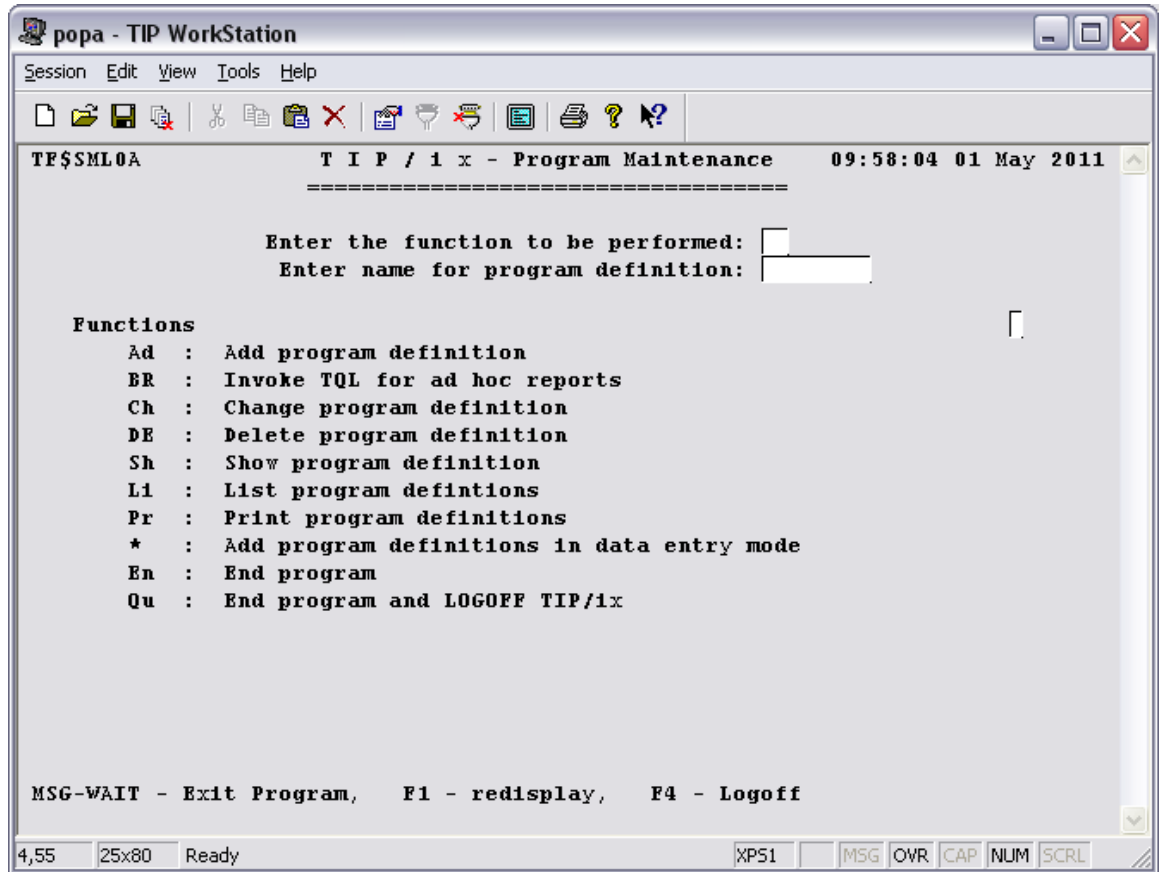
**smprog** must be run from within the TIP/ix shell (tipix).

### Syntax:

```
smprog [function] [programname]
```

If a valid function is supplied then **smprog** will perform that function and then terminate when that function has been completed. Most functions require a program name to act upon.

If no function is supplied then **smprog** displays the following program maintenance menu screen.



From this menu you may perform all the traditional maintenance functions on the program definitions. You may add new program definitions; change, delete, show or print current program definitions and list a summary of all current program definitions.

In the preceding screen, the function to be performed can be abbreviated to the portion shown in upper case. For example the function *add* can be entered as "A".

### Add a Program Definition

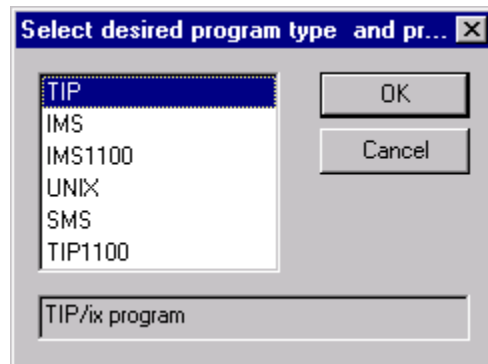
To define a new program definition, enter "A" for the function to be performed (add) and the name of the program ("NEWPROG" in our example). After pressing **XMIT**, you will be prompted to select the type of program to add from a list of valid program types. The program types available on TIP/ix are:

| PROGRAM TYPE | DESCRIPTION                                                                      |
|--------------|----------------------------------------------------------------------------------|
| TIP          | TIP/ix or TIP/30 program.                                                        |
| IMS          | IMS/90 program. TIP/ix emulates IMS functions (CALLs) for programs of this type. |

|         |                                                                                    |
|---------|------------------------------------------------------------------------------------|
| IMS1100 | IMS/1100 program. TIP/ix emulates IMS functions (CALLs) for programs of this type. |
| UNIX    | A standalone UNIX program. Program does not use either the TIP API or IMS API.     |
| SMS     | IMS transactions which use SMS/90 calls should be defined as type SMS.             |

Move the cursor to the desired program type and press either of ENTER or XMIT.

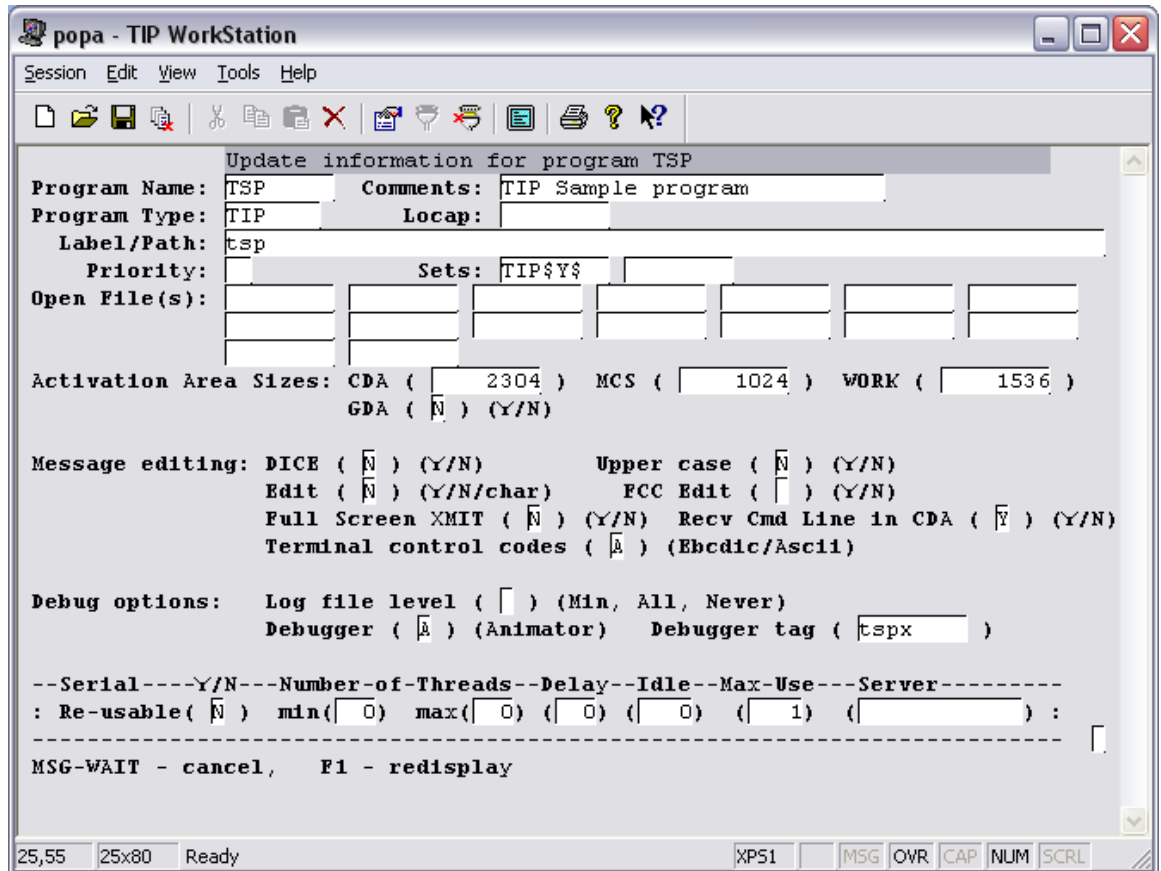
**Selecting a program type:**



After you have selected the program type a screen will be displayed to allow you to enter the program definition information applicable to the program type you have selected.

### Defining a TIP Program

The following screen is displayed when a request is made to add, display, change, or delete a TIP program definition.



At the bottom of the screen is a summary of function keys and their actions.

At any time you may enter MSG WAIT to cancel the function and return to the main menu screen.

Pressing F1 causes the screen to be re-displayed (Caution! any data entered on the screen will be lost).

Fill in the information for the program definition and press XMIT to add the definition to the TIP\$SYS file.

If the program definition is added successfully and no security record exists for the program (in TIP\$SEC) then a screen will be displayed prompting for the security information. Refer to [smsec](#) for details about security records.

The following is a description of each field in the preceding screen.

#### Program Name

The name of the program definition in the TIP\$SYS file. This is a logical name that will be referenced from a

security record in the TIP\$SEC file. This name can be different from the transaction name and the name of the UNIX executable.

**Comments**

Optional information describing the program.

**Program Type**

Supply one of the valid program types:

**TIP** TIP/ix or TIP/30 program  
**IMS** IMS/90 program  
**IMS1100**  
IMS/1100 program  
**UNIX** Standalone UNIX program  
**SMS** SMS/90 program.

**LOCAP**

The LOCAP name of the TIP system where the program is to be executed (see [smlocap](#) utility).

It is only necessary to fill in this field if the program is to be run on another TIP system. In this case the program will be executed remotely.

If the LOCAP name is left blank or is the LOCAP name for the current TIP system then the program will be run locally.

If this field is filled in and you exit the field or press XMIT then if the LOCAP does not match the current TIP system then a remote program definition screen will be displayed

**Label/Path**

Specifies the program executable module. Leaving this field blank causes TIP/ix to look for the executable with the same name as the Program Name field.

If the executable module name is the same as the Program Name, except that it is in lower case, you must specify the name in lower case in the Label/Path field.

If a relative path name (no leading slash) is entered then TIP/ix attempts to find this path in the directories specified in the PATH (or TIPPATH) environment variable of the user making the request (see section on TIP/ix Environment Variables).

One method of specifying a full (absolute) path is to use an environment variable to specify the UNIX directory and then append to it the executable file name. The environment variable must be defined in the *tipix.conf* configuration file in the **\$TIPROOT/conf** directory.

For example:  
In *tipix.conf* define: **HR=/apps/bin/hr**  
Enter Label as: **\$HR/newprog**

**Priority**

Not used.

**Sets** This entry is used to identify a set of records in the TIP\$SYS file that are in some way related. For example you could mark all TIP\$SYS records used in a payroll application by filling in the set name with "PAYROLL". The intention is that these records can be migrated as a set from one TIP/ix system to another using the ["tippack"](#) transaction.

**Open File(s)**

Files that are to be made available to the transaction program (via TIPFCS) when it starts up. It is not necessary to fill in file names used by the program as TIP will dynamically assign them to the process as they are accessed.

Filling in this field may be of some use in maintaining your application (so that you can see what programs use what files).

**CDA** The size of the Continuity Data Area (CDA) in bytes.

Use "precob -u" to set this value automatically. Memory allocation always rounded up to the nearest 256 bytes.

**MCS** The size of the program's MCS area in bytes.

Use "precob -u" to set this value automatically. Memory allocation always rounded up to the nearest 256 bytes.

**WORK**

The size of the program's WORK area in bytes.

Use "precob -u" to set this value automatically. Memory allocation always rounded up to the nearest 256 bytes.

**GDA** Enter "Y" in this field if the TIP/ix program uses the Global Data Area (GDA). The program cannot access the GDA unless a "Y" is entered here. Memory allocation always rounded up to the nearest 256 bytes.

**DICE**

**Y** Turn on DICE processing for this program. DICE sequences will be placed into input messages received by the program. DICE sequences will be acted upon in output messages that they appear in. However, this does not prevent DICE from

subsequently being removed from input messages if EDIT=Y or EDIT=char was specified for the program.

- N** Turn off DICE processing for this program. DICE will not be placed into input messages from received by the program. DICE sequences will NOT be acted upon in output messages from this program. This should be specified for programs that relied on DICE=OFF in terminal definitions on OS/3.

### Upper Case

- Y** All characters in input messages received by TIPTERM will be translated to upper case.
- N** Do not alter the case of any characters in input messages received by TIPTERM.

### Edit

- Y** Input messages received by TIPTERM or IMS emulation will be edited to remove all communication control characters such as DICE and FCC sequences. Multiple spaces are reduced to a single space.
- N** Input messages received by TIPTERM and IMS programs are NOT edited. DICE codes and FCC sequences are not removed from the input message.
- char** Input messages received by TIPTERM or IMS emulation will be edited to remove all communication control characters such as DICE and FCC sequences. All occurrences of the specified character are removed from input messages. Spaces are left as is when this style of editing is used.

### FCC Edit

- Y** Remove FCC sequences from input messages received by TIPTERM or IMS emulation.
- N** Leave FCC sequences intact.

### Full Screen XMIT

- Y** Force the entire screen to be received for every screen format input message. This is NOT desirable for any application where processing is dependent on the cursor position when XMIT is pressed.

### Recv Cmd Line in CDA



- Y When the program starts up the command line is parameterized and placed in the program's CDA
- N When the program starts up the command line is NOT placed in the program's CDA. The command line will be available as an input message which the program can read using TIPTERM.

### Terminal control codes

- A Program will use ASCII values for terminal control codes (example: SOE or Start Of Entry) in constructing TIPTERM messages.
- E Program will use EBCDIC values for terminal control codes (example: SOE or Start Of Entry) in constructing TIPTERM messages.

### Log file level

- M Minimal logging information is recorded in log files for this transaction when logging is requested.  
(tipix -D or tipix -d)
- A All possible logging information is recorded in log files for this transaction when logging is requested.  
(tipix -D or tipix -d)
- N No logging information is recorded in log files for this transaction **even** if logging is requested.  
(tipix -D or tipix -d)  
This is handy for some of system utilities like **fin**, **clear**, etc.

### Debugger

- A When request is made to run the program then execute the program using the Micro Focus Animator.
- Y Same as A for Micro Focus animator.
- N Turn invocation of debugger off
- O OpenCOBOL debugger
- C COBOL-IT debugger

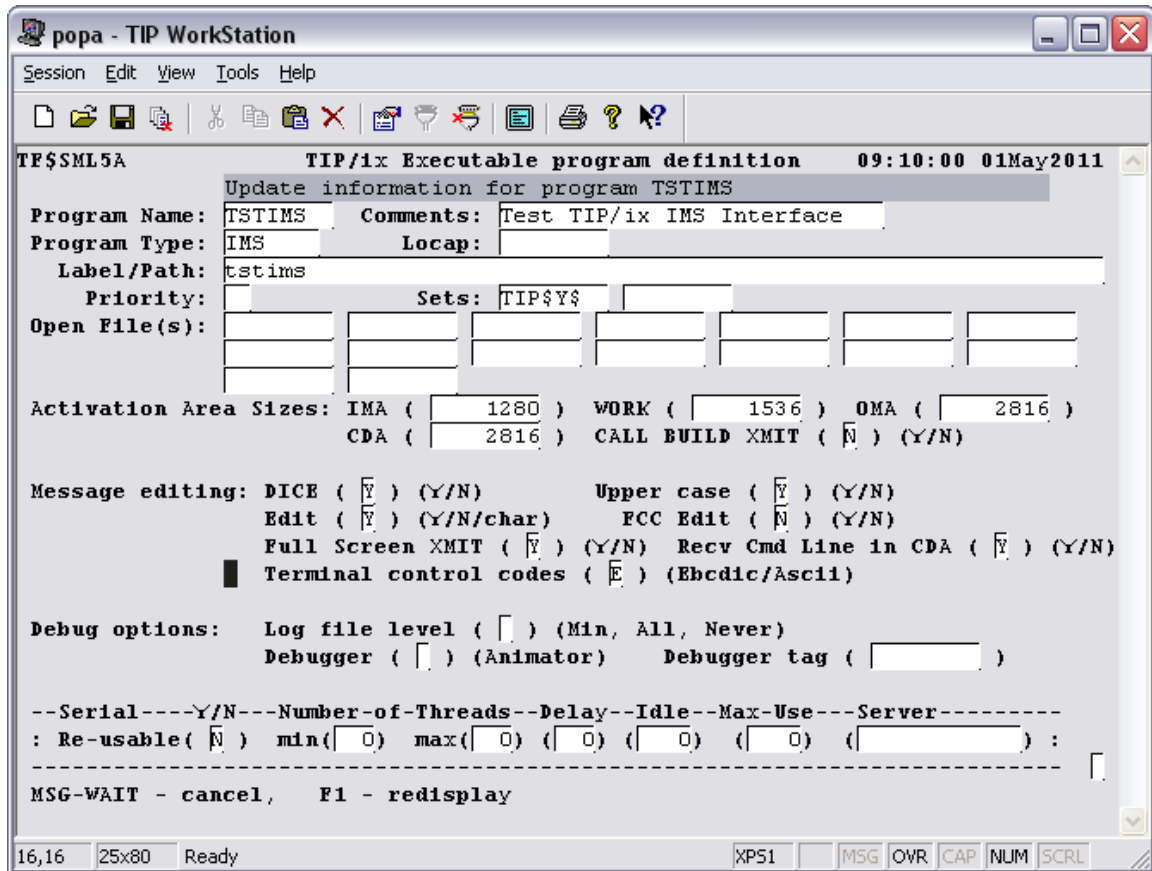
### Debugger tag

Value to be put into an environment variable for connection to the COBOL debugger when the transaction is executed. For Micro Focus COBOL the COBANIMSRV variable is set.

See the section on [Serial Reusable Execution Mode](#) for a description of the remaining parameters.

## Defining an IMS/90 or IMS/1100 Program

The following screen is displayed when a request is made to add, display, change, or delete an IMS/90 or IMS/1100 program definition.



```

popa - TIP WorkStation
Session Edit View Tools Help
[Icons]
TF$SML5A TIP/ix Executable program definition 09:10:00 01May2011
Update information for program TSTIMS
Program Name: TSTIMS Comments: Test TIP/ix IMS Interface
Program Type: IMS Locap: []
Label/Path: tstims
Priority: [] Sets: TIPY
Open File(s): [] [] [] [] [] [] []
Activation Area Sizes: IMA ([] 1280) WORK ([] 1536) OMA ([] 2816)
 CDA ([] 2816) CALL BUILD XMIT ([]) (Y/N)
Message editing: DICE ([]) (Y/N) Upper case ([]) (Y/N)
 Edit ([]) (Y/N/char) FCC Edit ([]) (Y/N)
 Full Screen XMIT ([]) (Y/N) Recv Cmd Line in CDA ([]) (Y/N)
 Terminal control codes ([]) (Ebcdic/Ascii)
Debug options: Log file level ([]) (Min, All, Never)
 Debugger ([]) (Animator) Debugger tag ([])
--Serial----Y/N---Number-of-Threads--Delay--Idle--Max-Use---Server-----
: Re-usable([]) min([] 0) max([] 0) ([] 0) ([] 0) ([] 0) ([]):
MSG-WAIT - cancel, F1 - redisplay
16,16 25x80 Ready XPS1 MSG OVR CAP NUM SCRL

```

At the bottom of the screen is a summary of function keys and their actions.

At any time you may enter MSG WAIT to cancel the function and return to the main menu screen.

Pressing F1 causes the screen to be re-displayed (Caution! any data entered on the screen will be lost).

Fill in the information for the program definition and press XMIT to add the definition to the TIP\$SYS file.

If the program definition is added successfully and no security record exists for the program (in TIP\$SEC) then a screen will be displayed prompting for the security information. Refer to [smsec](#) for details about security records.

The following is a description of each field in the preceding screen.

### Program Name

The name of the program definition in the TIP\$SYS file.  
This is a logical name that will be referenced from a

security record in the TIP\$SEC file. This name can be different from the transaction name and the name of the UNIX executable.

**Comments**

Optional information describing the program.

**Program Type**

Supply one of the valid program types:

**TIP** TIP/ix or TIP/30 program  
**IMS** IMS/90 program  
**IMS1100**  
IMS/1100 program  
**UNIX** Standalone UNIX program

**LOCAP**

The LOCAP name of the TIP system where the program is to be executed (see smlocap utility).

It is only necessary to fill in this field if the program is to be run on another TIP system. In this case the program will be executed remotely.

If the LOCAP name is left blank or is the LOCAP name for the current TIP system then the program will be run locally.

If this field is filled in and you exit the field or press XMIT then if the LOCAP does not match the current TIP system then a remote program definition screen will be displayed

**Label/Path**

Specifies the program executable module. Leaving this field blank causes TIP/ix to look for the executable with the same name as the Program Name field.

If the executable module name is the same as the Program Name, except that it is in lower case, you must specify the name in lower case in the Label/Path field.

If a relative path name (no leading slash) is entered then TIP/ix attempts to find this path in the directories specified in the PATH (or TIPPATH) environment variable of the user making the request (see section on TIP/ix Environment Variables).

One method of specifying a full (absolute) path is to use an environment variable to specify the UNIX directory and then append to it the executable file name. The environment variable must be defined in the *tipix.conf* configuration file in the *\$TIPROOT/conf* directory.

For example:  
In *tipix.conf* define: **HR=/apps/bin/hr**  
Enter Label as: **\$HR/newprog**

**Priority**

Not used.

**Sets** This entry is used to identify a set of records in the TIP\$SYS file that are in some way related. For example you could mark all TIP\$SYS records used in a payroll application by filling in the set name with "PAYROLL". The intention is that these records can be migrated as a set from one TIP/ix system to another using the "[tippack](#)" transaction.

**Open File(s)**

Files that are to be made available to the transaction program when it starts up. It is not necessary to fill in file names used by the program as TIP will dynamically assign them to the process as they are accessed.

Filling in this field may be of some use in maintaining your application (so that you can see what programs use what files).

**IMA** The size of the program's Input Message Area (IMA) in bytes. Use "precob -u" to set this value automatically. Memory allocation always rounded up to the nearest 256 bytes.

**WORK**

The size of the program's WORK area in bytes.

Use "precob -u" to set this value automatically. Memory allocation always rounded up to the nearest 256 bytes.

**OMA** The size of the program's Output Message Area (OMA) in bytes. Use "precob -u" to set this value automatically. Memory allocation always rounded up to the nearest 256 bytes.

**CDA** The size of the Continuity Data Area (CDA) in bytes.

Use "precob -u" to set this value automatically. Memory allocation always rounded up to the nearest 256 bytes.

**CALL BUILD XMIT**

**Y** TIP/ix will display output message when an IMS program does a CALL "BUILD". IMS programs that do a CALL "RETURN" immediately after a CALL "BUILD" will operate correctly with this setting. This is the default setting.

**N** TIP/ix will not display the output message when an IMS program does a CALL "BUILD". This setting is necessary for programs that alter the output message created by CALL "BUILD" prior to doing a CALL "RETURN".

**DICE**

**Y** Turn on DICE processing for this program. DICE sequences will be placed into input messages received by the program. DICE sequences will be acted upon in output messages that they appear in. However, this does not prevent DICE from subsequently being removed from input messages if EDIT=Y or EDIT=char was specified for the program.

**N** Turn off DICE processing for this program. DICE will not be placed into input messages from received by the program. DICE sequences will NOT be acted upon in output messages from this program. This should be specified for programs that relied on DICE=OFF in terminal definitions on OS/3.

**Upper Case**

**Y** All characters in input messages received will be translated to upper case.

**N** Do not alter the case of any characters in input messages.

**Edit**

**Y** Input messages received by TIPTERM or IMS emulation will be edited to remove all communication control characters such as DICE and FCC sequences. Multiple spaces are reduced to a single space.

**N** Input messages received by TIPTERM and IMS programs are NOT edited. DICE codes and FCC sequences are not removed from the input message.

**char** Input messages received by TIPTERM or IMS emulation will be edited to remove all communication control characters such as DICE and FCC sequences. All occurrences of the specified character are removed from input messages. Spaces are left as is when this style of editing is used.

**FCC Edit**

- Y Remove FCC sequences from input messages received by TIPTERM or IMS emulation.
- N Leave FCC sequences intact.

#### Full Screen XMIT

- Y Force the entire screen to be received for every screen format input message. This is NOT desirable for any application where processing is dependent on the cursor position when XMIT is pressed.

#### Recv Cmd Line in CDA

- Y This mode is not recommended for IMS programs as IMS programs do not expect the command line in to arrive in the CDA.
- N When the program starts up the command line is NOT placed in the program's CDA. The command line will be available as an input message which the program can read using TIPTERM.

#### Terminal control codes

- A Program will use ASCII values for terminal control codes (example: SOE or Start Of Entry) in constructing TIPTERM messages.
- E Program will use EBCDIC values for terminal control codes (example: SOE or Start Of Entry) in constructing TIPTERM messages.

#### Log file level

- M Minimal logging information is recorded in log files for this transaction when logging is requested. (tipix -D or tipix -d)
- A All possible logging information is recorded in log files for this transaction when logging is requested. (tipix -D or tipix -d)
- N No logging information is recorded in log files for this transaction **even** if logging is requested. (tipix -D or tipix -d)  
This is handy for some of system utilities like **fin**, **clear**, etc.

#### Debugger

- A When request is made to run the program then execute the program using the Micro Focus Animator.
- Y Same as A for Micro Focus animator.
- N Turn invocation of debugger off
- O OpenCOBOL debugger
- C COBOL-IT debugger

**Debugger tag**

Value to be put into an environment variable for connection to the COBOL debugger when the transaction is executed. For Micro Focus COBOL the COBANIMSRV variable is set.

See the section on Serial Reusable Execution Mode for a description of the remaining parameters.

**Serial Reusable Execution Mode**

TIP/ix has a program execution style, called Serially Reusable, that can be used to help performance in scheduling frequently used transactions of short duration. This would be beneficial for transactions that process a single input message and then terminate with an output message. IMS programs would meet this criteria.

This execution mode is **not** recommended for **most** TIP/ix programs. One exception would be TIP/ix programs that use TIPDXC to wait for terminal input.

COBOL IMS, IMS/1100, TIP/1100, and DPS/1100 programs can operate as serially reusable. Your native TIP/ix transactions can be defined as REUSEABLE as long as they do not do any terminal I/O (TIPMSGI, TIPTERM etc).

In your IMS programs, you must replace any "CALL RETURN" statements with "EXIT PROGRAM" statements.

**Reusable**

**Y** TIP/ix will re-use UNIX processes to service requests to run this program; this results in savings by avoiding overhead in process creation and destruction. The pool of processes is managed according to remaining parameter settings on this line of screen (To the right of Re-usable).

**Min Threads**

Keep at least this number of processes active to service requests to execute this program. If this field is set to '0' then any server process for this program will be destroyed if it has been inactive for the specified **Idle** time.

**Recommendations:** If the program is used all day and often, try a min of 1 or 2.

If the program is only used during a part of the day, set min to 0. This way, TIP/ix will shut the program down when it is not being used.

**Max Threads**

Maximum number of processes that can be executing this

program at any time. Further requests to run the program will wait until one of the active processes is available.

A process becomes available when an IMS program issues "EXIT PROGRAM" or a TIP/ix program calls "TIPRTN".

**Recommendation:** If you make max too large, there is no advantage to making the program re-usable. Try max at 5, then monitor it with status x. It is good to see some requests waiting because you can improve overall system throughput by throttling transactions that do a lot of I/O.

**Delay** Time in seconds to wait before starting a new process to service a request to run this program when all existing server processes for this program are currently executing.

Set this field to '0' to immediately start another process if all processes for this program are busy. However, a new process will not be created if the maximum number (as specified by *Max Threads*) are already active.

**Idle** Time in seconds to keep an idle process active. If a server process remains idle for this length of time it will be shutdown. A process will not be shutdown if the number of processes for this program is less than or equal to the minimum number (as specified by *Min Threads*).

**Max use**

Maximum number of times that a process will be reused to service requests to run this program. A server process for this program will be destroyed after being reused the specified number of times. This allows new server processes to be created periodically with a fresh data segment.

Set this field to '0' to allow a process to be reused indefinitely.

**Server**

Name of another **smprog** entry to use to determine what program to execute. However, all other information from this **smprog** entry is used to run the program. This allows the same server program to be invoked with different activation area sizes.

Normally, this field can be left blank.

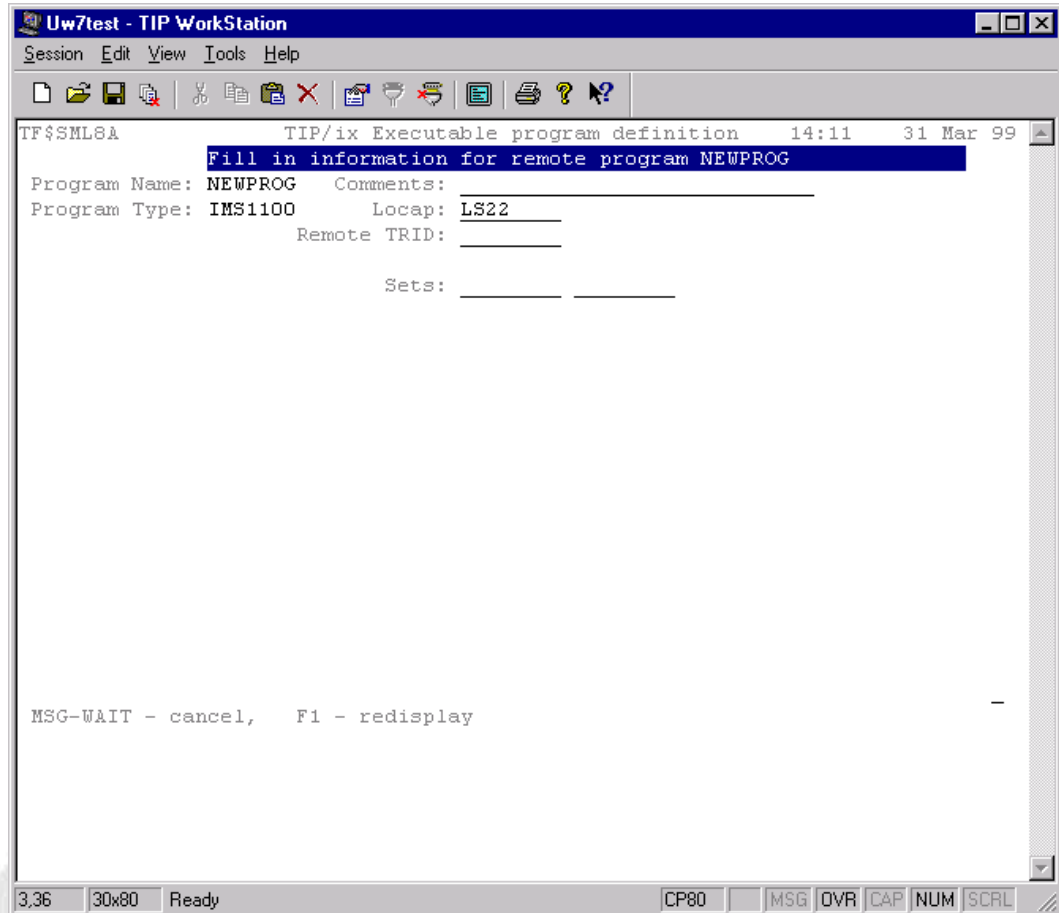
To see how many processes currently exist to service requests to run a program use the UNIX command "**ps -ef | grep Program Name**". The program name must be entered in upper case and match the name of the program definition.



## Defining a Remote Program

A remote program is defined to be a program to be run on another TIP/ix or TIP/30 system. Each TIP system is identified by a LOCAP name. [smlocap](#) is used to associate LOCAP names with TIP systems.

To define a remote program you first make a request to add or change a program record. Then on the program definition screen fill in the LOCAP name of the TIP system where the program is to be run. When a LOCAP name is filled in and it is not the LOCAP of the current TIP/ix system the following screen is displayed.



At the bottom of the screen is a summary of function keys and their actions.

At any time you may enter MSG WAIT to cancel the function and return to the main menu screen.

Pressing F1 causes the screen to be re-displayed (Caution! any data entered on the screen will be lost).

Fill in the information for the remote program definition and press XMIT to add the definition to the TIP\$SYS file.

If the program definition is added successfully and no security record exists for the file (in TIP\$SEC) then a screen will be displayed prompting for the security information. Refer to [smsec](#) for details about security records.

The only attribute that is unique to remote programs is “Remote TRID”.

#### **Remote TRID**

The smprog entry on the remote LOCAP to be used to determine what UNIX file to execute when a request is made to run this program.

## **List Program Definitions**

From the **smprog** menu enter “LI” for the function to be performed (List) and a name to use as the starting point for the list. This does not have to be a complete name. For example a name of “SM” could be used to start listing program definitions with a name greater than (or equal to) “SM”.

The list function displays a single line summary for each program definition showing the logical (**smprog**) program name, the program type, the program comments, and the UNIX executable file associated with the program definition.

The data displayed in the column titled “Executable” on the list display varies with the program type. If the program is a remote program then the LOCAP is displayed instead of the UNIX executable file.

When the UNIX executable name is too long to fit on the list display it will be truncated so that the trailing portion (suffix) is displayed. If executable name is an absolute path it will be truncated to **/.../<suffix>**. If the executable name is a relative path it will be truncated to **.../<suffix>**.

The following screen is displayed in response to the list command.

```

Uw7test - TIP WorkStation
Session Edit View Tools Help
TF$SML2A TIP/ix Program Maintenance 14:12 31 Mar 99

Cmd Program Type Description Executable
- AFT TIP Display Active File Table aft
- ARMCAT TIP TIP/ix catalogue utility armcat
- CALENDAR TIP Display Calendar $TIPROOT/bin/calendar
- CD TIP TIP/ix change directory tipcd

- CLEAR TIP TIP clear screen function tipclear
- CONNECT TIP remote login to other locaps connect
- DDU TIP Disk Display Update utility ddu
- DEFKEY TIP define function keys defkey

- DPAGER TIP1100 DPS Paging processor. dpager
- DTP$RFA TIP dtprfa
- DUMPCAT TIP Dump the TIP/ix catalogue dumpcat
- EOJ TIP Schedule a TIP/ix shutdown eoj

<-- Enter 'D' to delete, 'C' to change, 'S' to show, 'G' to get security rec
Transmit to process command(s)

MSG-WAIT - return to menu, F1 - redisplay, F2 - next, F3 - previous

```

Press MSG WAIT to return to the main menu screen.

Press F2 to display the next screen-full of programs.

Press F3 to display the previous screen-full of programs.

You can enter single character commands next to the programs and when you XMIT the commands will be processed sequentially (one at a time) from the top of the display. The commands that can be entered are:

- C** Change program definition
- D** Delete program definition
- S** Show program definition
- G** Get the security record(s) for this program definition.

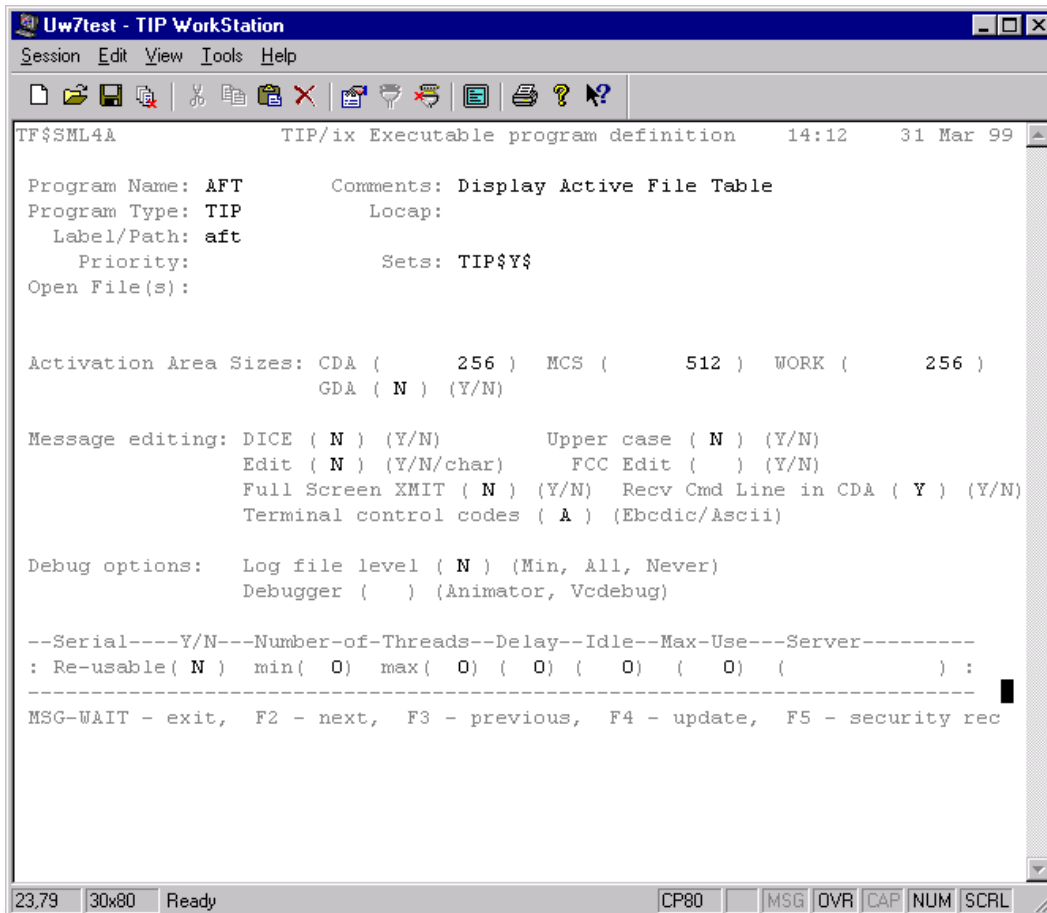
If no commands are entered and you press XMIT next to a program on the list display then this is interpreted as a show (S) request for this program definition.

## Display a Program Definition

**smprog** provides two ways to display a program definition.

1. From the **smprog** menu enter “S” for the function to be performed (show) and the name of the program definition to be displayed. (“AFT” in our example).
2. Use the list function to list program definitions and type “S” next to the program definition(s) that you want to display.

The following screen is displayed in response to a request to display a program definition.



```

Uw7test - TIP WorkStation
Session Edit View Tools Help
TF$SML4A TIP/ix Executable program definition 14:12 31 Mar 99
Program Name: AFT Comments: Display Active File Table
Program Type: TIP Locap:
Label/Path: aft
Priority: Sets: TIPY
Open File(s):

Activation Area Sizes: CDA (256) MCS (512) WORK (256)
GDA (N) (Y/N)

Message editing: DICE (N) (Y/N) Upper case (N) (Y/N)
Edit (N) (Y/N/char) FCC Edit () (Y/N)
Full Screen XMIT (N) (Y/N) Recv Cmd Line in CDA (Y) (Y/N)
Terminal control codes (A) (EbcDic/Ascii)

Debug options: Log file level (N) (Min, All, Never)
Debugger () (Animator, Vcdebug)

--Serial---Y/N---Number-of-Threads--Delay--Idle--Max-Use---Server-----
: Re-usable(N) min(0) max(0) (0) (0) (0) () :

MSG-WAIT - exit, F2 - next, F3 - previous, F4 - update, F5 - security rec
23,79 30x80 Ready CP80 MSG OVR CAP NUM SCRL

```

Press MSG WAIT to return to the **smprog** screen where the show request was made.

Press F2 to display the next program definition.

Press F3 to display the previous program definition.

Press F4 to update the program definition. The screen will be redisplayed with the data fields padded with underscores.

Press F5 to display the security information (from TIP\$SEC) for the program definition. This will show the transaction name(s) used to execute the program.

## Change a Program Definition

**smprog** provides two ways to change (or update) a program definition.

1. From the **smprog** menu enter “CH” for the function to be performed (change) and the name of the program definition to be changed. (“AFT” in our example).
2. Use the list function to list program definitions and type “C” next to the program definition(s) that you want to change.

The following screen is displayed in response to a request to change a program definition.

```

Uw7test - TIP WorkStation
Session Edit View Tools Help
TF$SML4A TIP/ix Executable program definition 14:12 31 Mar 99
Update information for program AFT
Program Name: AFT Comments: Display Active File Table
Program Type: TIP Locap:
Label/Path: aft
Priority: Sets: TIPY
Open File(s):
Activation Area Sizes: CDA (256) MCS (512) WORK (256)
GDA (N) (Y/N)
Message editing: DICE (N) (Y/N) Upper case (N) (Y/N)
Edit (N) (Y/N/char) FCC Edit (_) (Y/N)
Full Screen XMIT (N) (Y/N) Recv Cmd Line in CDA (Y) (Y/N)
Terminal control codes (A) (EbcDic/Ascii)
Debug options: Log file level (N) (Min, All, Never)
Debugger (_) (Animator, Vcdebug)
--Serial---Y/N---Number-of-Threads--Delay--Idle--Max-Use---Server-----
: Re-usable(N) min(0) max(0) (0) (0) (0) ():
MSG-WAIT - cancel, F1 - redisplay

```

Press MSG WAIT to cancel the update and return to the **smprog** screen where the change request was made.

Fill in any desired changes and press XMIT to change the program definition. For a description of the fields on the program definition refer to the section on [“Adding a Program Definition”](#).

If the program definition is changed successfully and no security record exists for the program (in TIP\$SEC) then a screen will be displayed prompting for the security information. Refer to [smsec](#) for details about security records.

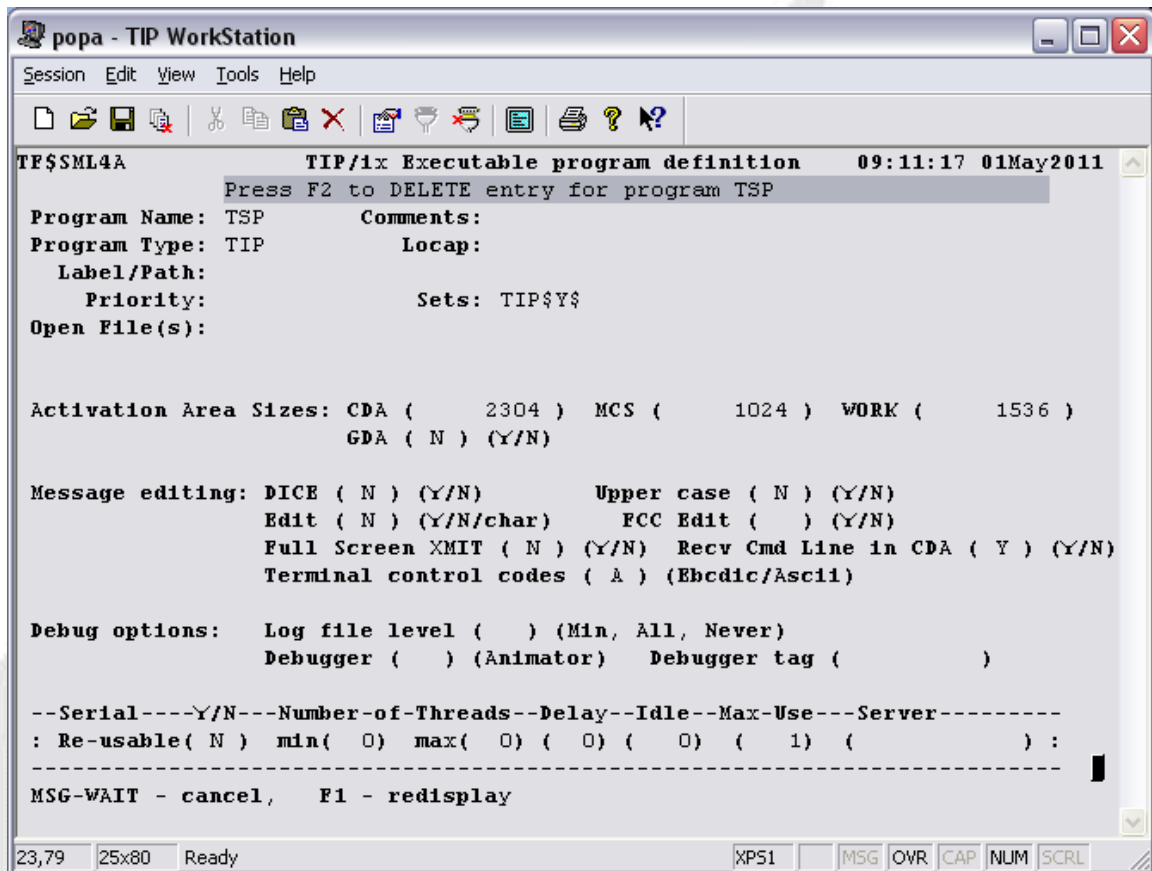
If the program name is changed then the original program definition is left unchanged and a new program definition is added for the new name.

## Delete a Program Definition

**smprog** provides two ways to delete a program definition.

1. From the **smprog** menu enter “DE” for the function to be performed (delete) and the name of the program definition to be deleted. (“AFT” in our example).
2. Use the list function to list program definitions and type “D” next to the program definition(s) that you want to delete.

The following screen is displayed in response to a request to delete a program definition.



```

popa - TIP WorkStation
Session Edit View Tools Help
TIP$SML4A TIP/ix Executable program definition 09:11:17 01May2011
Press F2 to DELETE entry for program TSP
Program Name: TSP Comments:
Program Type: TIP Locap:
Label/Path:
Priority: Sets: TIPY
Open File(s):

Activation Area Sizes: CDA (2304) MCS (1024) WORK (1536)
 GDA (N) (Y/N)

Message editing: DICE (N) (Y/N) Upper case (N) (Y/N)
Edit (N) (Y/N/char) FCC Edit () (Y/N)
Full Screen XMIT (N) (Y/N) Recv Cmd Line in CDA (Y) (Y/N)
Terminal control codes (A) (Ebcdic/Ascii)

Debug options: Log file level () (Min, All, Never)
 Debugger () (Animator) Debugger tag ()

--Serial---Y/N---Number-of-Threads--Delay--Idle--Max-Use---Server-----
: Re-usable(N) min(0) max(0) (0) (0) (1) () :

MSG-WAIT - cancel, F1 - redisplay
23,79 25x80 Ready XP51 MSG OVR CAP NUM SCRL

```

Press MSG WAIT to cancel the delete request and return to the **smprog** screen where the delete request was made.

Press F2 to confirm that this is the program definition that you want to delete.

If you press F2 and the program definition is deleted then if there are any security records for this program definition you will be given the

opportunity to delete them. If there is a single security record for the program it will be displayed and you are requested to press F2 to confirm that the security record is to be deleted. If there is more than one security record then they are displayed in a summary list (up to 12 will be shown) and a message indicating the number of security records that will be deleted is shown. Pressing F2 will delete all the security records (transaction names) that reference the program definition that was deleted.

## Get Security Records for Program Definitions

**smprog** provides two ways to retrieve security records for program definitions.

1. Use the list function to list program definitions and type "G" next to the program definition(s) for which you want to display the corresponding security information.
2. Display a program definition (using either the show command or pressing XMIT next to a program on the list display) and then press F5.

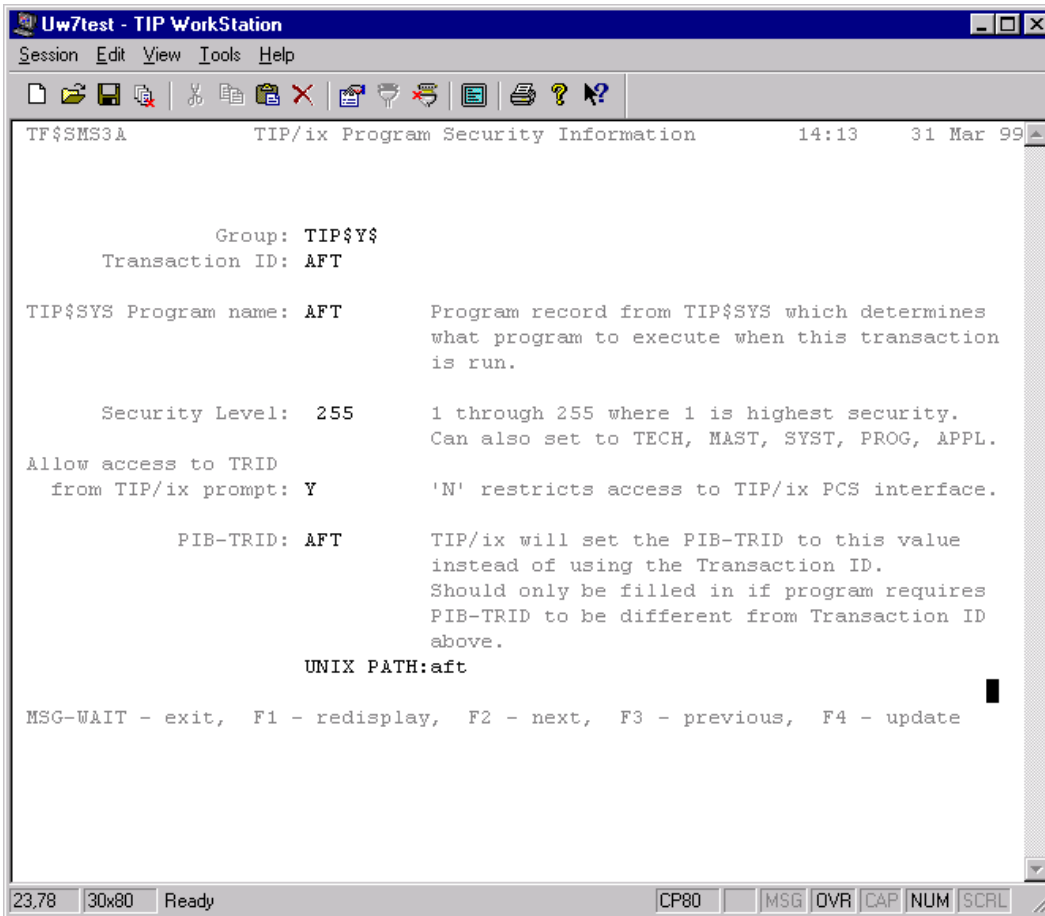
If there is only a single security record that references this program definition then that security record is displayed.

If there is more than one security record then a list of all security records referencing this program record will be displayed.

If no security record is defined then you will be asked if you wish to add one (via a Yes/No prompt).

For information about security records refer to the section on the [smsec](#) program.

The following screen is displayed if there is a single security record referencing this program definition. (This example shows a request for the security record for the program definition for AFT.)



```

Uw7test - TIP WorkStation
Session Edit View Tools Help
TF$SMS3A TIP/ix Program Security Information 14:13 31 Mar 99

 Group: TIPY
 Transaction ID: AFT

TIP$SYS Program name: AFT Program record from TIP$SYS which determines
 what program to execute when this transaction
 is run.

 Security Level: 255 1 through 255 where 1 is highest security.
 Can also set to TECH, MAST, SYST, PROG, APPL.

Allow access to TRID
from TIP/ix prompt: Y 'N' restricts access to TIP/ix PCS interface.

 PIB-TRID: AFT TIP/ix will set the PIB-TRID to this value
 instead of using the Transaction ID.
 Should only be filled in if program requires
 PIB-TRID to be different from Transaction ID
 above.

 UNIX PATH: aft

MSG-WAIT - exit, F1 - redisplay, F2 - next, F3 - previous, F4 - update

```

Press MSG WAIT to return to the program definition screen where the request was made for the security information.

Press F2 to display the next security record in the TIP\$SEC file (navigating by the 3rd key which is the associated TIP\$SYS program name). The subsequent security records may or may not relate to the program definition for which security information was originally requested.

Press F3 to display the previous security record in the TIP\$SEC file (navigating by the 3rd key which is the associated TIP\$SYS program name). The subsequent security records may or may not relate to the program definition for which security information was originally requested.

Press F4 to update the security record. The screen will be redisplayed with the data fields padded with underscores.

The following screen is displayed in response to a request for security information if there are multiple security records referencing the program definition. (This example shows a request for the security record for the program definition for CALENDAR.)



```

TF$SMSBA TIP/ix Security Maintenance Utility 14:13 31 Mar 99
=====
End of List
Command Type TIP$SYS TIP$SEC Security TIP$SYS
 Name Name Level Unix Path
- - - - - - - - -
PROG CALENDAR CAL TIPY 255 $TIPROOT/bin/calendar
- - - - - - - -
 CALENDAR TIPY 255 $TIPROOT/bin/calendar

<-- Enter 'D' to delete, 'C' to change, 'S' to show
 Transmit to process command(s)

MSG-WAIT - return to menu, F1 - redisplay, F2 - next, F3 - previous

```

Press MSG WAIT to return to the program definition screen where the request was made for the security information.

Press F2 to display the next screen of security records for the TIP\$SYS program name

Press F3 to display the previous screen of security records for the TIP\$SYS program name.

You can enter single character commands next to the security records and when you XMIT the commands will be processed sequentially (one at a time) from the top of the display. The commands that can be entered are:

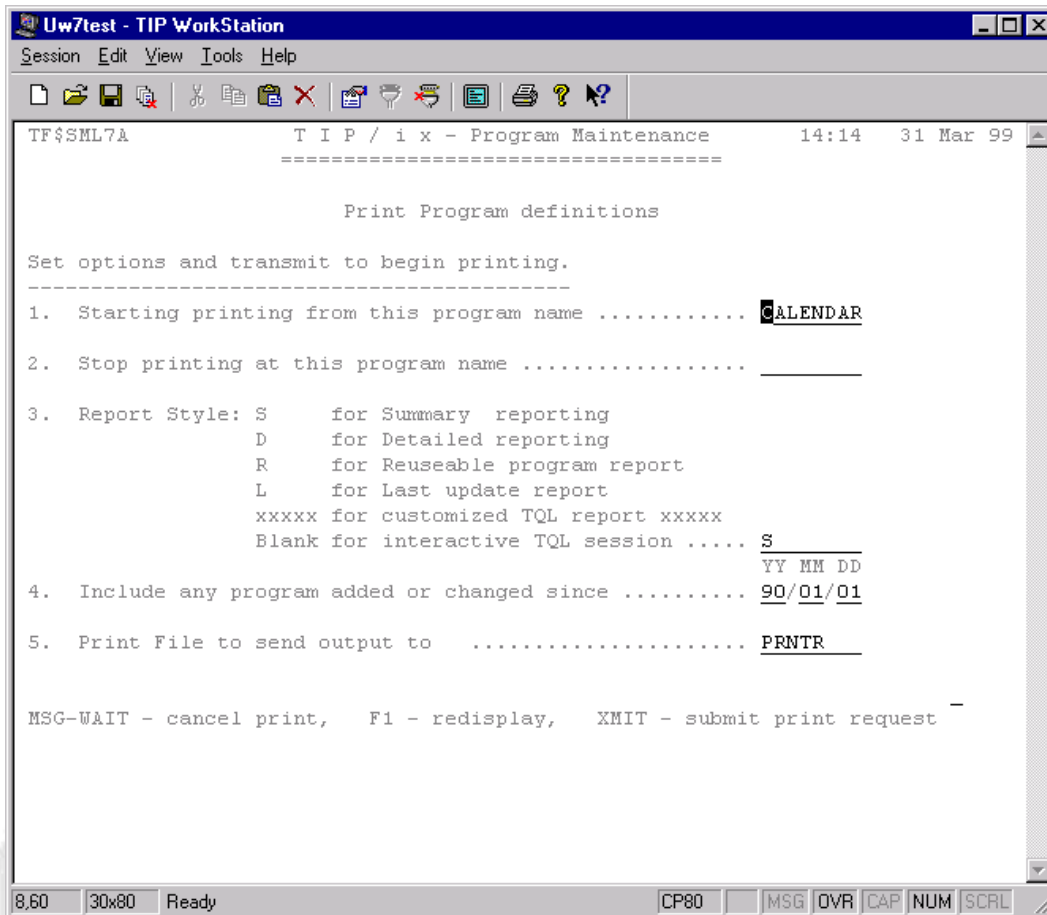
- C** Change security entry
- D** Delete security entry
- S** Show security entry

If no commands are entered and you press XMIT next to a security entry then this is interpreted as a show (S) request for that security entry.

## Print Program Definitions

To print program definitions enter “PR” as the function to be performed (print) on the **smprog** menu display. A program name can also be supplied to be used as the point to start printing program definitions from. After pressing **XMIT**, you will be prompted to fill in the print options and report style.

When you select the Print function from **smprog** the following screen is displayed. Enter the desired print options and press transmit to print TIP/ix program definition information.



```

Uw7test - TIP WorkStation
Session Edit View Tools Help
TF$SML7A T I P / i x - Program Maintenance 14:14 31 Mar 99

 Print Program definitions

Set options and transmit to begin printing.

1. Starting printing from this program name CALENDAR
2. Stop printing at this program name _____
3. Report Style: S for Summary reporting
 D for Detailed reporting
 R for Reuseable program report
 L for Last update report
 xxxxx for customized TQL report xxxxx
 Blank for interactive TQL session S
 YY MM DD
4. Include any program added or changed since 90/01/01
5. Print File to send output to PRNTR

MSG-WAIT - cancel print, F1 - redisplay, XMIT - submit print request

```

The following is a description of the fields in the preceding screen.

### Start at program name

### Stop at program name

Range of program definitions to include in report. For example if set to “K” and “R” respectively then all program names in the range K to R will be included in the report.

### Report style

Fill in one of the valid report styles or leave blank.

- S** Summary report that prints a single line of information about each program.
- D** Detailed report that includes all information in the program definition as well as all security records for the program definitions.
- R** Report of all programs defined to execute as reusable.
- L** Report date, time, and user id performing last update to the program definitions.

**<report name>**

Any valid report name in the TQL program TIPSYSX. This allows you to modify the TQL program and select your own reports. Fill in LASTUPDT for a TQL report showing the user id, date, and time of the last update for each program definition.

**<blank>**

Leave this entry blank to initiate an interactive TQL session with the program TIPSYSX.

**Change Date**

Only report program definitions that have changed since this date.

**Print File**

Print file to route report to. This must be a valid TIPPRINT destination. Print files are defined by smprint and of course aliases can be set up using smsec. AUX0 can be used to direct report output to the terminal.

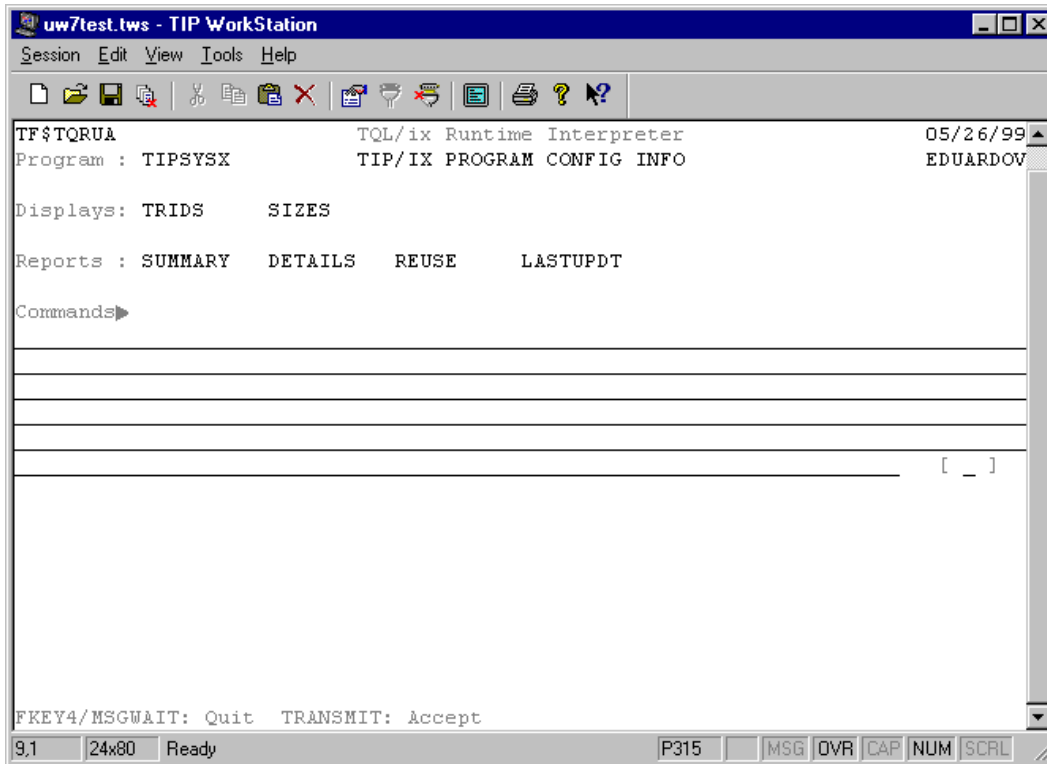
The print function is implemented by the supplied TQL program TIPSYSX. If this program is not compiled and available then the print function will not work.

## Browse Program Definitions with TQL

The **TIPSYSX** TQL program enables you to interrogate program definitions with ad hoc TQL queries. This TQL program can be run either:

- Directly from the TIP/ix command prompt by entering "TQL TIPSYSX"
- From the **smprog** menu enter "BR" for the function to be performed (browse).

The following screen is displayed when you run **TIPSYSX**.



At this point you can make any valid TQL request including:

- Request a predefined display.
- Request a predefined report.
- Make an ad hoc request to list (or print) fields from the program definitions.

See the TIP/ix TQL Reference for details on interacting with TQL.

The predefined displays include:

### TRIDS

Display all Transaction names (trids) that are associated with a program definition (via the smsec utility). The display consists of a few fields from the program definition, followed by a list of all security records that refer to it.

### SIZES

Display 16 program definitions at a time with a single line per definition showing the sizes defined for the various LINKAGE SECTION areas.

The predefined reports include:

### SUMMARY

Each program definition is summarized into a single line on the report.

### DETAILS

For each program definition the report includes all fields

from the program definition and any security records referencing it.

### REUSE

A report listing only program definitions with the “Serial Re-usable” attribute set to “Y”. The report consists of attributes related to serially re-usable program execution.

### LASTUPDT

A report showing who last updated program definitions (and when).

In addition to using the predefined displays and reports it is possible to perform ad hoc queries based on field settings in program definitions. The following is a list of program definition attributes and the corresponding field names:

| Program Definition Attribute | TQL Field Names  |
|------------------------------|------------------|
| Program Name                 | LM-NAME          |
| Comments                     | LM-CMT           |
| Program Type                 | PROG-TYPE        |
| LOCAP                        | LM-LOCAP         |
| Label/Path                   | LM-LBL           |
| Priority                     | LM-PRI           |
| Sets (1st one)               | LM-SET (1)       |
| Sets (2nd one)               | LM-SET (2)       |
| Files                        | LM-FILE (1 - 16) |
| CDA                          | LM-CDA           |
| MCS                          | LM-MCS           |
| WORK                         | LM-WORK          |
| GDA                          | LM-GDA           |
| IMA                          | LM-IMA           |
| OMA                          | LM-OMA           |
| CALL BUILD XMIT              | LM-DYNOMA        |
| DICE                         | LM-DICE          |
| Upper case                   | LM-UPR           |
| Edit                         | LM-EDIT          |
|                              | LM-EDCHR *       |
| FCC Edit                     | LM-FCCEDIT       |
| Full Screen XMIT             | LM-FSXMIT        |
| Recv Cmd Line in CDA         | LM-CML           |

|                        |                |
|------------------------|----------------|
| Terminal control codes | LM-CHARSET     |
| Log file level         | LM-TRANLOG     |
| Debugger               | LM-SYMDBG      |
| Serial Re-usable       | LM-REUSE       |
| Min Threads            | LM-MIN-THREADS |
| Max Threads            | LM-MAX-THREADS |
| Delay                  | LM-MAX-DELAY   |
| Idle                   | LM-MAX-IDLE    |
| Max use                | LM-MAX-USE     |
| Server                 | LM-SERVER      |

\* If "Edit" is set to a character then LM-EDIT = "Y" and LM-EDCHR will be set to the "edit character".

### Examples:

List Programs sorted in descending order by their WORK size.

```
SIZES SORT BY LM-WORK:D
```

List programs that are defined as "Serial Re-usable"

```
SUMMARY IF LM-REUSE = "Y"
```

or just use the predefined "REUSE" report

```
REUSE
```

List all the transaction codes referencing the program definition for "FOPEN"

```
TRIDS FOPEN
```

List TIP program definitions with MCS sizes that are at least 1500 sorted in descending order by MCS size

```
LIST (LM-NAME PROG-TYPE LM-MCS LM-LBL)
IF PROG-TYPE = "TIP" AND LM-MCS > 1500 SORT
BY LM-MCS:D
```

Print program details report on auxiliary printer for program definitions with names in the range PUR to PZZ.

```
DETAILS FROM "PUR" TO "PZZ" ON AUX1
```

List all programs defined with "CALL BUILD XMIT" set to "N".

```
LIST (LM-NAME LM-DYNOMA) IF LM-DYNOMA = "N"
```

## smsec - Define System Security

Using the **smsec** program, you may define system security records for programs, files, and queues. This information is kept in the TIP/ix security file, TIP\$SEC, and is used by TIP/ix to control access to the files, programs, and queues defined on the system. TIP\$SEC only contains security information related to these items. Configuration information for these items is kept in the TIP/ix system file, TIP\$SYS.

Refer to the other system maintenance (sm) utilities in this manual for a description of how to establish and maintain configuration information for [programs](#), [files](#), [queues](#), [LOCAPs](#), [users](#), and [group sets](#).

There are two dimensions to security in TIP/ix, groups and security levels. Users are granted access to groups via [smuser](#) and [smgrpset](#). Programs, files, and queues are associated with groups by security entries created with **smsec**.

Each record in the security file (TIP\$SEC) indicates the item type, group that can access the item, the logical name of the item, the security level required to access the item, and the name of the item definition in the TIP\$SYS file associated with this security entry.

This implementation means that there may be multiple security definitions associated with a single definition in the TIP\$SYS file. For example a single program definition in TIP\$SYS could be referenced by different transaction codes defined via **smsec**. Alternatively, a transaction code could be defined multiple times, once for each group that has access to it.

A TIP/ix user may only access programs, files, and queues that are defined in groups in which the user is a member. Furthermore, even though the user is a member of a group his/her access to programs, files, and queues in that group is restricted by the security level assigned to the user.

When a TIP/ix user attempts to run a program or access a file or queue, TIP/ix looks up the security definition for the item in the TIP\$SEC file. This look up follows a fixed order, known as the “*standard order of search*”. First TIP/ix looks for a definition in the user’s private group (group name equal to the user id), then in any of the user’s active groups and finally in the universal group, TIP\$Y\$. This order is important because TIP/ix only considers access to the first matching entry. If the user’s security level does not permit access to the first matching entry then TIP/ix will not permit the user access to the item.

The initial list of active groups for a user consists of the user’s elective groups followed by any groups in the user’s logon set. The number of active groups is limited to 16. However, the active groups can be altered (via the **groups** transaction) to any group that the user has been granted access; either as an elective group or as a member of a logon set or group set.

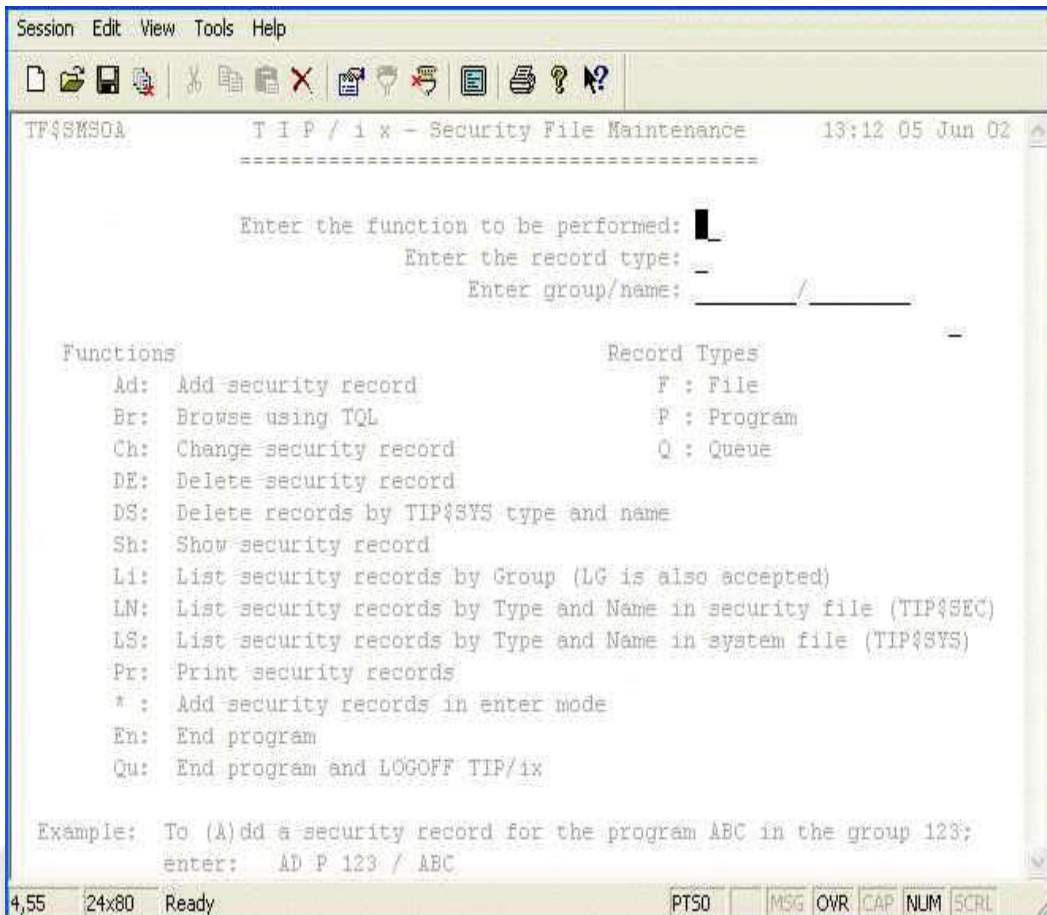
**smsec** must be run from within the TIP/ix shell (tipix).

**Syntax:**

**smsec [function] [type] [group] [itemname]**

If a valid function is supplied then **smsec** will perform that function and then terminate when that function has been completed. Most functions require an item type, group name and itemname.

If no function is supplied then **smsec** displays the following security file maintenance menu screen.



```

Session Edit View Tools Help

TIP$SMSSOA T I P / i x - Security File Maintenance 13:12 05 Jun 02

Enter the function to be performed: L
Enter the record type: -
Enter group/name: _____

Functions
Ad: Add security record
Br: Browse using TQL
Ch: Change security record
DE: Delete security record
DS: Delete records by TIP$SYS type and name
Sh: Show security record
Ll: List security records by Group (LG is also accepted)
LN: List security records by Type and Name in security file (TIP$SEC)
LS: List security records by Type and Name in system file (TIP$SYS)
Pr: Print security records
* : Add security records in enter mode
En: End program
Qu: End program and LOGOFF TIP/ix

Record Types
F : File
P : Program
Q : Queue

Example: To (A)dd a security record for the program ABC in the group 123;
enter: AD P 123 / ABC

```

From this menu you may perform all the traditional maintenance functions on security records. You may add new security records; change, delete, show or print existing security records and list a summary of all security records.

In the preceding screen, the function to be performed can be abbreviated to the portion shown in upper case. For example the function *add* can be entered as “A”.

Security records are identified by a record type, group name and item name:



### Record Type

- F security record for a TIP/ix file or printer (see smfile)
- P security record for a TIP/ix program (see smprog)
- Q security record for a TIP/ix queue (see smqueue)

### Group Name

Group that can access this item

### Item Name

Logical name used from TIP/ix.

For a program this is the transaction code (command used to execute the program).

For a file this is the name supplied by a program to access the file via TIPFCS.

For a queue this is the name supplied by a program when accessing the queue via TIPQUEUE.

## Add a Security Record

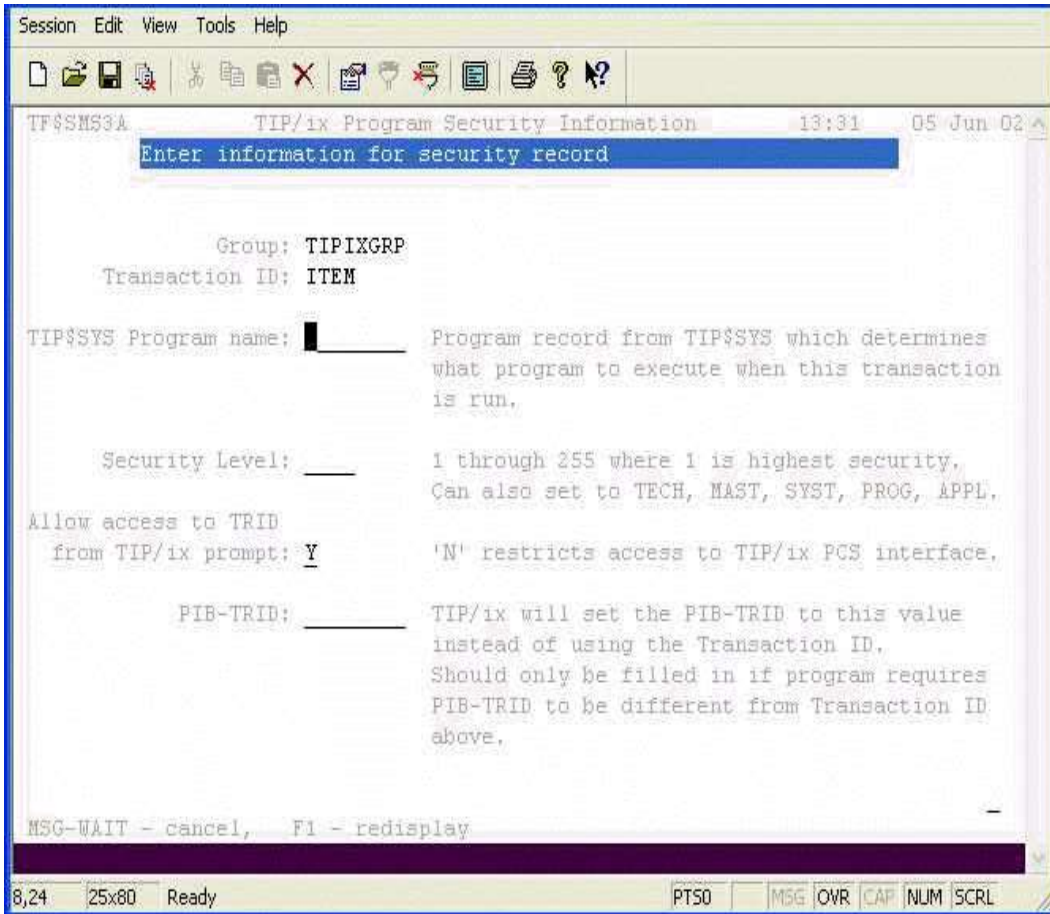
To define a new security record, enter "A" for the function to be performed (Add), a valid record type (F, P, or Q), the group being granted access to the item, and the item name.

After pressing XMIT a screen will be displayed to allow you to enter the security information applicable to the record type you have entered.

### Add a Security Record for a Program:

To add a security record for a program enter "A" for the function to be performed (Add); "P" for the record type; and fill in the desired group name and program (transaction) name. If the group name is left blank then it will default to TIP\$Y\$.

After pressing XMIT, the following screen is displayed.



```

Session Edit View Tools Help

TF$SMS3A TIP/ix Program Security Information 13:31 05 Jun 02
Enter information for security record

Group: TIPIXGRP
Transaction ID: ITEM

TIP$SYS Program name: _____ Program record from TIP$SYS which determines
what program to execute when this transaction
is run.

Security Level: _____ 1 through 255 where 1 is highest security.
Can also set to TECH, MAST, SYST, PROG, APPL.

Allow access to TRID
from TIP/ix prompt: Y 'N' restricts access to TIP/ix PCS interface.

PIB-TRID: _____ TIP/ix will set the PIB-TRID to this value
instead of using the Transaction ID.
Should only be filled in if program requires
PIB-TRID to be different from Transaction ID
above.

MSG-WAIT - cancel, F1 - redisplay

8,24 25x80 Ready PTSO MSG OVR CAP NUM SCRL

```

At the bottom of the screen is a summary of function keys and their actions.

At any time you may enter MSG WAIT to cancel the function and return to the main menu screen.

Pressing F1 causes the screen to be re-displayed (Caution! any data entered on the screen will be lost).

Fill in the information for the security record definition and press XMIT to add the record to the TIP\$SEC file. In the preceding example a security entry would be created for the transaction “TESTORD” in the group “TEST” and it would reference the program definition “TESTORD” from the TIP\$SYS file.

If the security record is added successfully and no configuration information exists for the program in the TIP\$SYS file then you will be asked if you would like to add that information at this time. Responding with a “Y” to this question will result in [smprog](#) being invoked to add a program definition. Refer to [smprog](#) for details about program records.

The following is a description of each field in the preceding screen.

### Group

The name of the group that this security record is

associated with. Users with access to this group are able to access this program (if they have the required security level). A program must have an entry for each group that is to have access to it. This implementation allows an item to have different security levels for each group having access to the item.

**Transaction ID**

Name entered by the TIP/ix user or value entered in PIB-TRID or SUCCESSOR-ID to run the program defined in the TIP\$SYS program entry below.

**TIP\$SYS Program Name**

The TIP\$SYS program definition which will determine what program to run when the above Transaction ID is requested from this group. The program may be defined to be TIP, IMS, or UNIX. See smprog for a description of TIP\$SYS program definitions.

Default is to use the Transaction ID as the TIP\$SYS Program Name.

**Security Level**

This is a number between 1 and 255 which represents the level of security a user must have (within this application group) to run the program that is defined by this security record. The highest level is 1 and the lowest is 255. If the user's security value is numerically greater than this number then access will be denied.

The following is a list of names associated with certain levels which may be used instead of using a number.

TECH 1  
MAST 9  
SYST 19  
PROG 29  
APPL 32

**Allow access to TRID from TIP/ix prompt**

Used to restrict command line access to transaction code.

- Y** Transaction can be run from the TIP/ix prompt as well as from a program
- N** Transaction can only be run from a program (via PIB-TRID or SUCCESSOR-ID).

Transaction can not be run from the TIP/ix command prompt.

**PIB-TRID**

TIP/ix will set the PIB-TRID to this value when the transaction is executed. The default value is the Transaction ID above. This field could be used to aid in

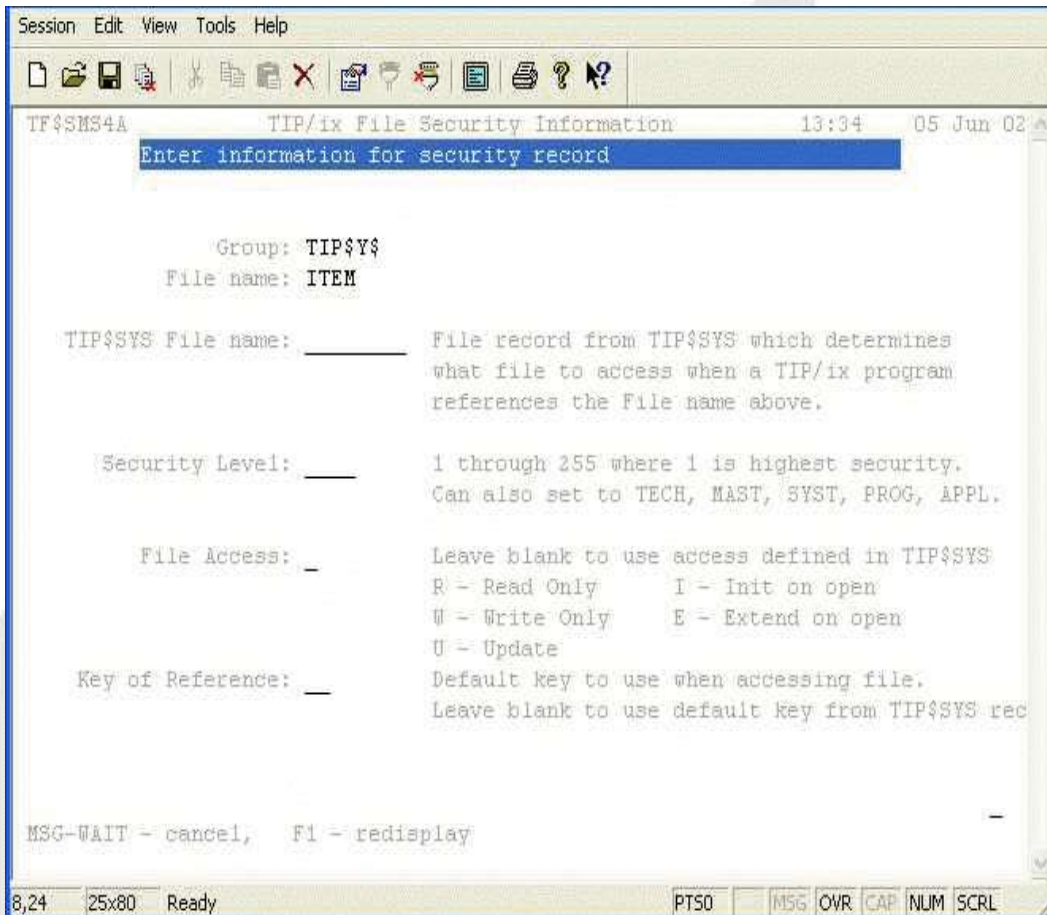
localization of an application. The Transaction ID could be some local language value but the PIB-TRID could be set to the language value understood by the program.

For example an American programmer may have written a program that takes certain actions when the PIB-TRID is "PRINT". However, at a German site it is desired to set the Transaction ID to "DRUCKEN" and so this field is set to "PRINT" and the program will not be aware that the user keyed in "DRUCKEN" instead of "PRINT".

### Add a Security Record for a File

If you want to enter new security information for a file, you would enter "A" for the function to be performed (Add); "F" for the record type; and enter the desired group name and file name. If the group name is left blank then it will default to TIP\$Y\$.

After pressing **XMIT**, the following screen is displayed.



```

Session Edit View Tools Help
TF$SMS4A TIP/ix File Security Information 13:34 05 Jun 02
Enter information for security record

Group: TIPY
File name: ITEM

TIP$SYS File name: _____ File record from TIP$SYS which determines
what file to access when a TIP/ix program
references the File name above.

Security Level: _____ 1 through 255 where 1 is highest security.
Can also set to TECH, MAST, SYST, PROG, APPL.

File Access: _____ Leave blank to use access defined in TIP$SYS
R - Read Only I - Init on open
W - Write Only E - Extend on open
U - Update

Key of Reference: _____ Default Key to use when accessing file.
Leave blank to use default Key from TIP$SYS rec

MSG-WAIT - cancel, F1 - redisplay
8,24 25x80 Ready PTSO MSG OVR CAP NUM SCRL

```

At the bottom of the screen is a summary of function keys and their actions.

At any time you may enter MSG WAIT to cancel the function and return to the main menu screen.

Pressing F1 causes the screen to be re-displayed (Caution! any data entered on the screen will be lost).

Fill in the information for the security record definition and press XMIT to add the record to the TIP\$SEC file. In the preceding example a security entry for the file "TESTFIL" would be created in the group "TEST" and it would reference the file definition "TESTFIL" from the TIP\$SYS file.

If the security record is added successfully and no configuration information exists for the file in the TIP\$SYS file then you will be asked if you would like to add that information at this time. Responding with a "Y" to this question will result in [smfile](#) being invoked to add a file definition. Refer to [smfile](#) for details about file definitions and [smprint](#) for details about printer file definitions.

The following is a description of each field in the preceding screen.

### Group

The name of the group that this security record is associated with. Users in this group are able to access this file (if they have the required security level). A file must have an entry for each group that is to have access to it. This implementation allows an item to have different security levels for each group having access to the item.

### File Name

Name used from TIPFCS file interface to access the file. When this file name is accessed from TIP/ix in this group it will use the "TIP\$SYS File name" below to determine what UNIX file to access.

TIP/ix 8 characters max  
IMS 7 characters max  
IMS/1100  
8 characters max (not 12 as on IMS/1100)

### TIP\$SYS File Name

The TIP\$SYS file definition which will determine what file to access when the above File Name is requested from this group. See [smfile](#) for a description of TIP\$SYS file definitions. See [smprint](#) for a description of TIP\$SYS printer file definitions.

Default is to use the TIPFCS file name (previous field) as the TIP\$SYS File Name.

### Security Level

This is a number between 1 and 255 which represents the level of security a user must have (within this application group) to access the file that is defined by this security record. The highest level is 1 and the lowest is 255. If the user's security value is numerically greater than this number then access will be denied.

The following is a list of names associated with certain levels which may be used instead of using a number.

```
TECH 1
MAST 9
SYST 19
PROG 29
APPL 32
```

### File Access

This field can be set to override the file access set for this file in the TIP\$SYS record above.

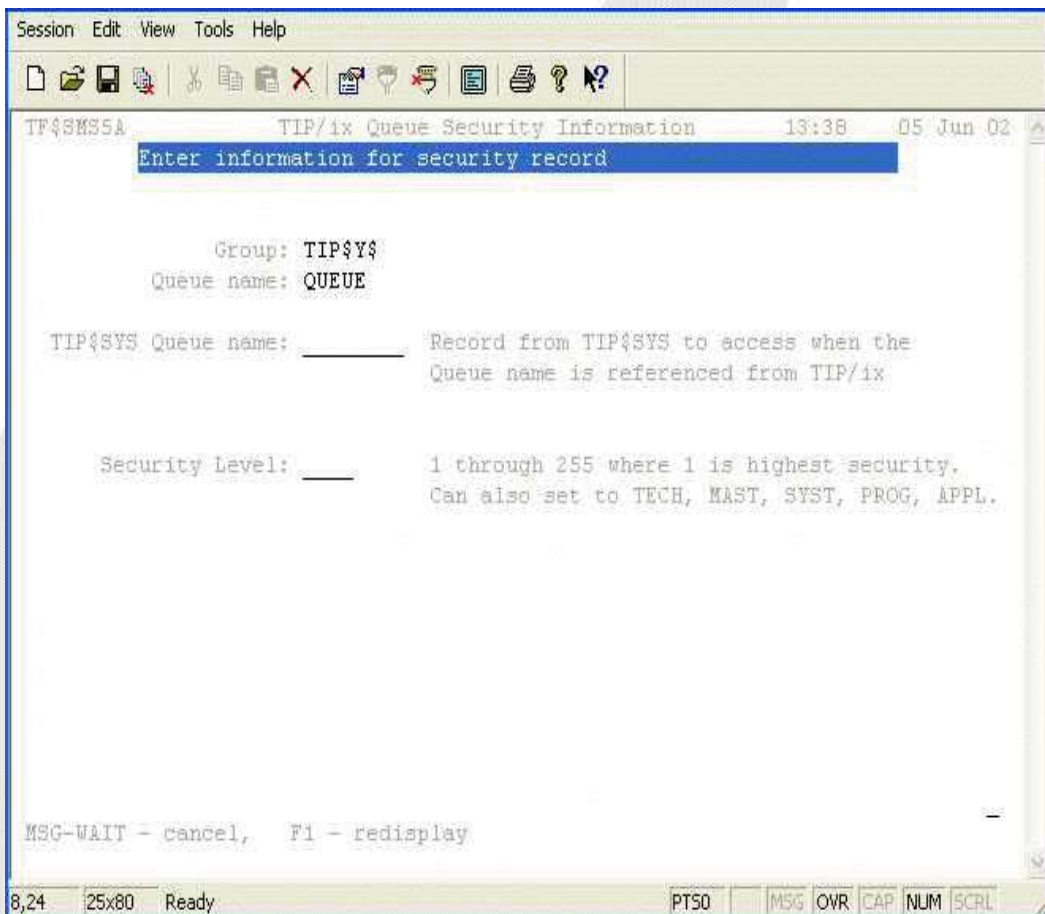
### Key of Reference

This field can be used to override the default key of reference set for this file in the TIP\$SYS record above.

## Add a Security Record for a Queue

If you want to enter new security information for a queue, you would enter "A" for the function to be performed (Add); "Q" for the record type; and enter the desired group name and queue name. If the group name is left blank then it will default to TIP\$Y\$.

After pressing **XMIT**, the following screen is displayed.



At the bottom of the screen is a summary of function keys and their actions.

At any time you may enter MSG WAIT to cancel the function and return to the main menu screen.

Pressing F1 causes the screen to be re-displayed (Caution! any data entered on the screen will be lost).

Fill in the information for the security record definition and press XMIT to add the record to the TIP\$SEC file. In the preceding example a security entry would be created for the queue "ORDERQ" in the group "TIP\$\$" and it would reference the queue definition "ORDERQ" from the TIP\$SYS file.

If the security record is added successfully and no configuration information exists for the queue in the TIP\$SYS file then you will be asked if you would like to add that information at this time. Responding with a "Y" to this question will result in [smqueue](#) being invoked to add a queue definition. Refer to [smqueue](#) for details about queue records.

The following is a description of each field in the preceding screen.

**Group**

The name of the group that this security record is associated with. Users in this group are able to access this queue (if they have the required security level). A queue must have an entry for each group that is to have access to it. This implementation allows an item to have different security levels for each group having access to the item.

**Queue Name**

Name used from TIPQUEUE interface. When this queue name is accessed from TIP/ix in this group it will use the TIP\$SYS queue name below to determine what queue definition to use.

**TIP\$SYS Queue Name**

The TIP\$SYS queue definition to use when the above Queue Name is requested from this group. See [smqueue](#) for a description of TIP\$SYS queue definitions.

Default is to use the queue name (previous field) as the TIP\$SYS Queue Name.

**Security Level**

This is a number between 1 and 255 which represents the level of security a user must have (within this application group) to access the queue that is defined by this security record. The highest level is 1 and the lowest is 255. If the user's security value is numerically greater than this number then access will be denied.

The following is a list of names associated with certain levels which may be used instead of using a number.

```
TECH 1
MAST 9
SYST 19
PROG 29
APPL 32
```

## List Security Records

### List Security Records by Group

From the **smsec** menu enter “LI” (or “LG”) for the function to be performed (List) and a group to use as the starting point for the list. This does not have to be a complete group name. For example a group value of “B” could be used to start listing security records with a group name greater than (or equal to) “B”.

If a record type is supplied then the list will only contain records of the requested type.

If a complete group name and record type is supplied then an item name (partial or complete) can be supplied as the starting point for listing items of the record type supplied.

The list function displays a single line summary for each security record showing the record type (PROG, FILE, or QUEUE), group name, TIP\$SEC name, TIP\$SYS name, security level, and TIP\$SYS UNIX path from the security record.

The data displayed in the column titled “TIP\$SYS Unix path” on the list display varies with the record type and the item definition. For remote files and programs the LOCAP is displayed instead of the UNIX path. For print files (see [smprint](#)) instead of a UNIX path it will indicate the action (PIPE, FORK, or WRITE) associated with the print file definition and show either the print command or print directory. For queues the name of the queue service program is displayed.

When the UNIX path name is too long to fit on the list display it will be truncated so that the trailing portion (suffix) is displayed. If path name is an absolute path it will be truncated to `/.../<suffix>`. If the path name is a relative path it will be truncated to `.../<suffix>`.

The following screen is displayed in response to the list command. In this example the group name was specified as “TEST” and the record type was left blank.



```

Session Edit View Tools Help

TF$SMS2A TIP/ix Security Maintenance Utility 13:41 05 Jun 02

Command Type Group TIP$SEC TIP$SYS Security TIP$SYS

- PROG ARC ARC1 CONNECT 255 connect
- PROG ARC CAT TCM 29 tcm
- PROG ARC TTTT WMI 255 wmi
- PROG ARC1 WWWW WMI 255 wmi

- FILE TIPY ALLANPRT ALLANPRT 255 /tmp
- FILE TIPY ARBCTRL ARBCTRL 255 /u/asad/8654/ARBCTRLF
- FILE TIPY B1447 B1447 255 /u/allan/7891/B1447
- FILE TIPY BLUE1 BLUE1 255 BLUE1

- FILE TIPY BLUE2 BLUE2 255 BLUE2
- FILE TIPY BSPFIL BSPFIL 255 /u/javed/8636/bspfil
- FILE TIPY BUDGET BUDGET 255 /u/asad/8863/budget
- FILE TIPY COMMEXP COMMEXP 255 /u/allan/8002/COMMDAT

<-- Enter 'D' to delete, 'C' to change, 'S' to show, 'G' to get TIP$SYS rec
Transmit to process command(s)

MSG-WAIT - return to menu, F1 - redisplay, F2 - next, F3 - previous

6,5 25x80 Ready PTSO MSG OVR CAP NUM SCRL

```

Press MSG WAIT to return to the main menu screen.

Press F2 to display the next screen of security records.

Press F3 to display the previous screen of security records.

You can enter single character commands next to the security entries and when you press XMIT the commands will be processed sequentially (one at a time) from the top of the display. The commands that can be entered are:

- C** Change security record
- D** Delete security record
- S** Show security record
- G** Get the TIP\$SYS record for this security record.

If no commands are entered and you press XMIT next to a security record on the list display then this is interpreted as a show (S) request for this security record.

## List Security Records by Type and Item Name in Security File

This command provides a convenient method for finding all application groups that have access to a program, file, or queue. Each line displayed will show a program, file, or queue and an application group that has access to it.

From the **smsec** menu enter “LN” for the function to be performed (List by Name) and enter the desired record type and the name (partial or complete) to start listing from. Since group name is not used by this function the item name can be filled in either the group name or item name position on the main menu screen for **smsec**. For example filling in a record type of “P” and an item name of “M” would list security records for programs with transaction names greater than (or equal to) “M”.

The record type must be supplied with this command and so only a single type of security record can be listed when listing by the name in the security file. For programs this is the name used to run a program. For files this is the name used to access the file via TIPFCS. For queues this is the name used to reference the queue via TIPQUEUE.

The list function displays a single line summary for each security record showing the record type (PROG, FILE, or QUEUE), TIP\$SEC name, TIP\$SYS name, group name, security level, and TIP\$SYS UNIX path from the security record.

The data displayed in the column titled “TIP\$SYS Unix path” on the list display varies with the record type and the item definition. For remote files and programs the LOCAP is displayed instead of the UNIX path. For print files (see [smprint](#)) instead of a UNIX path it will indicate the action (PIPE, FORK, or WRITE) associated with the print file definition and show either the print command or print directory. For queues the name of the queue service program is displayed.

When the UNIX path name is too long to fit on the list display it will be truncated so that the trailing portion (suffix) is displayed. If path name is an absolute path it will be truncated to */.../<suffix>*. If the path name is a relative path it will be truncated to *.../<suffix>*.

The following screen is displayed in response to the “LN” command. In this example the record type was specified as “P” and the item name was specified as “M”.

```

Session Edit View Tools Help

TIP/ix Security Maintenance Utility 13:42 05 Jun 02

Command Type TIP$SEC TIP$SYS Security TIP$SYS
Name Name Group Level Unix Path

| PROG MAIL MAIL TIP$$ 255 xt-m11
| PROG MAILEDT MAILEDT TIP$$ 255 fse
| PROG MAILLIST MAILLIST TIP$$ 255 xt-m19
| PROG MEM NOTIMPL TIP$$ 255 notimpl
|
| PROG MENU MENU TIP$$ 255 $TIPROOT/bin/menu
| PROG MENU$$ MENU TIP$$ 255 $TIPROOT/bin/menu
| PROG MENU.SH MENU.SH TIP$$ 255 ../utils/menu.sh
| PROG MENU0001 MENU0001 TIP$$ 255 imsmnu
|
| PROG MENUAR MENUAR TIP$$ 255 menuar
| PROG MENUDEF MENUDEF TIP$$ 255 menudef
| PROG MFM MFM TIP$$ 29 ttxfm
| PROG MSG XTNOTE TIP$$ 255 xtnote
|
|<-- Enter 'D' to delete, 'C' to change, 'S' to show, 'G' to get TIP$SYS rec
|Transmit to process command(s)
|
MSG-WAIT - return to menu, F1 - redisplay, F2 - next, F3 - previous

6,5 25x80 Ready PT50 MSG OVR CAP NUM SCRL

```

Press MSG WAIT to return to the main menu screen.

Press F2 to display the next screen of security records.

Press F3 to display the previous screen of security records.

You can enter single character commands next to the security entries and when you press XMIT the commands will be processed sequentially (one at a time) from the top of the display. The commands that can be entered are:

- C** Change security record
- D** Delete security record
- S** Show security record
- G** Get the TIP\$SYS record for this security record.

If no commands are entered and you press XMIT next to a security record on the list display then this is interpreted as a show (S) request for this security record.

### List Security Records by Type and Associated System File Record

This command provides a convenient method for discovering aliases for programs, files, and queues. For example, the program definition for

WHOSON in the TIP\$SYS file is used to process requests to run the [DIE](#), [PURGE](#), and [WHOSON](#) transactions. These are 3 aliases (entries in security file) that refer to a single program definition in the system file (TIP\$SYS).

From the **smsec** menu enter “LS” for the function to be performed (List by definition name in system file) and enter the desired record type and the name (partial or complete) to start listing from. Since group name is not used by this function the item name can be filled in either the group name or item name position on the main menu screen for **smsec**. For example filling in a record type of “P” and an item name of “W” would list security records that reference program definitions (from TIP\$SYS) with names greater than (or equal to) “W”.

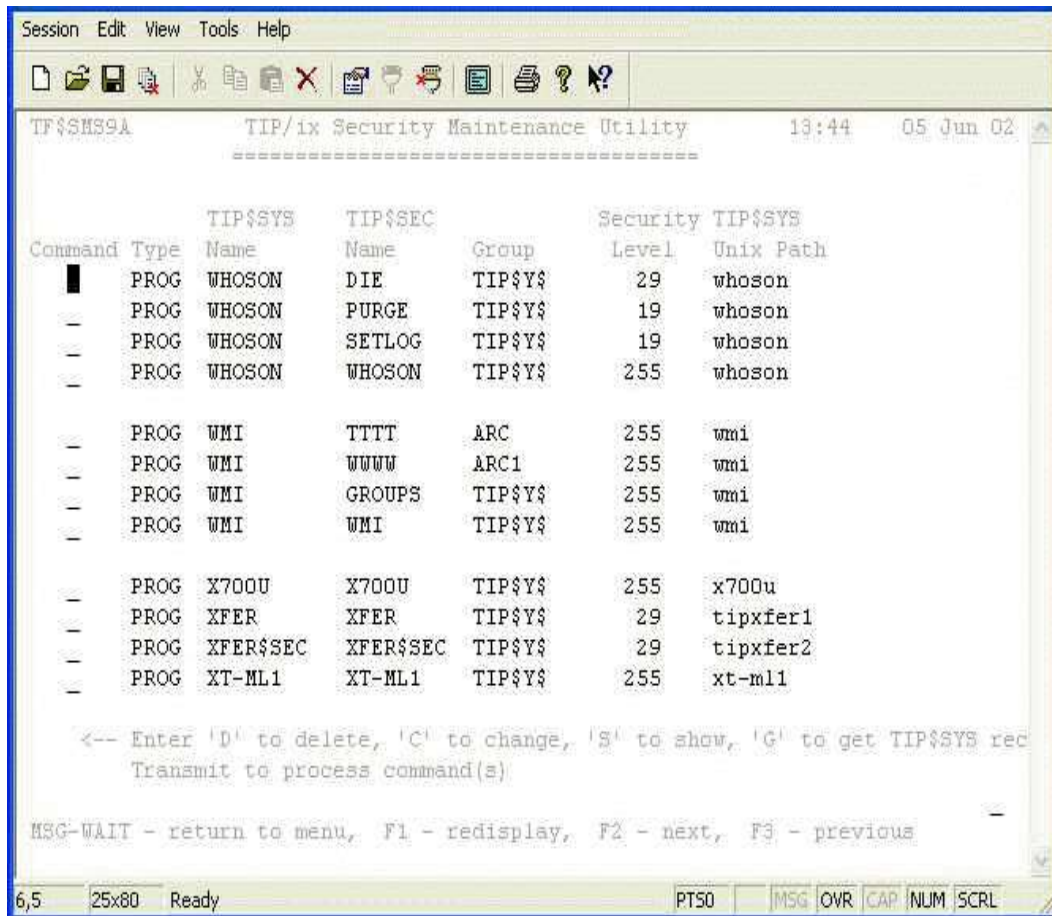
The record type must be supplied with this command and so only a single type of security record can be listed when listing by the name of definitions in the system file.

The list function displays a single line summary for each security record showing the record type (PROG, FILE, or QUEUE), TIP\$SYS name, TIP\$SEC name, group name, security level, and TIP\$SYS UNIX path from the security record.

The data displayed in the column titled “TIP\$SYS Unix path” on the list display varies with the record type and the item definition. For remote files and programs the LOCAP is displayed instead of the UNIX path. For print files (see **smprint**) instead of a UNIX path it will indicate the action (PIPE, FORK, or WRITE) associated with the print file definition and show either the print command or print directory. For queues the name of the queue service program is displayed.

When the UNIX path name is too long to fit on the list display it will be truncated so that the trailing portion (suffix) is displayed. If path name is an absolute path it will be truncated to */.../<suffix>*. If the path name is a relative path it will be truncated to *.../<suffix>*.

The following screen is displayed in response to the “LS” command. In this example the record type was specified as “P” and the item name was specified as “W”.



Press MSG WAIT to return to the main menu screen.

Press F2 to display the next screen of security records.

Press F3 to display the previous screen of security records.

You can enter single character commands next to the security entries and when you press XMIT the commands will be processed sequentially (one at a time) from the top of the display. The commands that can be entered are:

- C** Change security record
- D** Delete security record
- S** Show security record
- G** Get the TIP\$SYS record for this security record.

If no commands are entered and you press XMIT next to a security record on the list display then this is interpreted as a show (S) request for this security record.

## Display a Security Record

**smsec** provides two ways to display a security record.

1. From the **smsec** menu enter “S” for the function to be performed (show) and the record type, group, and item name of the security record to be displayed. (“P”, “TIP\$Y\$”, “DIE” in our example). If the group name is left blank then it will default to TIP\$Y\$.
2. Use the list function to list security records and type “S” next to the security record that you want to display.

The following screen is displayed in response to a request to display a security record.

\*\*\*\*\***Screen shot missing**\*\*\*\*\*

Press MSG WAIT to return to the **smsec** screen where the show request was made.

Press F2 to display the next security record.

Press F3 to display the previous security record.

Press F4 to update the security record. The screen will be redisplayed with the data fields padded with underscores.

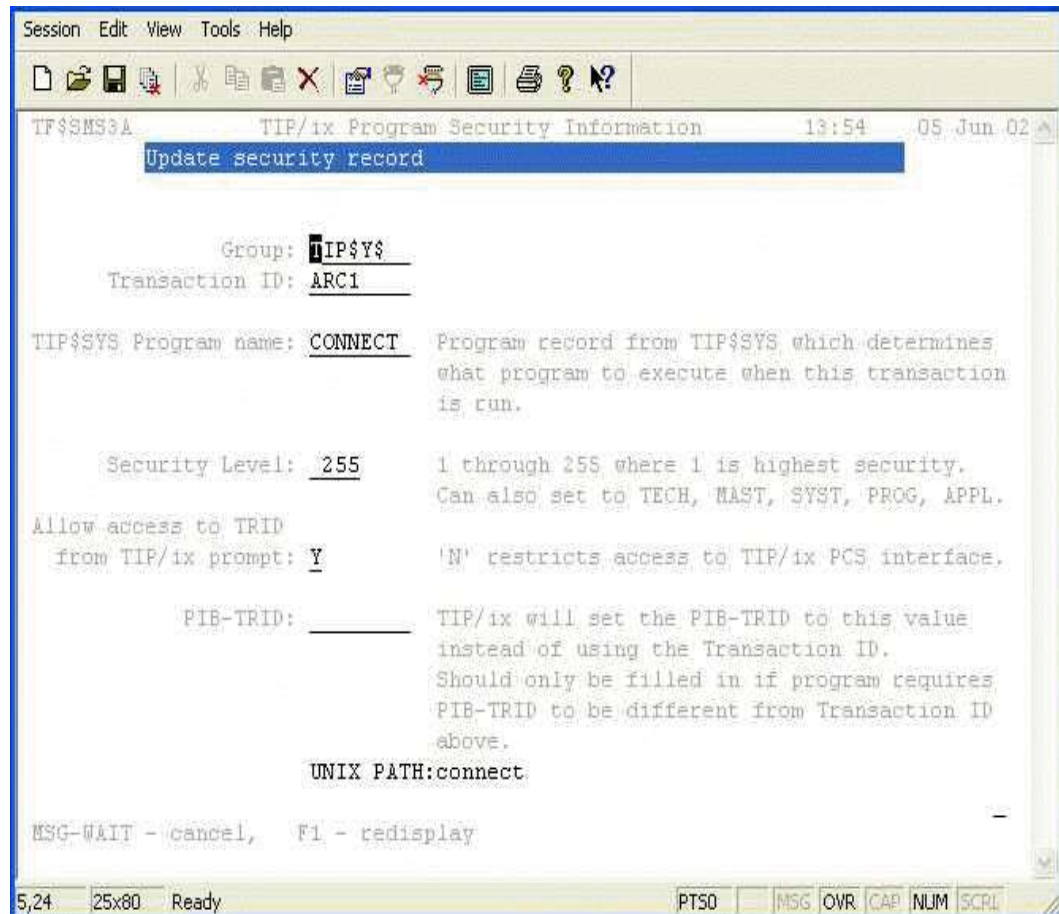
Press F5 to display the program, file, or queue definition referenced by this security record. This information is displayed from the TIP\$SYS file by invoking [smprog](#), [smfile](#), or [smqueue](#).

## Change a Security Record

**smsec** provides two ways to change (or update) a security record.

1. From the **smsec** menu enter “C” for the function to be performed (change) and the record type, group, and item name of the security record to be displayed. (“P”, “ARC”, “ARC1” in our example). If the group name is left blank then it will default to TIP\$Y\$.
2. Use the list function to list security records and type “C” next to the security record(s) that you want to change.

The following screen is displayed in response to a request to change a security record.



Press MSG WAIT to cancel the update and return to the **smsec** screen where the change request was made.

Fill in any desired changes and press XMIT to change the security record. For a description of the fields from the security record refer to the section on “Adding a Security Record”.

If the security record is changed successfully and no system record exists for the file, program, or queue (in TIP\$SYS) then a screen will be displayed prompting for the security information. Refer to [smfile](#), [smprog](#), and [smqueue](#) for details about how to define files, programs, and queues in TIP/ix.

If the group or security entry name is changed then the original security record is left unchanged and a new security entry is created in the new group or with the new name (Transaction ID, file name, or queue name).

## Delete a Security Record

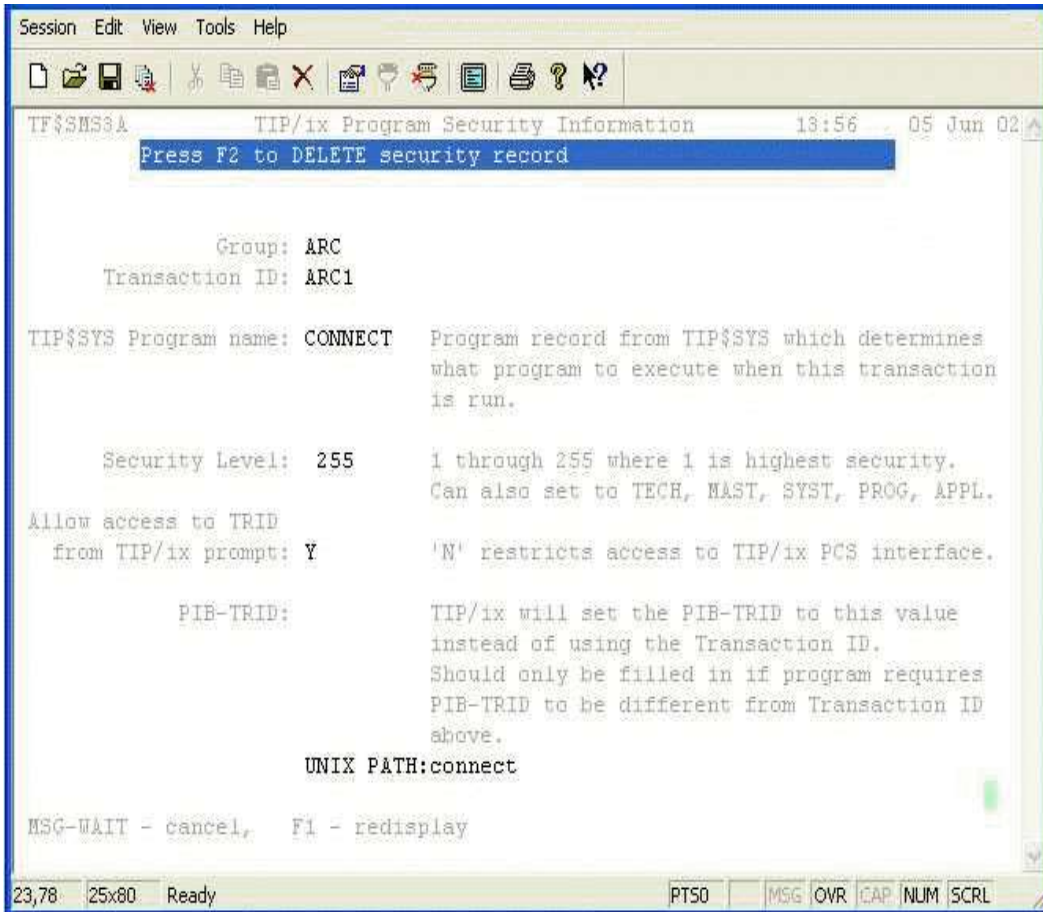
**smsec** provides two ways to delete a security record.

1. From the **smsec** menu enter “DE” for the function to be performed (delete) and the record type, group, and item name of the security

record to be deleted. (“P”, “ARC”, “ARC1” in our example). If the group name is left blank then it will default to TIP\$Y\$.

- Use the list function to list security records and type “D” next to the security record(s) that you want to delete.

The following screen is displayed in response to a request to delete a security record.



```

Session Edit View Tools Help
TIP$SMS3A TIP/ix Program Security Information 13:56 05 Jun 02
Press F2 to DELETE security record

 Group: ARC
Transaction ID: ARC1

TIP$SYS Program name: CONNECT Program record from TIP$SYS which determines
 what program to execute when this transaction
 is run.

 Security Level: 255 1 through 255 where 1 is highest security.
 Can also set to TECH, M&ST, SYST, PROG, APPL.

Allow access to TRID
from TIP/ix prompt: Y 'N' restricts access to TIP/ix PCS interface.

 PIB-TRID: TIP/ix will set the PIB-TRID to this value
 instead of using the Transaction ID.
 Should only be filled in if program requires
 PIB-TRID to be different from Transaction ID
 above.

 UNIX PATH:connect

MSG-WAIT - cancel, F1 - redisplay

23,78 25x80 Ready PTSO MSG OVR CAP NUM SCRL

```

Press MSG WAIT to cancel the delete request and return to the **smsec** screen where the delete request was made.

Press F2 to confirm that this is the security record that you want to delete.

If you press F2 and the security record is deleted then if there are no other security records that refer to the same TIP\$SYS record you will be asked if the TIP\$SYS record should be deleted (removed). Responding with “Y” will delete the system record that was referenced by the security record that was deleted.

### Delete all Security Records For a given File, Program, or Queue

This command provides a convenient method for deleting all security entries associated with a given file, program, or queue. From the **smsec** menu enter “DS” for the function to be performed (Delete by definition

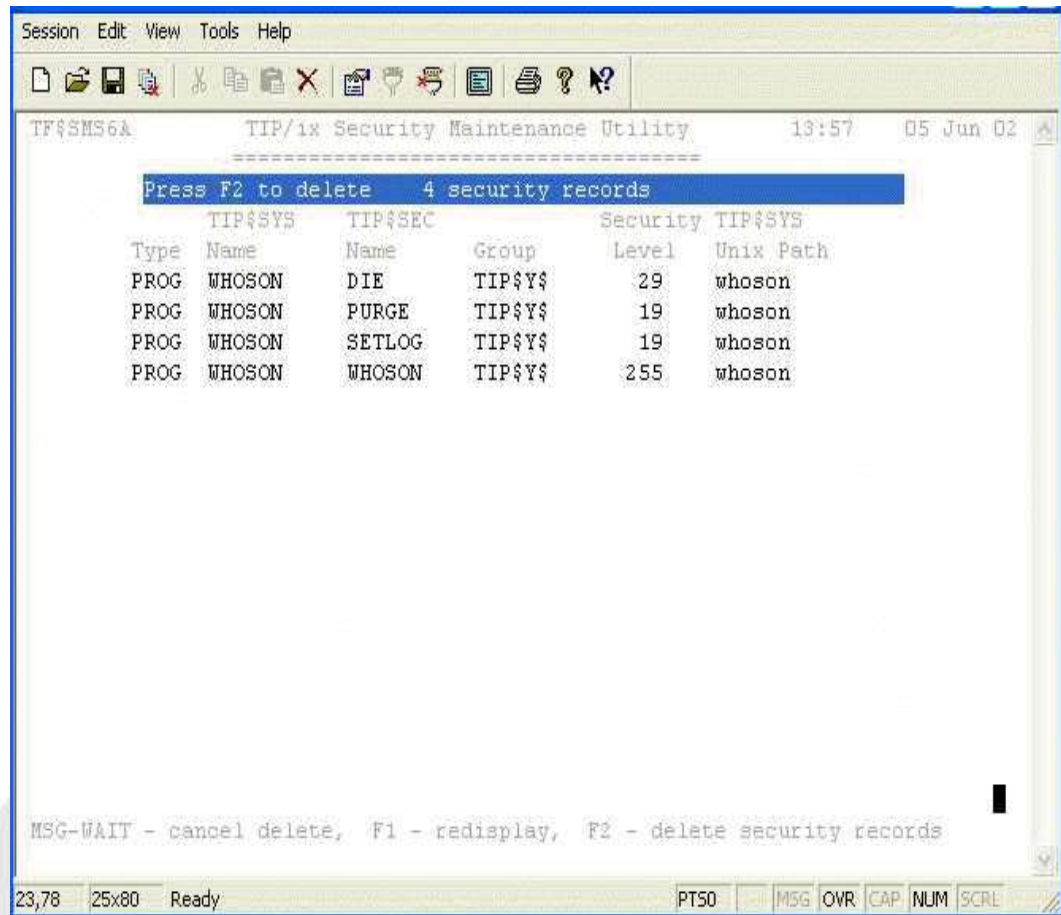


name in system file) and enter the type and name of the record from the system file (TIP\$SYS).

Note that you can only delete security records that have a security level greater than or equal to your own security level.

Since group name is not used by this function the item name can be filled in either the group name or item name position on the main menu screen for **smsec**.

The following screen is displayed in response to the “DS” command. In this example a request was made to delete all security records for the program definition for WHOSON (type “P” and item name of “WHOSON”).



Press MSG WAIT to cancel the delete request and return to the **smsec** screen where the delete request was made.

Press F2 to confirm that security records displayed are the ones that you want to delete. The message on row 3 indicates the number of security records that will be deleted when F2 is pressed. This information is most useful if more than a screen (12) of security records will be deleted when F2 is pressed.

If you press F2 and the security records are deleted then if there are no other security records that refer to the same TIP\$SYS record you will be

asked if the TIP\$SYS record should be deleted (removed). Responding with “Y” will delete the system record that was referenced by the security record that was deleted.

## Get System Record for a Security Record

**smsec** provides two ways to retrieve file, program, and queue definitions for security records.

Use the list function to list security records and type “G” next to the security record(s) for which you want to display the corresponding system records (from TIP\$SYS).

Display a security record (using either the show command or pressing XMIT next to a security entry on the list display) and then press F5.

The system record referenced by the security record will be displayed. This is done by calling one of [smfile](#), [smprog](#), or [smqueue](#) to display the file, program or queue definition associated with the security record.

If the system record does not exist then you will be asked if you wish to add one (via Yes/No prompt).

For information about file, program and queue definitions refer to the sections on the [smfile](#), [smprog](#), and [smqueue](#) programs.

The following screen is displayed when a request is made to display the system record for a program security record. (This example shows a request for the program definition associated with a security record.)

```

Session Edit View Tools Help
[Icons]
TF$SML4A TIP/ix Executable program definition 14:57 05-Jun-02
Program Name: TSP Comments: TIP Sample program
Program Type: TIP Locap:
Label/Path: /u/tipsrc/bin/tsp
Priority: Sets: TIPY
Open File(s): INVEN

Activation Area Sizes: CDA (2560) MCS (1536) WORK (1536)
 GDA (N) (Y/N)

Message editing: DICE (N) (Y/N) Upper case (N) (Y/N)
 Edit (N) (Y/N/char) FCC Edit () (Y/N)
 Full Screen XMIT (N) (Y/N) Recv Cmd Line in CDA (Y) (Y/N)
 Terminal control codes (A) (Ebcdic/Ascii)

Debug options: Log file level () (Min, All, Never)
 Debugger () (Animator, Vcdebug)

---Serial---Y/N---Number-of-Threads---Delay---Idle---Max-Use---Server-----
: Re-usable(N) min(0) max(0) (0) (0) (0) () :

MSG-WAIT - exit, F1 - redisplay, F2 - next, F3 - previous, F4 - update

23,79 25x80 Ready PTSO MSG OVR CAP NUM SCRL

```

Press MSG WAIT to return to the screen where the request was made for the system record.

Press F2 to display the next system record (of the same record type) in the TIP\$SYS file. The subsequent system records will not be associated with the original security record (since a security record is only refers to a single system record).

Press F3 to display the previous system record (of the same record type) in the TIP\$SYS file. The subsequent system records will not be associated with the original security record (since a security record is only refers to a single system record).

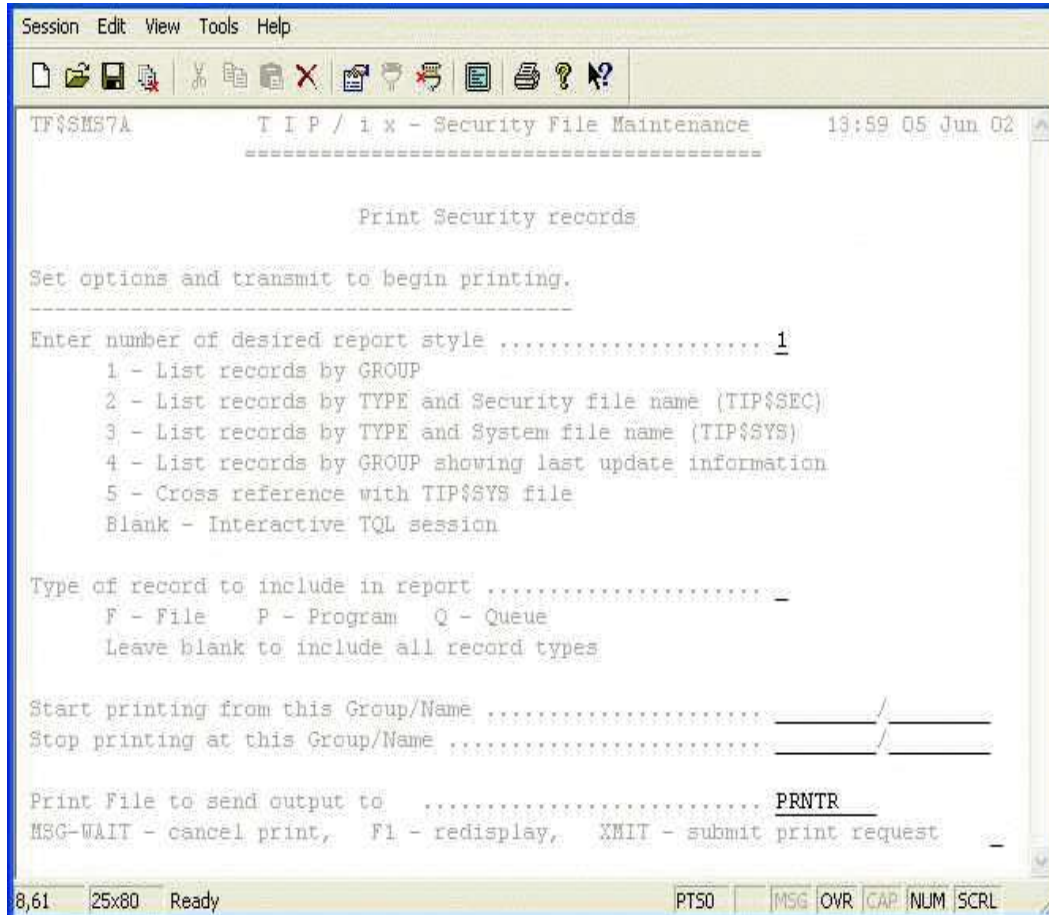
Press F4 to update the system record (file, program, or queue definition). The screen will be redisplayed with the data fields padded with underscores.

## Print Security Records

To print security records enter "PR" as the function to be performed (print) on the **smsec** menu display. A record type can be supplied to limit the print request to a particular type (file, program, queue) of security record.

After pressing **XMIT**, you will be prompted to fill in the print options and report style.

When you select the Print function from **smsec** the following screen is displayed. Enter the desired print options and press transmit to print security records from the TIP\$SEC file.



```

Session Edit View Tools Help

TIP / i x - Security File Maintenance 13:59 05 Jun 02

Print Security records

Set options and transmit to begin printing.

Enter number of desired report style 1
 1 - List records by GROUP
 2 - List records by TYPE and Security file name (TIP$SEC)
 3 - List records by TYPE and System file name (TIP$SYS)
 4 - List records by GROUP showing last update information
 5 - Cross reference with TIP$SYS file
 Blank - Interactive TQL session

Type of record to include in report
 F - File P - Program Q - Queue
 Leave blank to include all record types

Start printing from this Group/Name /
Stop printing at this Group/Name /

Print File to send output to PRNTR
MSG-WAIT - cancel print, F1 - redisplay, XMIT - submit print request

8,61 25x80 Ready PTSO MSG OVR CAP NUM SCRL

```

The following is a description of the fields in the preceding screen.

### Report style

- 1** Report security information ordered by application group, record type, and security entry name. A single line is printed for each security record.
- 2** Report security information ordered by record type and security entry name. A single line is printed for each security record.
- 3** Report security information ordered by record type and system record (TIP\$SYS) name. A single line is printed for each security record.
- 4** Report date, time, and user id performing last update of each security record. Report is ordered by application group, record type, and security

entry name. A single line is printed for each security record.

- 5** A cross reference report that shows the UNIX path associated with program and file security records and shows the server name associated with queue security records.

**<blank>**

Leave this entry blank to initiate an interactive TQL session with the TQL program TIPSEC

**Record Type**

Type of security record to include in report. Leave blank for all record types.

**Start printing from this Group/Name**

**Stop printing at this Group/Name**

Range of security records to include in report. For report types that are not ordered by group (types 2 and 3) the name can be filled in either position. However, for types 2 and 3 the name is only meaningful if a record type is filled in.

**Print File**

Print file to route report to. This must be a valid TIPPRINT destination. Print files are defined by smprint and of course aliases can be set up using smsec. AUX0 can be used to direct report output to the terminal.

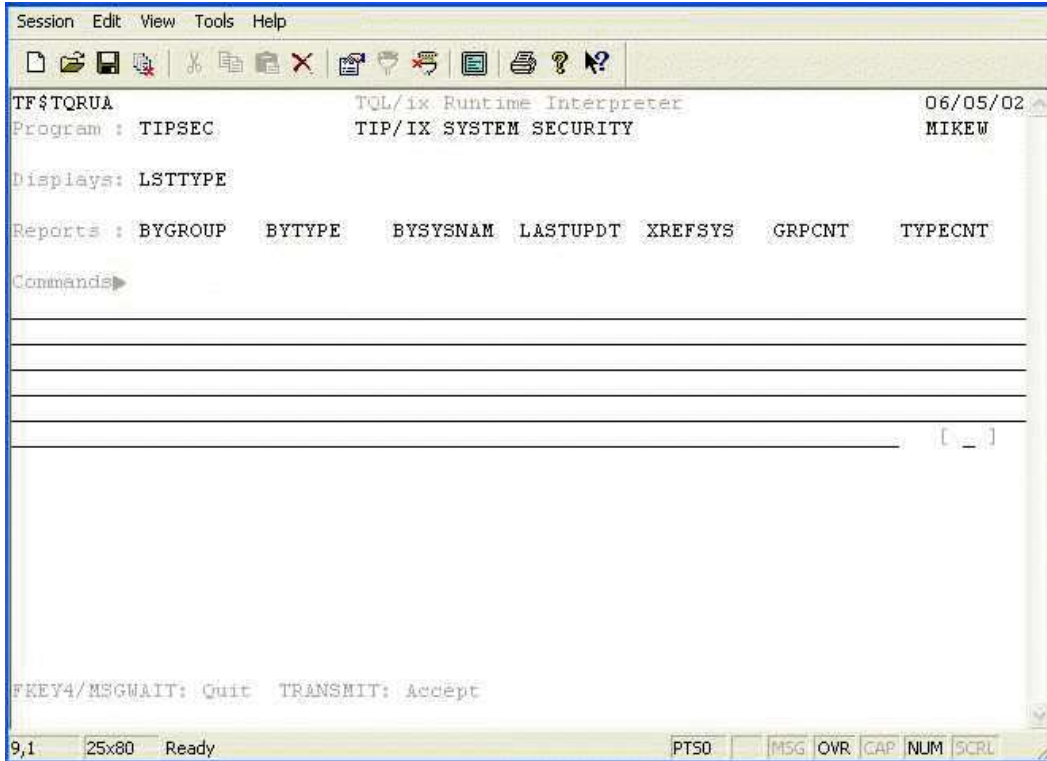
The print function is implemented by the supplied TQL program TIPSEC. If this program is not compiled and available then the print function will not work.

## Browse Security Definitions with TQL

The **TIPSEC** TQL program enables you to interrogate security definitions with ad hoc TQL queries. This TQL program can be run either:

- Directly from the TIP/ix command prompt by entering "TQL TIPSEC"
- From the **smsec** menu enter "BR" for the function to be performed (browse).

The following screen is displayed when you run **TIPSEC**.



At this point you can make any valid TQL request including:

- Request a predefined display.
- Request a predefined report.
- Make an ad hoc request to list (or print) fields from the security definitions.

See the TIP/ix TQL Reference for details on interacting with TQL.

The predefined displays include:

#### **LSTTYPE**

Display 16 security records at a time with a single line per security record. The display is ordered by record type (file, queue, or program) and name of object in security file.

The predefined reports include:

#### **BYGROUP**

Lists security records order by group, record type, and name of object in security file. Level breaks are displayed after each group.

#### **BYTYPE**

Lists security records order by record type, and name of object in security file. Level breaks are displayed after each record type (file, queue, program).

**BYSYSNAME**

Lists security records order by record type, and entry referenced in the TIP\$SYS file (smfile, smqueue, smprog entry). Level breaks are displayed after each record type (file, queue, program).

**LASTUPDT**

A report showing who last updated security records (and when). The report is ordered by group with level breaks after each group.

**XREFSYS**

A report showing for each the security record the path information from the associated TIP\$SYS record.

**GRPCNT**

Summary report that displays the total number of security records of each type in each application group.

**TYPECNT**

Summary report that displays the total number of security records of each type (file, queue, program).

In addition to using the predefined displays and reports it is possible to perform ad hoc queries based on field settings in security definitions. The following is a list of field names and the corresponding security definition attribute:

| Security Definition Attribute           | TQL Field Names       |
|-----------------------------------------|-----------------------|
| Group                                   | SC-GROUP              |
| Record Type                             | SC-TYPE               |
|                                         | SC-FILE (88 LEVEL)    |
|                                         | SC-PROGRAM (88 LEVEL) |
|                                         | SC-QUEUE (88 LEVEL)   |
| Item Name                               | SC-ITEM               |
| Transaction ID                          |                       |
| File name                               |                       |
| Queue name                              |                       |
| TIP\$SYS Name                           | SC-REAL-ITEM          |
| Security Level                          | SC-SECURITY           |
| Allow access to TRID from TIP/ix prompt | SC-PROG-PERMS         |
| PIB-TRID                                | SC-TRID               |
| File Access                             | SC-FILE-PERMS         |

Key of Reference

SC-KEY-REF

**Examples:**

List all security entries for the group ARC

```
BYGROUP FROM ARC TO ARC
```

List all security entries ordered by security

```
LIST (SC-ITEM SC-GROUP SC-TYPE SC-SECURITY
 SORT BY SC-SECURITY
```

List all program records

```
BYTYPE FROM SC-PROGRAM
```

Print security records on auxiliary printer ordered by group

```
BYGROUP ON AUX1
```

---

## smterm - Terminal Definition

The **smterm** transaction is used to maintain terminal definitions. A terminal definition associates a terminal name with a device and a session number at the device. A device can be identified by an IP address, a host name from the */etc/hosts* file, or the name of a device from the */dev* directory. Terminal definitions are also used to set and control terminal attributes that are related to operation in a TIP/ix environment.

If a user has set the environment variable **TIPTERM**, the value of this variable is used by TIP/ix as the terminal name (**PIB-TERM-NAME**) and TIP/ix does not attempt to use a terminal definition to set the terminal attributes for the TIP/ix session. In this case, **PIB-TID** is set to the last 4 characters of the environment variable **TIPTERM**.

If **TIPTERM** does not exist as a variable in the user's environment then starting a TIP/ix session (**tipix**) results in TIP looking for a terminal definition based on the device and session number at that device where the session was initiated. If a terminal definition is found then the terminal attributes are set using the information in the definition.

One use of **smterm** is to assign terminal names (**PIB-TERM-NAME**) and terminal ids (**PIB-TID**). Other uses are to set attributes related to the operation of **TIPPRINT**, to control which terminals can receive unsolicited messages, and to assign terminal relationships (master and bypass terminals).

Some programming facilities (**TIPPRINT** for example) make use of generic terminal names (**\*MST** for master and **\*BYP** for bypass) so that the terminal referenced depends on where the program (using the generic



name) is run. Each terminal definition created with **smterm** can assign the name of a terminal definition to use as its master (\*MST) and a terminal definition to use as its bypass (\*BYP).

The terminal definitions are kept in a file known to TIP/ix as TIP\$SYS. However, a second file, TIP\$TRM, is used to provide additional keys of reference to the terminal definitions. The terminal name is used in composing the primary key in TIP\$SYS and it is the primary key in TIP\$TRM. The combination of device and session number at a device are used to create a *normalized address* which is an alternate key to TIP\$TRM (and for which duplicates are not allowed).

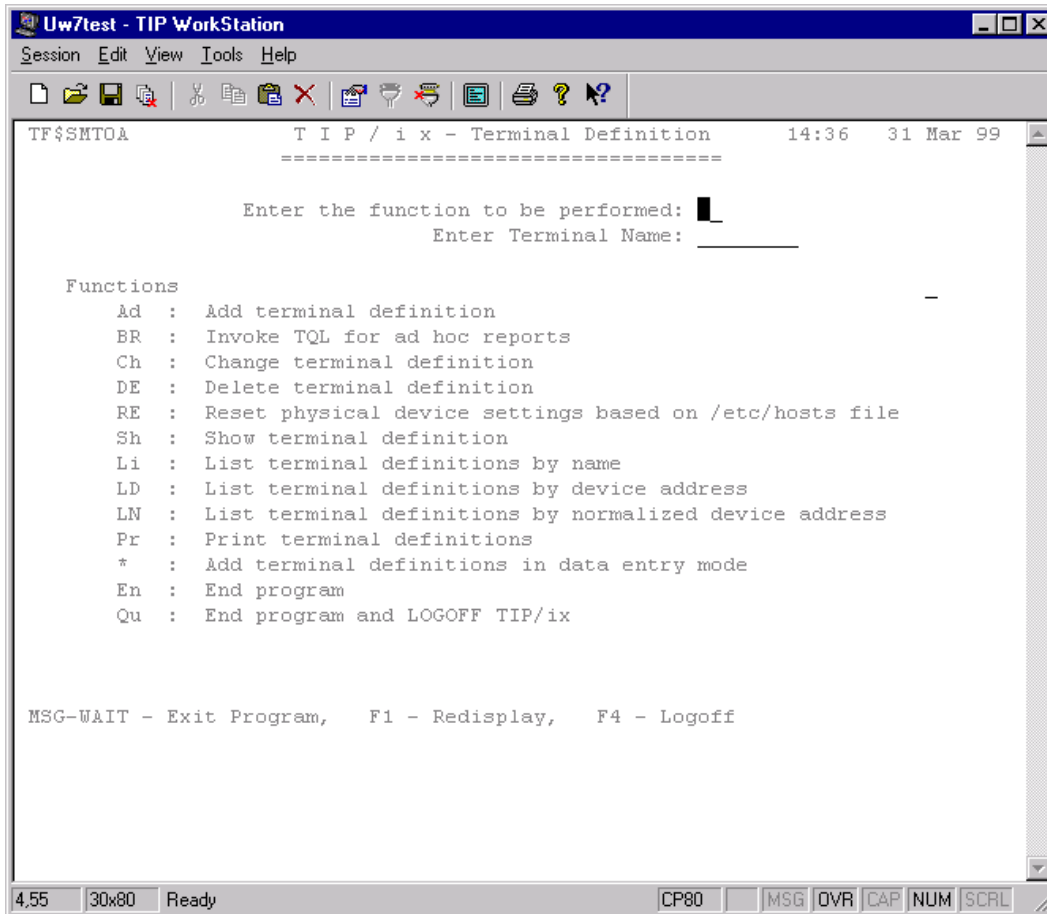
When migrating from a TIP/30 environment the terminal definitions created by **smterm** replace the terminal cluster definitions in the TIP/30 configuration.

**Syntax:**

```
smterm [function] [terminalname]
```

If a valid function is supplied then **smterm** will perform that function and then terminate when that function has been completed. Most functions require a terminal name to act upon.

If no function is supplied then **smterm** displays the following terminal maintenance menu screen.



```

Uw7test - TIP WorkStation
Session Edit View Tools Help
TF$SMT0A T I P / i x - Terminal Definition 14:36 31 Mar 99
=====
Enter the function to be performed: L
Enter Terminal Name: _____

Functions
Ad : Add terminal definition
BR : Invoke TQL for ad hoc reports
Ch : Change terminal definition
DE : Delete terminal definition
RE : Reset physical device settings based on /etc/hosts file
Sh : Show terminal definition
Li : List terminal definitions by name
LD : List terminal definitions by device address
LN : List terminal definitions by normalized device address
Pr : Print terminal definitions
* : Add terminal definitions in data entry mode
En : End program
Qu : End program and LOGOFF TIP/ix

MSG-WAIT - Exit Program, F1 - Redisplay, F4 - Logoff

4,55 30x80 Ready CP80 MSG OVR CAP NUM SCRL

```

From this menu you may perform all the traditional maintenance functions on the terminal definitions. You may add new terminal definitions; change, delete, show, or print existing terminal definitions and list a summary of all terminal definitions.

In the preceding screen, the function to be performed can be abbreviated to the portion shown in upper case. For example the function *add* can be entered as “A”.

## Add a Terminal Definition

To define a new terminal definition, enter “A” for the function to be performed (add) and the name of the terminal (“TRM1” in our example). After pressing **XMIT**, the following screen is displayed to allow you to enter the terminal definition information.

```

Uw7test - TIP WorkStation
Session Edit View Tools Help
TF$SMT1A TIP/ix Terminal Definition 14:37 03/31/99
Enter information for terminal TRM1
Term Name (PIB-TERM-NAME): TRM1 Comments:
PIB-TID for this terminal: (leave blank to use system default)
OS/1100 PID: (only applies to IMS/1100 & DPS/1100)
Sets:
Device associated with this definition (fill in 1 of the following):
IP address.....:
OR Host name from /etc/hosts file:
OR Device name from /dev:
Session number at this device:

Terminal Name of MASTER (*MST): BYPASS (*BYP):
Maximum user security:
User must be member of group: or
Allow terminal to receive unsolicited messages (Y/N)..... Y
TIPPRINT settings
Default Lines per Page...:
Form Feed requirements (Top/Bottom/None/Yes(top & bottom))
Precede each Form Feed with a Line Feed (Y/N)
Insert Line Feed at end of Print Buffer (Y/N)
Unattended operation (Y/N/B)
MSG-WAIT - cancel, F1 - redisplay

```

At the bottom of the screen is a summary of function keys and their actions.

At any time you may enter MSG WAIT to cancel the function and return to the main menu screen.

Pressing F1 causes the screen to be re-displayed (Caution! any data entered on the screen will be lost).

Fill in the information for the terminal definition and press XMIT to add the definition to the TIP\$SYS file.

The following is a description of each field in the preceding screen.

**Term name:**

The name of this terminal definition. This value will be used for PIB-TERM-NAME for a TIP/ix session using the device and session number from this definition.

**Maximum user security:**

The maximum security of any user who connects to TIP/ix from that terminal. If the user record shows higher security then it will be reduced.

**User must be member of group:**

The group names (if given) are cross checked against the user's groups. If there is no match the user is rejected. This feature provides some security on the use of specific terminals.

**Comments:**

Optional information describing the terminal definition.

**PIB-TID**

Value to use for PIB-TID for a TIP/ix session using the device and session number from this definition.

If this field is left blank then PIB-TID will be the last 4 characters of the PIB-TERM-NAME. For example if PIB-TERM-NAME is "JOHN1" then PIB-TID will be "OHN1".

**OS/1100 PID**

Value to use for PID (position identifier) when emulating IMS/1100 and DPS/1100 applications from a TIP/ix session using this terminal definition.

**Sets** This entry is used to identify a set of records in the TIP\$SYS file that are in some way related. For example you could mark all TIP\$SYS records used in a payroll application by filling in the set name with "PAYROLL". The intention is that these records can be migrated as a set from one TIP/ix system to another using the "[tippack](#)" transaction.

**Fill in one of: IP address, Host name, or Device name****IP address**

IP address of device. This could be filled in for a computer on a network accessing TIP/ix via a terminal emulator (such as TIP/fe).

**Host name**

If this is filled in then TIP/ix will use the IP address corresponding to this host name in the /etc/hosts file. This could be filled in for a computer on a network accessing TIP/ix via a terminal emulator (such as TIP/fe).

**Device name**

If this is filled in then it should correspond to the device name from the /dev directory that corresponds to this terminal. This could be filled in for a real UNIX terminal.

**Session number at this device**

Since a single device may be capable of having multiple sessions this field is required to provide a unique terminal name for each session. For example with TIP/fe it is possible to have many sessions. This number corresponds

to the session number with the UNIX system where this TIP/ix system is running.

For example, TIP/fe sessions 1 and 3 could be attached to the UNIX system “alpha” and session 2 could be attached to the UNIX system “beta”. In this case the 3rd TIP/fe session would be session number 2 (2nd session for the device on alpha) for TIP/ix’s terminal naming purposes.

#### **Terminal Name of MASTER (\*MST)**

The name of the terminal definition to be considered the MASTER terminal for this terminal from a TIP/ix programming perspective. For TIP/ix sessions using this terminal definition use this value whenever a program references the generic terminal id of \*MST.

#### **Terminal Name of BYPASS (\*BYP)**

The name of the terminal definition to be considered the BYPASS terminal for this terminal from a TIP/ix programming perspective. For TIP/ix sessions using this terminal definition use this value whenever a program references the generic terminal id of \*BYP.

#### **Maximum user security**

The Max security is the maximum security of any user who connects to TIP/ix from that terminal. If the user record shows higher security then it is reduced.

#### **User must be member of group:**

The group names (if given) are cross checked against the user's groups. If there is no match, the user is rejected.

#### **TIPPRINT settings**

##### **Default Lines per Page**

The default value to use for lines per page when printing from this terminal. TIPPRINT uses this value if the value of PRINT-PAG-LEN is zero (or spaces) when printing is initiated (FCS-OPEN function).

##### **Form Feed requirements**

The default form feed processing performed by TIPPRINT when printing from this terminal. TIPPRINT performs the default processing when the value of PRINT-TOP-OF-FORM is set to a space when printing is initiated (FCS-OPEN function).

**T** TIPPRINT forces the first character of output to be a form feed (inserting a form feed if one is not supplied by the program) and .ensures that there is no form feed at

the end of printing (stripping any form feeds from the end of the print buffer).

- B** TIPPRINT ensures that there is no form feed at the beginning of printing (stripping any form feeds that precede print data) and forces the last character of output to be a form feed (inserting a form feed if one was not supplied by the program).
- N** TIPPRINT ensures that if there is no form feed at the beginning or end of printing. Form feeds preceding print data and trailing print data are removed by TIPPRINT.
- Y** TIPPRINT ensures that there is a form feed at both the beginning and end of printing. If the form feeds are not supplied by the program then they will be inserted by TIPPRINT

#### **Precede each Form Feed with a Line Feed**

The line feed handling for form feeds when using TIPPRINT from this terminal.

- Y** TIPPRINT will insert a line feed ahead of any form feeds received. This setting may be required for printers that do not automatically supply a line feed when a form feed is received.
- N** TIPPRINT will not insert any characters when a form feed is received.

#### **Insert Line Feed at end of Print Buffer**

The default line feed handling at end of buffer when using TIPPRINT from this terminal. TIPPRINT uses this value if the value of PRINT-LINE-FEED is a space when printing is initiated (FCS-OPEN function).

- Y** TIPPRINT will insert a line feed at then end of each print buffer.

#### **Allow terminal to receive unsolicited msgs**

Use this field to control whether or not a terminal will be allowed to receive unsolicited messages. The programs msg and apb are used to send unsolicited messages.

- Y** TIP/ix will allow the terminal to receive unsolicited messages.
- N** TIP/ix will not allow the terminal to receive unsolicited messages and there will be no indication at this terminal of any attempts to send unsolicited messages to it.

#### **Unattended operation**

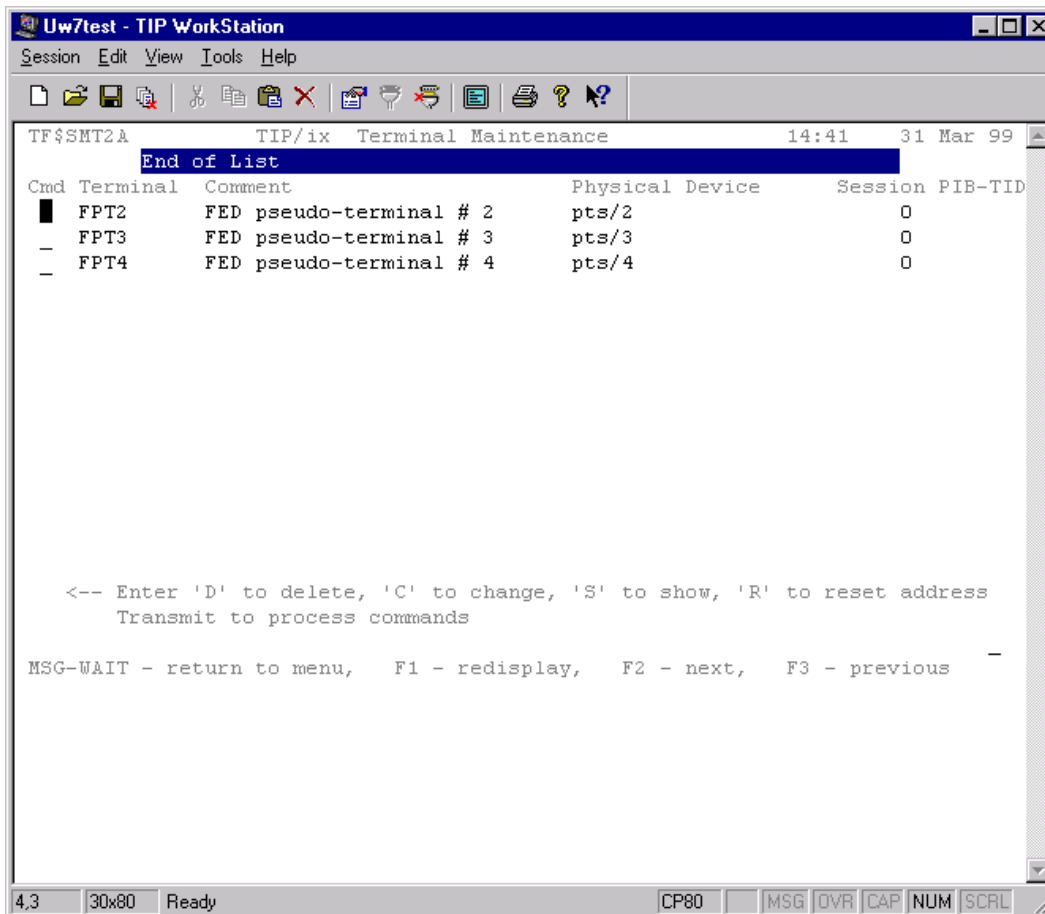
- Y** If set to Y then this terminal will open when started but will not accept any input from the user. It will be used the same way an unattended terminal was used in ICAM on OS/3.
- B** When set to B (Background terminal), the terminal will act the same way as when it is set to Y only that the user will not see the session but may send messages to it.
- N** This is the default setting. N or Blank, are one in the same.

## List Terminal Definitions By Terminal Name

From the **smterm** menu enter "LI" for the function to be performed (List) and a name to use as the starting point for the list. This does not have to be a complete name. For example a name of "F" could be used to start listing terminal definitions with a name greater than (or equal to) "F".

The list function displays a single line summary for each terminal definition showing the terminal name, the terminal description, the physical device and session number at that device, and the PIB-TID value.

The following screen is displayed in response to the list command.



```

Uw7test - TIP WorkStation
Session Edit View Tools Help
TF$SMT2A TIP/ix Terminal Maintenance 14:41 31 Mar 99
End of List
Cmd Terminal Comment Physical Device Session PIB-TID
FPT2 FED pseudo-terminal # 2 pts/2 0
FPT3 FED pseudo-terminal # 3 pts/3 0
FPT4 FED pseudo-terminal # 4 pts/4 0

<-- Enter 'D' to delete, 'C' to change, 'S' to show, 'R' to reset address
Transmit to process commands

MSG-WAIT - return to menu, F1 - redisplay, F2 - next, F3 - previous
4,3 30x80 Ready CP80 MSG OVR CAP NUM SCRL

```

Press MSG WAIT to return to the main menu screen.

Press F2 to display the next screen-full of terminals.

Press F3 to display the previous screen-full of terminals.

You can enter single character commands next to the terminals and when you XMIT the commands will be processed sequentially (one at a time) from the top of the display. The commands that can be entered are:

- C** Change terminal definition
- D** Delete terminal definition
- S** Show terminal definition
- R** Reset the normalized addresses (IP address and session number) based on the values for any references to host names in /etc/hosts.

If no commands are entered and you press XMIT next to a terminal on the list display then this is interpreted as a show (S) request for this terminal definition.



## List Terminal Definitions By Device Address

From the **smterm** menu enter “LD” for the function to be performed and a starting point for the list. For example a name of “A” could be used to start listing terminal definitions where the device field value is greater than (or equal to) “A”.

The list function displays a single line summary for each terminal definition showing the physical device and session number at that device, the terminal name, and the *normalized* address.

A *normalized* address is created from the device and session information. If the device is a host name from */etc/hosts* then the normalized address is created using the IP address for that host. All IP addresses are padded with leading zeros so that all segments consist of 3 digits and the session number is padded with leading zeros to produce a 4 digit number. Then the session number is concatenated to the IP address separated by a # character.

The following screen is displayed in response to the list command.

```

Uw7test - TIP WorkStation
Session Edit View Tools Help
TF$SMT4A TIP/ix Terminal Maintenance 14:42 31 Mar 99
End of List
Cmd Physical Device Session Terminal Normalized Device & Session
pts/2 0 FPT2 pts/2
pts/3 0 FPT3 pts/3
pts/4 0 FPT4 pts/4

<-- Enter 'D' to delete, 'C' to change, 'S' to show, 'R' to reset address
Transmit to process commands

MSG-WAIT - return to menu, F1 - redisplay, F2 - next, F3 - previous
4,3 30x80 Ready CP80 MSG OVR CAP NUM SCRL

```

Press MSG WAIT to return to the main menu screen.

Press F2 to display the next screen-full of terminals.

Press F3 to display the previous screen-full of terminals.

You can enter single character commands next to the terminals and when you XMIT the commands will be processed sequentially (one at a time) from the top of the display. The commands that can be entered are:

- C** Change terminal definition
- D** Delete terminal definition
- S** Show terminal definition
- R** Reset the normalized addresses (IP address and session number) based on the values for any references to host names in `/etc/hosts`.

If no commands are entered and you press XMIT next to a terminal on the list display then this is interpreted as a show (S) request for this terminal definition.

## List Terminal Definitions By Normalized Device Address

From the **smterm** menu enter "LN for the function to be performed and a starting point for the list. For example a name of "022" could be used to start listing terminal definitions where the normalized device field value is greater than (or equal to) "022".

A *normalized* address is created from the device and session information. If the device is a host name from `/etc/hosts` then the normalized address is created using the IP address for that host. All IP addresses are padded with leading zeros so that all segments consist of 3 digits and the session number is padded with leading zeros to produce a 4 digit number. Then the session number is concatenated to the IP address separated by a # character.

The list function displays a single line summary for each terminal definition showing the *normalized* address, terminal name, and the physical device and session number at that device.

The following screen is displayed in response to the list command.

```

Uw7test - TIP WorkStation
Session Edit View Tools Help
TF$SMT3A TIP/ix Terminal Maintenance 14:42 31 Mar 99
End of List
Cmd Normalized Device & Session Terminal Address Session
pts/2 FPT2 pts/2 0
pts/3 FPT3 pts/3 0
pts/4 FPT4 pts/4 0

<-- Enter 'D' to delete, 'C' to change, 'S' to show, 'R' to reset address
Transmit to process commands

MSG-WAIT - return to menu, F1 - redisplay, F2 - next, F3 - previous

```

Press MSG WAIT to return to the main menu screen.

Press F2 to display the next screen-full of terminals.

Press F3 to display the previous screen-full of terminals.

You can enter single character commands next to the terminals and when you XMIT the commands will be processed sequentially (one at a time) from the top of the display. The commands that can be entered are:

- C** Change terminal definition
- D** Delete terminal definition
- S** Show terminal definition
- R** Reset the normalized addresses (IP address and session number) based on the values for any references to host names in /etc/hosts.

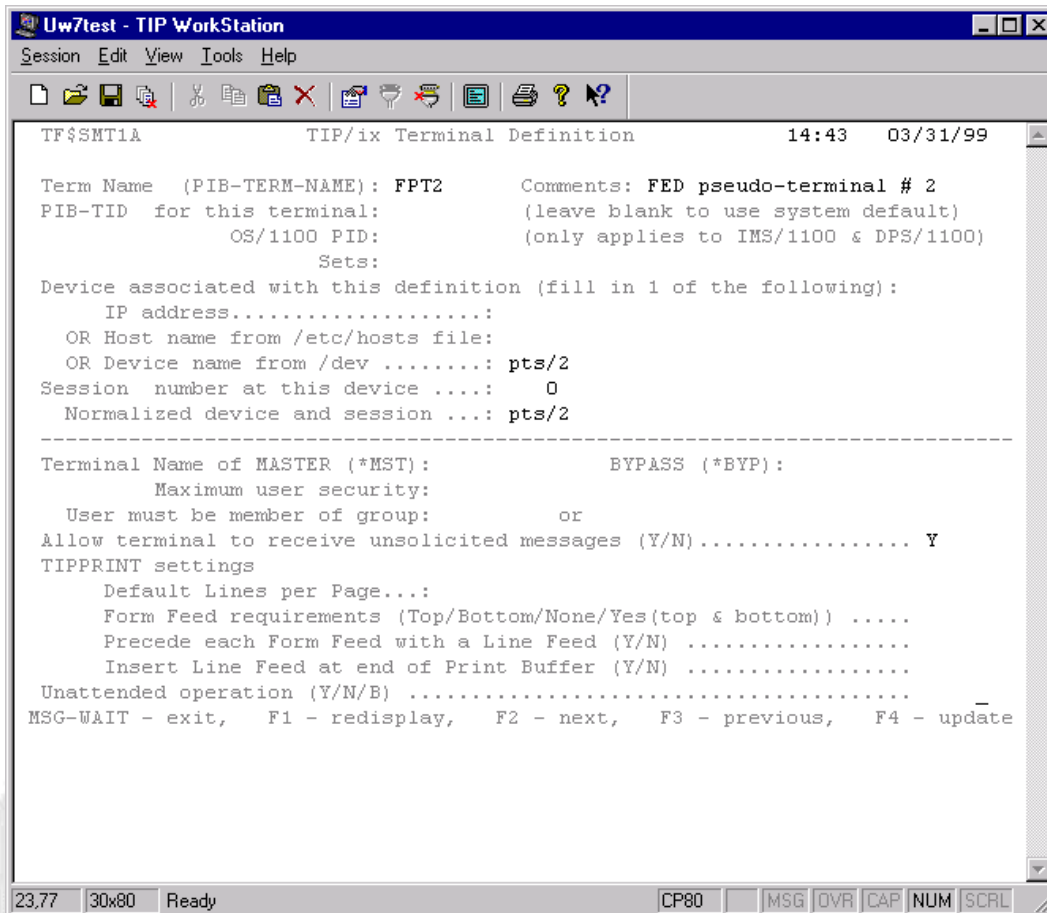
If no commands are entered and you press XMIT next to a terminal on the list display then this is interpreted as a show (S) request for this terminal definition.

## Display a Terminal Definition

**smterm** provides two ways to display a terminal definition.

1. From the **smterm** menu enter “S” for the function to be performed (show) and the name of the terminal definition to be displayed. (“FPT2” in our example).
2. Use the list function to list terminal definitions and type “S” next to the terminal definition(s) that you want to display.

The following screen is displayed in response to a request to display a terminal definition.



```

Uw7test - TIP WorkStation
Session Edit View Tools Help
TF$SMT1A TIP/ix Terminal Definition 14:43 03/31/99
Term Name (PIB-TERM-NAME): FPT2 Comments: FED pseudo-terminal # 2
PIB-TID for this terminal: (leave blank to use system default)
OS/1100 PID: (only applies to IMS/1100 & DPS/1100)
Sets:
Device associated with this definition (fill in 1 of the following):
IP address.....:
OR Host name from /etc/hosts file:
OR Device name from /dev: pts/2
Session number at this device: 0
Normalized device and session: pts/2

Terminal Name of MASTER (*MST): BYPASS (*BYP):
Maximum user security:
User must be member of group: or
Allow terminal to receive unsolicited messages (Y/N)..... Y
TIPPRINT settings
Default Lines per Page...:
Form Feed requirements (Top/Bottom/None/Yes(top & bottom))
Precede each Form Feed with a Line Feed (Y/N)
Insert Line Feed at end of Print Buffer (Y/N)
Unattended operation (Y/N/B)
MSG-WAIT - exit, F1 - redisplay, F2 - next, F3 - previous, F4 - update
23.77 30x80 Ready CP80 MSG OVR CAP NUM SCRL

```

Press MSG WAIT to return to the **smterm** screen where the show request was made.

Press F2 to display the next terminal definition.

Press F3 to display the previous terminal definition.

Press F4 to update the terminal definition. The screen will be redisplayed with the data fields padded with underscores.

## Change a Terminal Definition

**smterm** provides two ways to change (or update) a terminal definition.

1. From the **smterm** menu enter “CH” for the function to be performed (change) and the name of the terminal definition to be changed. (“FPT2” in our example).
2. Use the list function to list terminal definitions and type “C” next to the terminal definition(s) that you want to change.

The following screen is displayed in response to a request to change a terminal definition.

The screenshot shows a window titled "Uw7test - TIP WorkStation" with a menu bar (Session, Edit, View, Tools, Help) and a toolbar. The main window displays the following text:

```

TF$SMT1A TIP/ix Terminal Definition 14:43 03/31/99
Update information for terminal FPT2
Term Name (PIB-TERM-NAME): FPT2 Comments: FED pseudo-terminal # 2
PIB-TID for this terminal: _____ (leave blank to use system default)
OS/1100 PID: _____ (only applies to IMS/1100 & DPS/1100)
Sets: _____
Device associated with this definition (fill in 1 of the following):
 IP address.....: _____
 OR Host name from /etc/hosts file: _____
 OR Device name from /dev: pts/2
Session number at this device: 0
Normalized device and session ...: pts/2

Terminal Name of MASTER (*MST): _____ BYPASS (*BYP): _____
Maximum user security: _____
User must be member of group: _____ or _____
Allow terminal to receive unsolicited messages (Y/N)..... Y
TIPPRINT settings
Default Lines per Page....: _____
Form Feed requirements (Top/Bottom/None/Yes(top & bottom)) -
Precede each Form Feed with a Line Feed (Y/N) -
Insert Line Feed at end of Print Buffer (Y/N) -
Unattended operation (Y/N/B) - -
MSG-WAIT - cancel, F1 - redisplay

```

At the bottom of the window, there is a status bar showing "3.31 30x80 Ready" and control keys: "CP80 MSG OVR CAP NUM SCRL".

Press MSG WAIT to cancel the update and return to the **smterm** screen where the change request was made.

Fill in any desired changes and press XMIT to change the terminal definition. For a description of the fields on the terminal definition refer to the section on “Adding a Terminal Definition”.

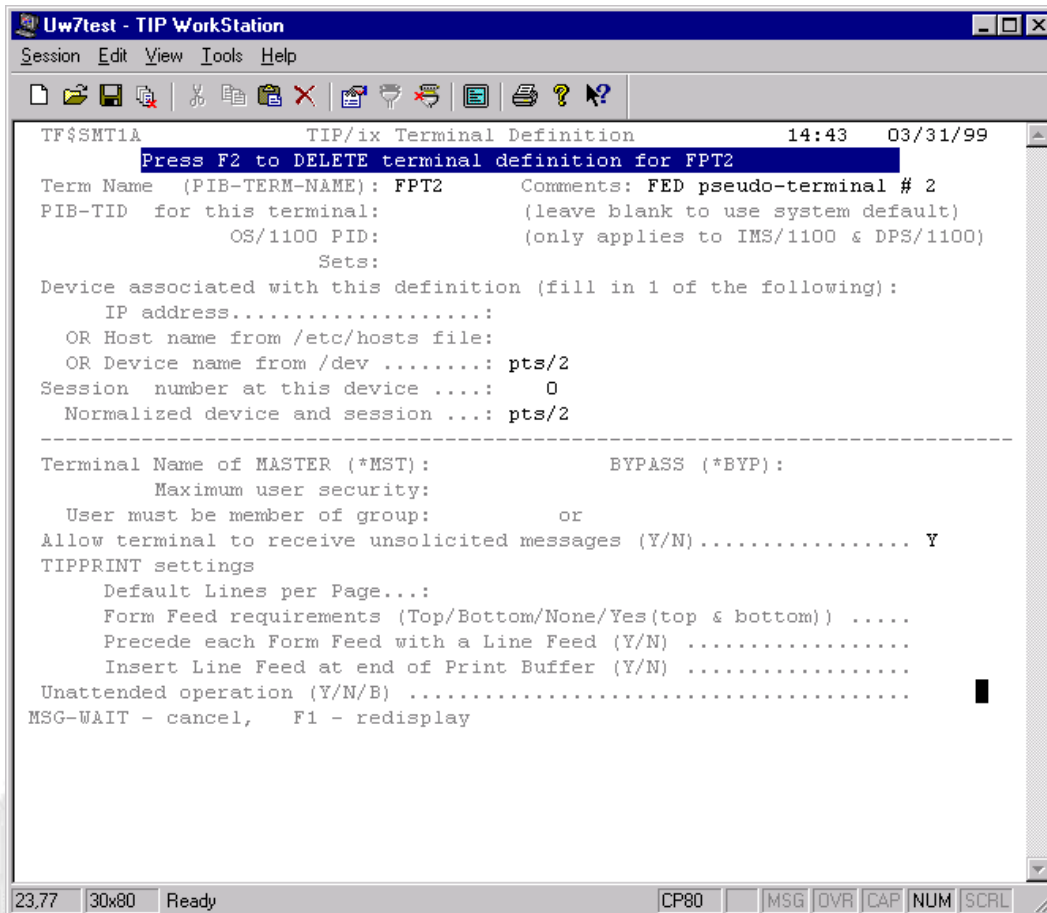
If the terminal name is changed then the original terminal definition is left unchanged and a new terminal definition is added for the new name.

## Delete a Terminal Definition

**smterm** provides two ways to delete a terminal definition.

1. From the **smterm** menu enter “DE” for the function to be performed (delete) and the name of the terminal definition to be deleted. (“FPT2” in our example).
2. Use the list function to list terminal definitions and type “D” next to the terminal definition(s) that you want to delete.

The following screen is displayed in response to a request to delete a terminal definition.



```

Uw7test - TIP WorkStation
Session Edit View Tools Help
TF$SMT1A TIP/ix Terminal Definition 14:43 03/31/99
Press F2 to DELETE terminal definition for FPT2
Term Name (PIB-TERM-NAME): FPT2 Comments: FED pseudo-terminal # 2
PIB-TID for this terminal: (leave blank to use system default)
OS/1100 PID: (only applies to IMS/1100 & DPS/1100)
Sets:
Device associated with this definition (fill in 1 of the following):
IP address.....:
OR Host name from /etc/hosts file:
OR Device name from /dev: pts/2
Session number at this device: 0
Normalized device and session: pts/2

Terminal Name of MASTER (*MST): BYPASS (*BYP):
Maximum user security:
User must be member of group: or
Allow terminal to receive unsolicited messages (Y/N)..... Y
TIPPRINT settings
Default Lines per Page...:
Form Feed requirements (Top/Bottom/None/Yes(top & bottom))
Precede each Form Feed with a Line Feed (Y/N)
Insert Line Feed at end of Print Buffer (Y/N)
Unattended operation (Y/N/B)
MSG-WAIT - cancel, F1 - redisplay
23.77 30x80 Ready CP80 MSG OVR CAP NUM SCRL

```

Press MSG WAIT to cancel the delete request and return to the **smterm** screen where the delete request was made.

Press F2 to confirm that this is the terminal definition that you want to delete.

## Reset Physical Device Settings from `/etc/hosts` file

When creating a terminal definition it is possible to enter a device address as a host name from the `etc/hosts` file. The IP address for that host name in the `/etc/hosts` file is stored with the terminal definition. If changes are made to a network and IP addresses are reassigned then it is necessary for the TIP/ix terminal definitions to reflect the new IP addresses. If host names are used in terminal definitions then all that is required to update the terminal definitions is to first update the `/etc/hosts` file with the new IP address assignments and then use the “RE” (reset) command of **smterm**.

The “RE” command recalculates the *normalized* address for terminal definitions. A *normalized* address is created from the device and session fields in the terminal definition. If the device is a host name from `/etc/hosts` then the normalized address is created using the IP address for that host. All IP addresses are padded with leading zeros so that all segments consist of 3 digits and the session number is padded with leading zeros to produce a 4 digit number. Then the session number is concatenated to the IP address separated by a # character.

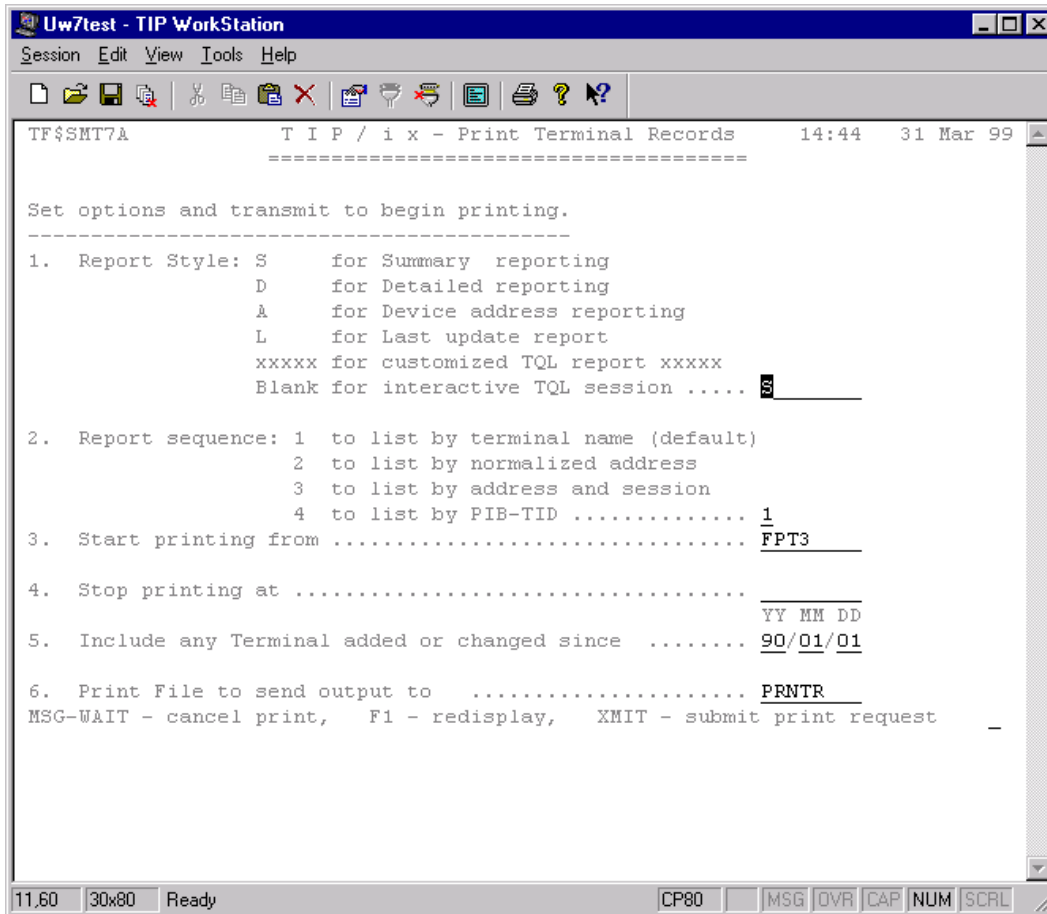
**smterm** provides two ways to reset a terminal definition.

1. From the **smterm** menu enter “RE” for the function to be performed (reset) and the name of the terminal definition for which the normalized address is to be recalculated. If no terminal name is supplied then the normalized address is recalculated for all terminal definitions and the number of terminal definitions that were reset is reported.
2. Use the list function to list terminal definitions and type “R” next to the terminal definition(s) for which you wish to have the normalized address recalculated (reset).

## Print Terminal Definitions

To print terminal definitions enter “PR” as the function to be performed (print) on the **smterm** menu display. A terminal name can also be supplied to be used as the point to start printing terminal definitions from. After pressing **XMIT**, you will be prompted to fill in the print options and report style.

When you select the Print function from **smterm** the following screen is displayed. Enter the desired print options and press transmit to print TIP/ix terminal definition information.



```

Uw7test - TIP WorkStation
Session Edit View Tools Help
TF$SMT7A T I P / i x - Print Terminal Records 14:44 31 Mar 99
=====
Set options and transmit to begin printing.

1. Report Style: S for Summary reporting
 D for Detailed reporting
 A for Device address reporting
 L for Last update report
 xxxxx for customized TQL report xxxxx
 Blank for interactive TQL session S

2. Report sequence: 1 to list by terminal name (default)
 2 to list by normalized address
 3 to list by address and session
 4 to list by PIB-TID 1

3. Start printing from FPT3

4. Stop printing at YY MM DD

5. Include any Terminal added or changed since 90/01/01

6. Print File to send output to PRNTR
MSG-WAIT - cancel print, F1 - redisplay, XMIT - submit print request

11.60 30x80 Ready CP80 MSG OVR CAP NUM SCRL

```

The following is a description of the fields in the preceding screen.

### Report style

- S** Summary report that prints a single line of information about each terminal.
- D** Detailed report that includes all information in the terminal definition.
- A** Device address report that shows for each terminal in the report the device and session at the device that resolve to the terminal definition, the normalized device address (which is a combination of device and session), the PIB-TID value and the terminal name of the terminal's bypass (\*BYP).
- L** Report date, time, and user id performing last update to the terminal definitions.

### <report name>

Any valid report name in the TQL program TIPSYST. This allows you to modify the TQL program and select your own reports. Fill in LASTUPDT for a TQL report showing the user id, date, and time of the last update for each terminal definition.



**<blank>**

Leave this entry blank to initiate an interactive TQL session with the TQL program TIPSYST

**Report sequence**

- 1 Report terminal definitions in sequence by terminal name.
- 2 Report terminal definitions in sequence by normalized address. The normalized address is derived from the device and session information. Any device that is specified as a host name is resolved to an IP address in constructing the normalized terminal address.
- 3 Report terminal definitions in sequence by address and session number. This is the address value that was filled in when defining the terminal definition and could be an IP address, host name, or device name. In this report any terminal definitions where the device was entered as an IP address will usually precede other definitions (since IP addresses begin with a digit and a typical host name or device name does not).
- 4 Report terminal definitions in sequence by the PIB-TID field.

**Start printing from this Terminal****Stop printing at this Terminal**

Range of terminal records to include in report. For example if set to "K" and "R" respectively then all terminal names in the range K to R will be included in the report.

**Change Date**

Only report terminal definitions that have been added or changed (via smterm) since this date.

**Print File**

Print file to route report to. This must be a valid TIPPRINT destination. Print files are defined by smprint and of course aliases can be set up using smsec. AUX0 can be used to direct report output to the terminal.

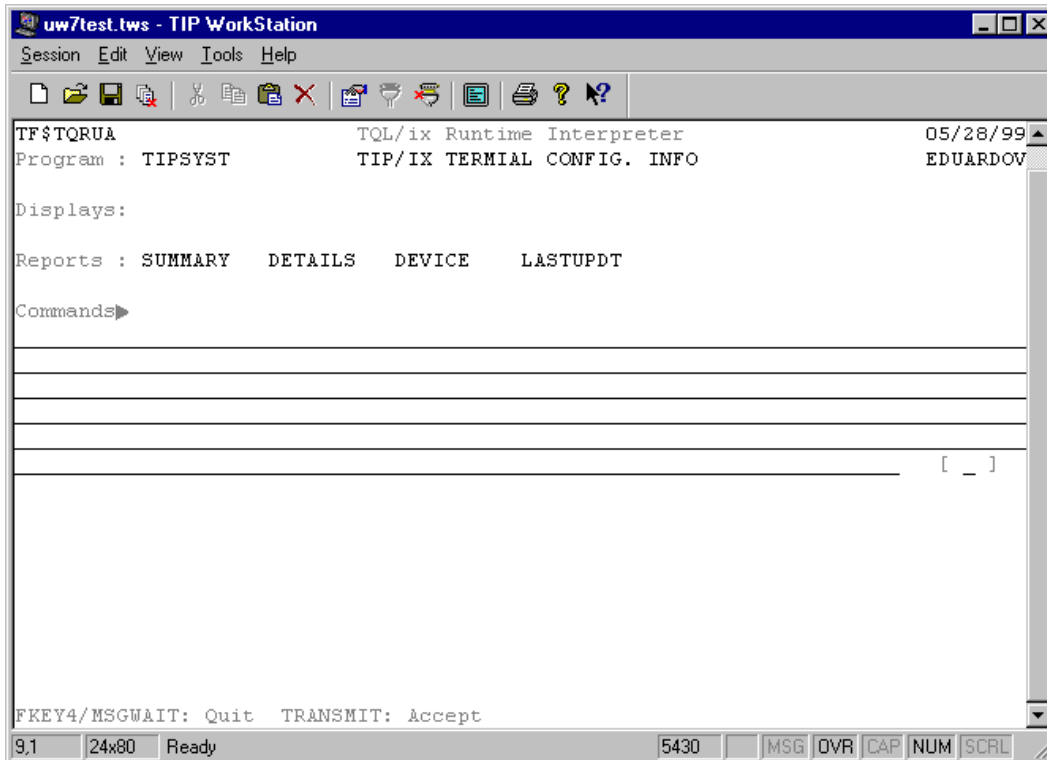
The print function is implemented by the supplied TQL program TIPSYST. If this program is not compiled and available then the print function will not work.

## Browse Terminal Definitions with TQL

The **TIPSYST** TQL program enables you to interrogate terminal definitions with ad hoc TQL queries. This TQL program can be run either:

- Directly from the TIP/ix command prompt by entering "TQL TIPSYST"
- From the **smterm** menu enter "BR" for the function to be performed (browse).

The following screen is displayed when you run **TIPSYST**.



At this point you can make any valid TQL request including:

- Request a predefined report.
- Make an ad hoc request to list (or print) fields from the terminal definitions.

See the **TIP/ix TQL Reference** for details on interacting with TQL.

The predefined reports include:

### SUMMARY

Each terminal definition is summarized into a single line on the report.

### DETAILS

For each terminal definition, the report includes all fields from the definition and any security records referencing it.

### DEVICE

Each terminal definition is summarized into a single line on the report showing the device information as it was entered and then again in a normalized format (999.999.999.999#9999). The TID and BYPASS fields are also included.

### LASTUPDT

A report showing who last updated terminal definitions (and when).

In addition to using the predefined reports, you can also perform ad hoc queries based on field settings in terminal definitions. Each terminal definition consists a record in the TIP\$SYS and the TIP\$TRM files. All information in a terminal definition appears in the TIP\$SYS record and some of this information is duplicated in TIP\$TRM file for keyed access. The following is a list of field names and the corresponding terminal definition attribute:

| Terminal Definition Attribute | TQL Field Names                            |
|-------------------------------|--------------------------------------------|
| Term Name                     | TM-NAME<br>TD-NAME (key 1)                 |
| Comments                      | TM-CMT                                     |
| PIB-TID                       | TM-PIB-TID<br>TD-PIB-TID (key 4)           |
| OS/1100 PID                   | TM-PID-1100                                |
| Sets (1st one)                | TM-SET (1)                                 |
| Sets (2nd one)                | TM-SET (2)                                 |
| IP address                    | TM-ADDRESS<br>TD-ADDRESS (prefix of key3)  |
| Host name from /etc/hosts     | TM-ADDRESS<br>TD-ADDRESS (prefix of key3)  |
| Device name from /dev         | TM-ADDRESS<br>TD-ADDRESS (prefix of key3)  |
| Session number                | TM-SESSION<br>TD-SESSION (suffix of key 3) |
| Normalized device and session | TM-ADDR-AND-SESS<br>TD-NRML-ADDR (key 2)   |

|                                                |                                |
|------------------------------------------------|--------------------------------|
| Terminal Name of MASTER                        | TM-MASTER<br>TD-MASTER (key 5) |
| BYPASS (*BYP)                                  | TM-BYPASS<br>TD-BYPASS (key 6) |
| Maximum user security                          | TM-MAX-SECUR                   |
| User must be member of group                   | TM-GROUP (1)<br>TM-GROUP (2)   |
| Allow terminal to receive unsolicited messages | TM-UNSOL-MSG                   |
| Default Lines per Page                         | TM-PRINT-LPP                   |
| Form Feed requirements                         | TM-PRINT-TOF                   |
| Precede each Form Feed with a Line Feed        | TM-PRINT-LFFF                  |
| Insert Line Feed at end of Print Buffer        | TM-PRINT-LF                    |

**Examples:**

List terminal names for terminals that can not receive unsolicited messages.

```
LIST (TM-NAME TM-ADDR-AND-SESS TM-UNSOL-MSG)
IF TM-UNSOL-MSG = "N"
```

Print a summary report by terminal name to your auxiliary printer.

```
SUMMARY ON AUX1
```

Display device report ordered by normalized address.

```
DEVICE BY KEY2
```

List terminals ordered by bypass for terminals that have a bypass.

```
LIST (TD-NAME TD-BYPASS) BY TD-BYPASS
IF TD-BYPASS > " "
```

Print detailed report by terminal name for a range of terminals and route the output to your auxiliary printer.

```
DETAILS BY KEY1 FROM "TD01" TO "TD99" ON AUX1
```

---

## smuser - Define a user id

The **smuser** transaction is used to define the attributes and system access for TIP/ix users. The information in the user id definition determine

what things (programs, files, queues) the user will be able to access from TIP/ix.

It is usual to establish a user id for each and every user of the TIP/ix system. Typically a user would have the same user id for both UNIX and TIP/ix. However, UNIX and TIP/ix each have their own files for storing their respective user id information. On UNIX the file “*/etc/passwd*” is used and on TIP/ix the user id information is stored in the “*TIP\$SYS*” file.

Normally TIP/ix would reject an attempt to run a TIP/ix shell (**tipix**) if the user id making the attempt is not defined to TIP/ix. However, if a user id of **DEFAULT** is defined then any user not defined to TIP/ix that runs a TIP/ix shell will be assigned the TIP/ix user id **DEFAULT** and the user will have all the attributes and access permissions associated with the user id **DEFAULT**.

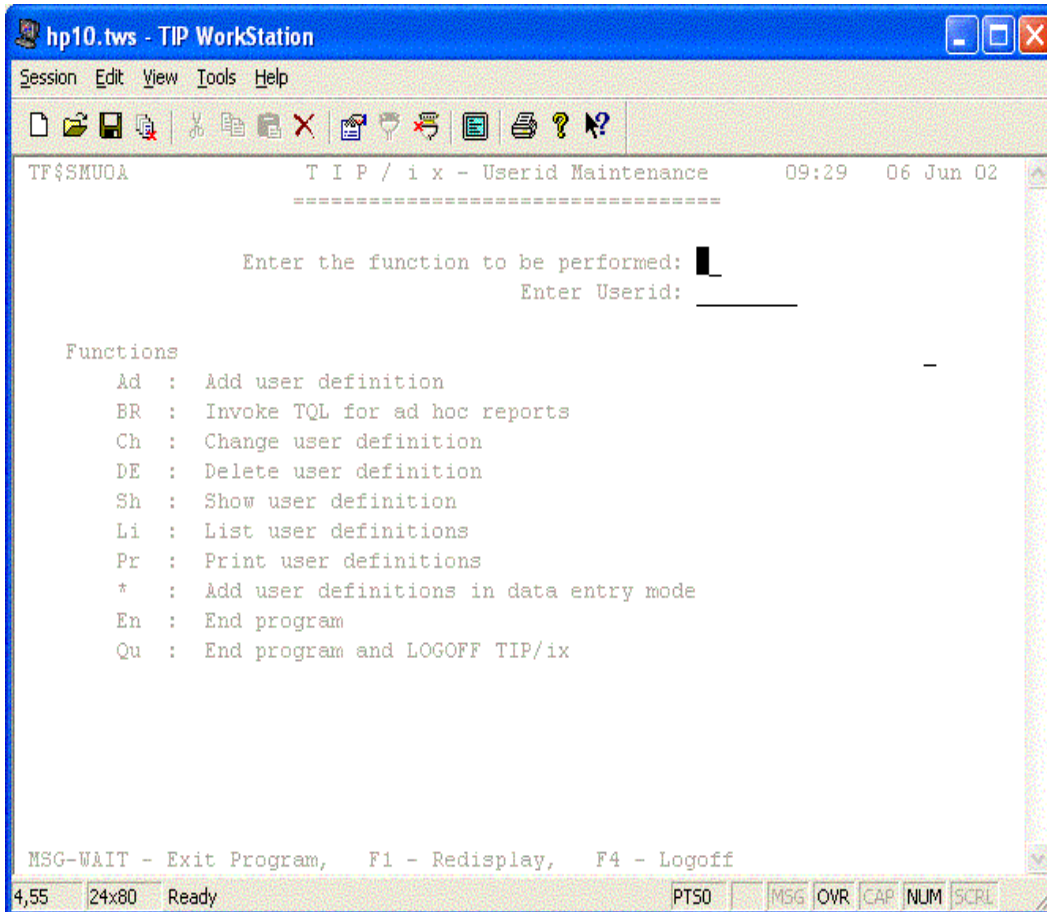
**smuser** must be run from within the TIP/ix shell (**tipix**).

**Syntax:**

**smuser** [**function**] [**user id**]

If a valid function is supplied then **smuser** will perform that function and then terminate when that function has been completed. Most functions require a user id to act upon.

If no function is supplied then **smuser** displays the following user id maintenance menu screen.



From this menu you may perform all the traditional maintenance functions on the user id definitions. You may add new user id definitions; change, delete, show or print current user id definitions and list a summary of all current user id definitions.

In the preceding screen, the function to be performed can be abbreviated to the portion shown in upper case. For example the function *add* can be entered as "A".

## Mechanics of TIP/ix security

In granting access permissions it is necessary to understand how system security is implemented on TIP/ix. There are two dimensions to security in TIP/ix, application groups and security levels. Users are granted access to groups and assigned security levels via **smuser** and **smgrpset**. Programs, files, and queues are associated with groups and assigned security levels by security entries created with **smsec**.

A TIP/ix user may only access programs, files, and queues that are defined in groups in which the user is a member. All users are automatically given access to the user id group (group name equal to the user id) and the universal group (**TIP\$\$**) Furthermore, even though the

user is a member of a group his/her access to programs, files, and queues in that group is restricted by the security level assigned to the user (within that group).

When a TIP/ix user attempts to run a program or access a file or queue, TIP/ix looks up the security definition for the item in the TIP\$SEC file. This look up follows a fixed order, known as the “*standard order of search*”. First TIP/ix looks for a definition in the user id group (group name equal to the user id), then in any of the user’s active groups and finally in the universal group, TIP\$Y\$. This order is important because TIP/ix only considers access to the first matching entry. If the user’s security level does not permit access to the first matching entry then TIP/ix will not permit the user access to the item.

It is possible to limit the scope of the “*standard order of search*” via the *Group Search* field in the user id definition. This provides a means for restricting the list of groups to search but not for changing the order in which they are searched.

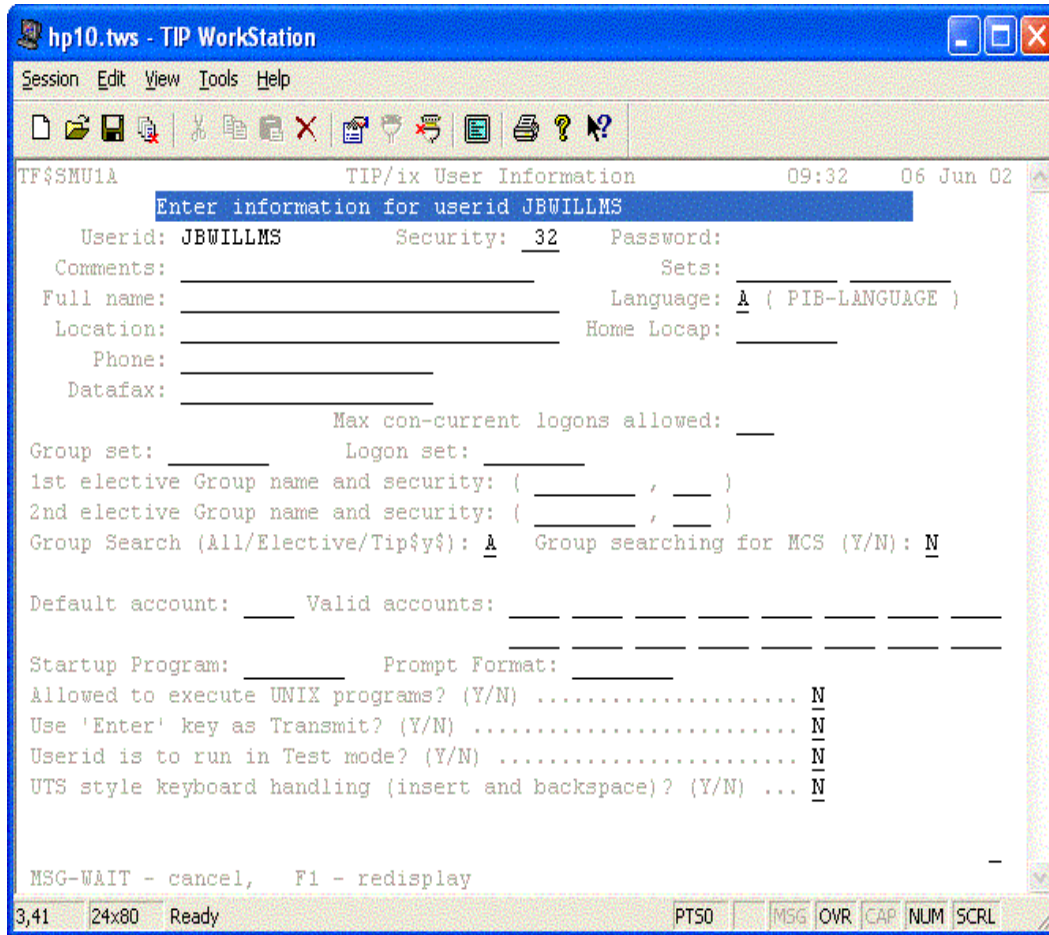
The initial list of active groups for a user consists of the user’s elective groups followed by any groups in the user’s logon set. The number of active groups is limited to 16. However, the active groups can be altered (via the **groups** transaction or by a program making a call to **TIPGRPST**) to any group that the user has been granted access; either as an elective group or as a member of a logon set or group set.

Security levels are represented as a number in the range 1 to 255 where 1 is considered the highest security level (greatest access) and 255 is considered the lowest security level (least access). So if a user’s security level (for a given group) is numerically less than or equal to the security level for an item (program, file, or queue) in a group then the user will be able to access the item.

A user is assigned a general security level which applies to the user id group, the universal group (**TIP\$Y\$**), and any groups assigned to the user without a specific security level. If a security level is assigned to an elective group or any group assigned via a group set then that security level applies to the user’s access in that group. So it is possible for a user to have a different security level in each group that the user has been granted access to.

## Add a user id

To define a new user id, enter “A” for the function to be performed (add) and the user id to add (“JBWILLMS” in our example). After pressing **XMIT**, the following screen appears to allow you to enter the user id attributes.



```

hp10.tws - TIP WorkStation
Session Edit View Tools Help
Enter information for userid JBWILLMS
 Userid: JBWILLMS Security: 32 Password:
 Comments: _____ Sets: _____
 Full name: _____ Language: A (PIB-LANGUAGE)
 Location: _____ Home Locap: _____
 Phone: _____
 Datafax: _____
 Max con-current logons allowed: ____
 Group set: _____ Logon set: _____
 1st elective Group name and security: (_____ , ____)
 2nd elective Group name and security: (_____ , ____)
 Group Search (All/Elective/Tipy): A Group searching for MCS (Y/N): N
 Default account: ____ Valid accounts: _____
 Startup Program: _____ Prompt Format: _____
 Allowed to execute UNIX programs? (Y/N) N
 Use 'Enter' key as Transmit? (Y/N) N
 Userid is to run in Test mode? (Y/N) N
 UTS style keyboard handling (insert and backspace)? (Y/N) ... N
 MSG-WAIT - cancel, F1 - redisplay
 3,41 24x80 Ready PTSO MSG OVR CAP NUM SCRL

```

At the bottom of the screen is a summary of function keys and their actions.

At any time you may enter MSG WAIT to cancel the function and return to the main menu screen.

Pressing F1 causes the screen to be re-displayed (Caution! any data entered on the screen will be lost).

Fill in the attributes for the user id and press XMIT to add the user id to the TIP\$SYS file.

The following is a description of each field in the preceding screen.

#### user id

The name by which TIP/ix knows this user. Usually, this is the user's UNIX user id converted to upper case (TIP/ix user ids are all upper case).

To start a TIP/ix session for a user id other than the logged on UNIX user id run the TIP/ix shell as "**tipix -u user id**".

#### Security

This is a number between 1 and 255 which represents the



security clearance granted to this user. The greatest access (highest level) is 1 and the least access (lowest level) is 255. The user will be able to access files, programs, and queues defined by [smsec](#) within application groups that the user has access to and that have a security level that is equal to or numerically greater than this value (the user's security level).

**Note** that this security level applies to the user id group (group name the same as the user id), the universal group **TIP\$\$**, and any group assigned to the user without a security level (for that group).

### Password

The password is used to prevent unauthorized use of the user id. When a TIP/ix shell is started and the user id is the same as the UNIX user id then a password is not required to proceed with the TIP/ix session. However, if the TIP/ix user id does not match the UNIX user id (tipix -u altuid) then TIP/ix will force the password to be entered (if one exists) before proceeding with the TIP/ix session.

This is a non-display field. Upon exiting the field TIP/ix will display the value entered and ask for confirmation to set the password to this value. After responding to this prompt you can continue entering values in the remaining fields.

Fill in with spaces (and exit the field prior to XMIT) to clear the password for a user.

### Comments

Optional information describing the user id.

**Sets** This entry is used to identify a set of records in the TIP\$\$SYS file that are in some way related. For example you could mark all TIP\$\$SYS records used in a payroll application by filling in the set name with "PAYROLL". The intention is that these records can be migrated as a set from one TIP/ix system to another using the [tippack](#) transaction.

### Full name

This information can be displayed for each user with the **users** program. The purpose of this field is to be able to relate the user id to the person using it.

### Language

Set PIB-LANGUAGE to this value for this user. TIP/ix uses PIB-LANGUAGE to resolve screen names and canned message references for its utility programs.

**Default** is "A" for American English.

Pick a value from the following list. Note that use of TIP/ix utilities will require a set of screens and messages corresponding to the language letter.

|          |                  |
|----------|------------------|
| <b>A</b> | American English |
| <b>B</b> | Canadian English |
| <b>C</b> | Canadian French  |
| <b>D</b> | Dutch            |
| <b>E</b> | UK English       |
| <b>F</b> | French           |
| <b>G</b> | German           |
| <b>H</b> | Swiss-German     |
| <b>I</b> | Italian          |
| <b>K</b> | Afrikaans        |
| <b>N</b> | Norwegian        |
| <b>P</b> | Portuguese       |
| <b>R</b> | Swiss-French     |
| <b>S</b> | Spanish          |
| <b>W</b> | Swedish          |
| <b>X</b> | Danish           |
| <b>Y</b> | Finnish          |

**Location**

Informational only. Department or location in building are suggested uses.

**Home LOCAP**

Informational only. The intention of this field is to allow the site administrator to indicate which TIP system this user normally uses (or establishes an initial TIP session with).

**Phone**

Informational only.

**Last Update**

Informational only

**Datafax**

Informational only.

**Max concurrent logons allowed:**

Specify the maximum number of times a user can logon (as himself or herself) at the same time. The **default**, blank, means users can have unlimited concurrent sessions.

**Group set:**

The name of a TIP/ix group set record (see [smgrpset](#)) that defines a list of potential application groups to which this user may belong. Users (or programs running on behalf of the user) may dynamically alter the user's current (active) group membership by calling the **groups** transaction or by issuing the appropriate TIP/ix subroutine call.

If the intention is to make the groups listed in the group set record part of the user's active groups when a TIP/ix shell is started then use the field *Logon set* instead of *Group set*.

**Logon set:**

The name of a TIP/ix group set record (see [smgrpset](#)) that is used to establish the user's active group membership when a TIP/ix shell (**tipix**) is started. When a TIP/ix shell is started the active groups for the user will consist of the *1st elective group* (if specified), the *2nd elective group* (if specified), and then the groups listed in the group set referred to by *Logon set* up to a maximum of 16 groups. TIP/ix allows a user to have active access to at most 16 elective groups. There is no limit on the number of potential groups a user may have access to (see [smgrpset](#)).

For many user ids the *1st elective group* and *2nd elective group* will be adequate to define their group membership requirements (and *Logon set* will not be required).

**1st elective Group and security**

The 1st elective group in the user's active group list when a TIP/ix session (tipix) is started. If the security level is filled in then this value will be used to determine the files, programs, and queues available to this user in this group. If the security level is left blank then the value in the Security field is used to determine the user's access in this group.

**2nd elective Group and security**

The 2nd elective group in the user's active group list when a TIP/ix session (tipix) is started. If the security level is filled in then this value will be used to determine the files, programs, and queues available to this user in this group. If the security level is left blank then the value in the Security field is used to determine the user's access in this group.

**Group Search**

Controls what groups are searched when a user accesses a file, program, or queue. This field can be used to prevent automatic access to the user id group and to override elective group access. Fill in one of the following values:

- A** Search all groups. The order of search is the user id group (group name equal to the user id), the active elective groups (as shown by [wmi](#)), the universal group (**TIP\$Y\$**).
- E** Search elective groups. The order of search is the active elective groups (as shown by [wmi](#)) and then

- the universal group (**TIP\$\$**). User can not access items defined in the user id group.
- T** Only allow the user to access files, programs, and queues in the universal group (**TIP\$\$**). User can not access items in the user id group or any of the user's active elective groups.

### Group searching for MCS

Enables or disables group searching when requests are made to access screen formats. Fill in on of the following values:

- Y** Attempts to access screen formats will be resolved by searching for the screen format names in the groups designated by the Group Search field. This facility can be used to support multiple languages. For example screen formats could be created in the group SPANISH and displayed for users that are members of the group SPANISH. The same screen names in the group TIP\$\$ could be displayed for users not in the group SPANISH.
- N** Attempts to access screen formats will be resolved by searching for the screen format names in the universal group (TIP\$).

### Default account

PIB-ACCOUNT-NUMBER is set to this value for this user.

### Valid accounts

PIB-ACCOUNT-NUMBER is set to the 1st account number in the list if a *Default account* is not specified.

### Startup Program

The name of a transaction that is to be automatically executed when the user starts a TIP/ix shell (**tipix**). When the program terminates the TIP/ix shell also terminates. This facility can be useful to limit the user's view of TIP/ix to the application area related to his/her job.

The startup program is often specified as a menu program which restricts the user to making selections from his/her application menu. The TIP/ix **menu** transaction may be a suitable program for this purpose.

### Prompt Format

The name of a screen format to be used as the TIP/ix shell prompt for this user. This facility is not currently supported.

### Allowed to execute Unix programs

Enter "Y" to allow this user to execute UNIX programs from the TIP/ix command line.

**Use 'Enter' key as Transmit**

Enter "Y" in this field to allow the Enter key to be considered the same as XMIT.

**user id is to run in test mode**

If this field is set to "Y" then the TIP/ix sessions for this user id will operate in test mode. Test mode prevents the user from updating data files that are defined with the "Record hold" field set to "T" which means hold for transaction (see smfile).

If test mode is active then whenever a transaction end (commit point) is reached TIP/ix will roll back any updates for files defined as hold for transaction. It is as if all TIP/ix transactions set the PIB-LOCK-INDICATOR to PIB-ROLLBACK at transaction end.

This field is intended to allow for testing the flow of control in programs without updating files and to allow for training of new users. Be careful though since test mode does not prevent updates (adds, and deletes) to files that are defined with "Record hold" set to "U" (update) or "Y" (yes).

The TIP/ix prompt will indicate the user is running in test mode.

**UTS style keyboard handling**

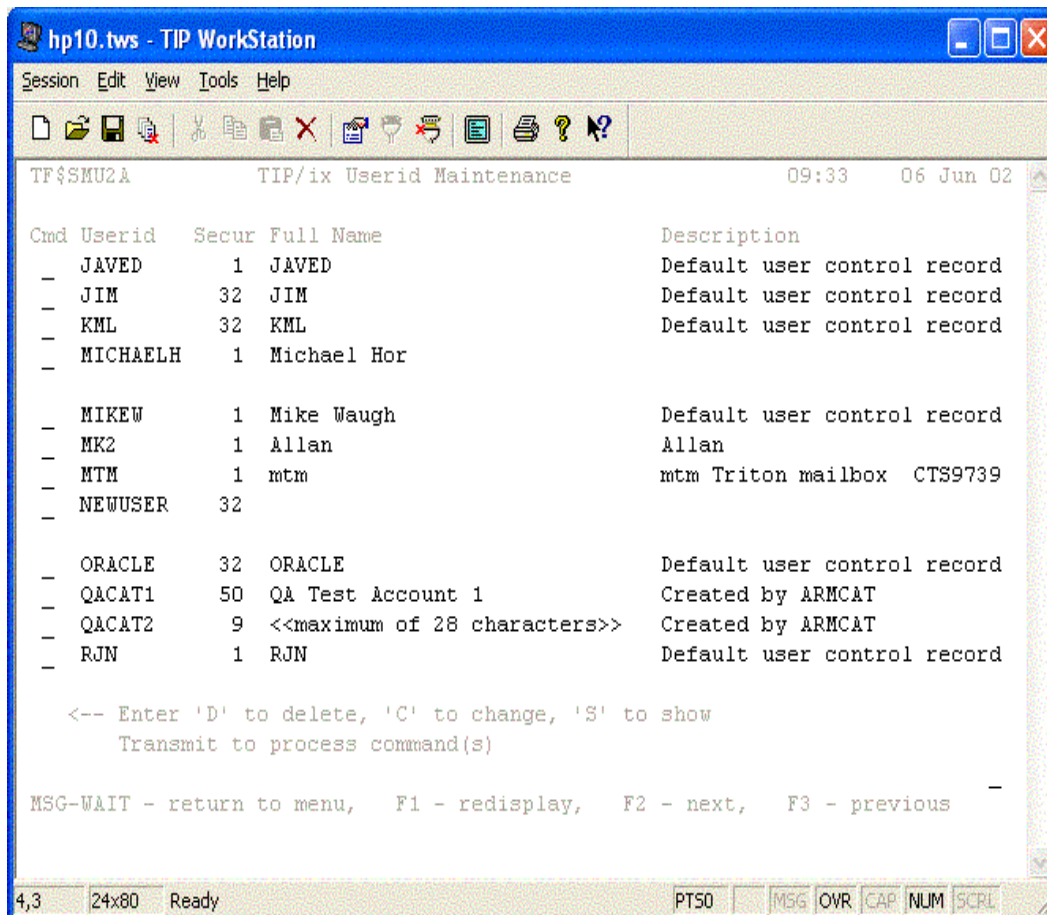
- Y** The "Insert" and "Delete" keys will operate as they would on a UTS terminal. Insert will insert a single character. Backspace will move the cursor left a single character and replace the character at the new cursor position with a space.
- N** The "Insert" and "Delete" keys will operate as they would on a computer (DOS or Windows). Insert will toggle between insert mode and type over mode. Backspace will shift a line (or field) left a single character from the cursor position overlaying the character that was previously to the left of the cursor.

**List user ids**

From the **smuser** menu enter "LI" for the function to be performed (List) and a name to use as the starting point for the list. This does not have to be a complete name. For example a name of "F" could be used to start listing user ids with a name greater than (or equal to) "F".

The list function displays a single line summary for each user id showing the user id, security level assigned to the user id, full name of the user, and description.

The following screen is displayed in response to the list command.



```

hp10.tws - TIP WorkStation
Session Edit View Tools Help
TF$SMU2A TIP/ix Userid Maintenance 09:33 06 Jun 02

Cmd Userid Secur Full Name Description
- JAVED 1 JAVED Default user control record
- JIM 32 JIM Default user control record
- KML 32 KML Default user control record
- MICHAELH 1 Michael Hor
- MIKEW 1 Mike Waugh Default user control record
- MK2 1 Allan Allan
- MTM 1 mtm mtm Triton mailbox CTS9739
- NEWUSER 32
- ORACLE 32 ORACLE Default user control record
- QACAT1 50 QA Test Account 1 Created by ARMCAT
- QACAT2 9 <<maximum of 28 characters>> Created by ARMCAT
- RJN 1 RJN Default user control record

<-- Enter 'D' to delete, 'C' to change, 'S' to show
Transmit to process command(s)

MSG-WAIT - return to menu, F1 - redisplay, F2 - next, F3 - previous
4,3 24x80 Ready PT50 MSG OVR CAP NUM SCRL

```

Press MSG WAIT to return to the main menu screen.

Press F2 to display the next screen-full of user ids.

Press F3 to display the previous screen-full of user ids.

You can enter single character commands next to the user ids and when you XMIT the commands will be processed sequentially (one at a time) from the top of the display. The commands that can be entered are:

- C** Change user id definition
- D** Delete user id definition
- S** Show user id definition

If no commands are entered and you press XMIT next to a user id on the list display then this is interpreted as a show (S) request for this user id definition.

## Display a user id Definition

**smuser** provides two ways to display a user id definition.

1. From the **smuser** menu enter “S” for the function to be performed (show) and the name of the user id definition to be displayed. (“BJONES” in our example).
2. Use the list function to list user ids and type “S” next to the user id definition(s) that you want to display.

The following screen is displayed in response to a request to display a user id definition.

```

hp10.tws - TIP WorkStation
Session Edit View Tools Help
[Icons]
TF$SMU1A TIP/ix User Information 09:34 06 Jun 02

 Userid: MIKEW Security: 1 Password:
 Comments: Default user control record Sets: TIP$SYS
 Full name: Mike Waugh Language: A (PIB-LANGUAGE)
 Location: Home Locap:
 Phone: Last Update: 01/08/13 14:28 MICHAELH
 Datafax: Last Logon: 02/06/06 09:31
 Max con-current logons allowed:
 Group set: Logon set:
 1st elective Group name and security: (,)
 2nd elective Group name and security: (,)
 Group Search (All/Elective/Tipy): A Group searching for MCS (Y/N): N

 Default account: Valid accounts:

 Startup Program: Prompt Format:
 Allowed to execute UNIX programs? (Y/N) Y
 Use 'Enter' key as Transmit? (Y/N) Y
 Userid is to run in Test mode? (Y/N) N
 UTS style keyboard handling (insert and backspace)? (Y/N) ... N

MSG-WAIT - exit, F1 - redisplay, F2 - next, F3 - previous, F4 - update
23,78 24x80 Ready PTSO MSG OVR CAP NUM SCRL

```

The fields *Last Update* and *Last Logon* show the last time the user definition was updated (and by who) and the last time the user started a TIP/ix session on this LOCAP. If the user id has never been used to access this TIP/ix LOCAP then the *Last Logon* field (heading and data) is omitted.

Press MSG WAIT to return to the **smuser** screen where the show request was made.

Press F2 to display the next user id definition.

Press F3 to display the previous user id definition.

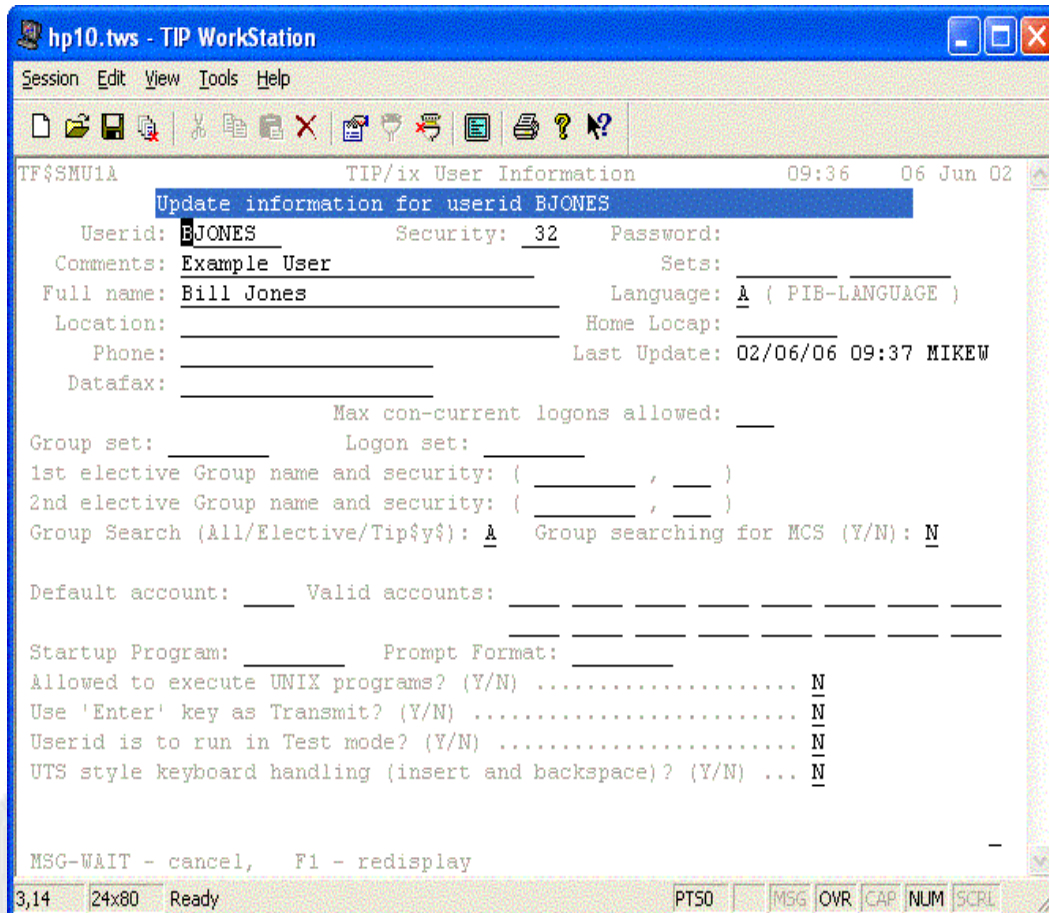
Press F4 to update the user id definition. The screen will be redisplayed with the data fields padded with underscores.

## Change a user id Definition

**smuser** provides two ways to change (or update) a user id definition.

1. From the **smuser** menu enter “CH” for the function to be performed (change) and the name of the user id definition to be changed. (“BJONES” in our example).
2. Use the list function to list user ids and type “C” next to the user id definition(s) that you want to change.

The following screen is displayed in response to a request to change a user id definition.



```

hp10.tws - TIP WorkStation
Session Edit View Tools Help
Update information for userid BJONES
 Userid: BJONES Security: 32 Password:
 Comments: Example User Sets:
 Full name: Bill Jones Language: A (PIB-LANGU&AGE)
 Location: Home Locap:
 Phone: Last Update: 02/06/06 09:37 MIKEW
 Datafax:
 Max con-current logons allowed:
 Group set: Logon set:
 1st elective Group name and security: (,)
 2nd elective Group name and security: (,)
 Group Search (All/Elective/Tipy): A Group searching for MCS (Y/N): N
 Default account: Valid accounts:
 Startup Program: Prompt Format:
 Allowed to execute UNIX programs? (Y/N) N
 Use 'Enter' key as Transmit? (Y/N) N
 Userid is to run in Test mode? (Y/N) N
 UTS style keyboard handling (insert and backspace)? (Y/N) ... N
 MSG-WAIT - cancel, F1 - redisplay
3,14 24x80 Ready PTSO MSG OVR CAP NUM SCRL

```

Press MSG WAIT to cancel the update and return to the **smuser** screen where the change request was made.

Fill in any desired changes and press XMIT to change the user id definition. For a description of the fields on the user id definition refer to the section on [“Add a user id”](#).

If the *userid* field is changed then the original user id definition is left unchanged and a new user id definition is added.



## Delete a user id Definition

**smuser** provides two ways to delete a user id definition.

1. From the **smuser** menu enter “DE” for the function to be performed (delete) and the name of the user id definition to be deleted. (“BJONES” in our example).
2. Use the list function to list user ids and type “D” next to the user id definition(s) that you want to delete.

The following screen is displayed in response to a request to delete a user id definition.

```

hp10.tws - TIP WorkStation
Session Edit View Tools Help
[Icons]
TF$SMU1A TIP/ix User Information 09:37 06 Jun 02
Press F2 to DELETE userid BJONES
 Userid: BJONES Security: 32 Password:
 Comments: Example User Sets:
 Full name: Bill Jones Language: A (PIB-LANGUAGE)
 Location: Home Locap:
 Phone: Last Update: 02/06/06 09:37 MIKEW
 Datafax:
 Max con-current logons allowed:
 Group set: Logon set:
 1st elective Group name and security: (,)
 2nd elective Group name and security: (,)
 Group Search (All/Elective/Tipy): A Group searching for MCS (Y/N): N
 Default account: Valid accounts:
 Startup Program: Prompt Format:
 Allowed to execute UNIX programs? (Y/N) N
 Use 'Enter' key as Transmit? (Y/N) N
 Userid is to run in Test mode? (Y/N) N
 UTS style keyboard handling (insert and backspace)? (Y/N) ... N
 MSG-WAIT - cancel, F1 - redisplay
23,78 24x80 Ready PTSO MSG OVR CAP NUM SCRL

```

Press MSG WAIT to cancel the delete request and return to the **smuser** screen where the delete request was made.

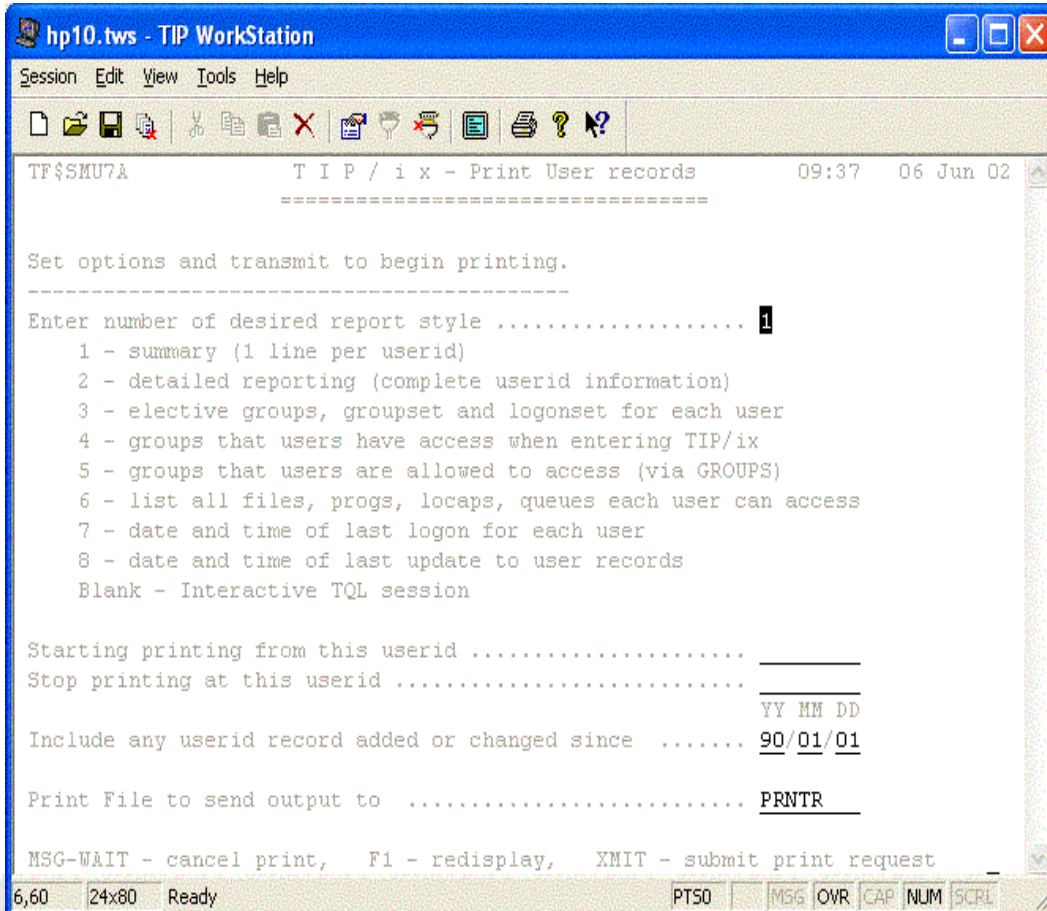
Press F2 to confirm that this is the user id definition that you want to delete.

If the user id record is deleted successfully then you will be asked if you would like delete the user’s TIP/ix mailbox and function keys. Responding with a “Y” to this question will result in the [scratch](#) being called to remove these TIP/ix *dynamic* files (from the directory (\$TIPROOT/tipfiles/dynamic)).

## Print user id Definitions

To print user ids enter “PR” as the function to be performed (print) on the **smuser** menu display. A user id name can also be supplied to be used as the point to start printing user id definitions from. After pressing **XMIT**, you will be prompted to fill in the print options and report style.

When you select the Print function from **smuser** the following screen is displayed. Enter the desired print options and press transmit to print TIP/ix user id definitions.



The following is a description of the fields in the preceding screen.

### Report style

- 1 Summary report that prints a single line of information about each user id.
- 2 Detailed report that includes all information in the user id definitions.
- 3 Report summarizing the access permissions for each user id. A single line per user id consisting of user id, security, 1st elective group and security, 2nd elective group and security, logon set and

- group set. An additional line will be printed for each group set in a group set chain.
- 4 Report showing the active groups and security level(s) for each user upon starting TIP/ix. If a user is assigned a specific security within a group that value is shown next to the group name. For example "MIS,29".
  - 5 Report showing for each user id all groups (and security levels within those groups) that the user may gain access to. This report includes all active groups at logon (report style 4) and also any groups from the user's group set. Therefore, this report is a superset of report style 4.
  - 6 For each user id in the report list all items (programs, files, queues) that the user could potentially access in all groups that the user has access to. So for each group that the user has access to all items with a security level less than or equal to the user's security level for that group are included in the report.
  - 7 Report showing system usage information, such as, total number of TIP/ix sessions, date and time last TIP/ix session was initiated, date of any future TIP/ix mail.
  - 8 Report who last updated each user id record and when each user id was last updated.
- <blank>**  
Leave this entry blank to initiate an interactive TQL session with the TQL program TIPSYSU

**Start printing from this user id****Stop printing at this user id**

Range of user id records to include in report. For example if set to "K" and "R" respectively then all user ids in the range K to R will be included in the report.

**Change Date**

Only report user id definitions that have been added or changed (via smuser) since this date.

**Print File**

Print file to route report to. This must be a valid TIPPRINT destination. Print files are defined by smprint and of course aliases can be set up using smsec. AUX0 can be used to direct report output to the terminal.

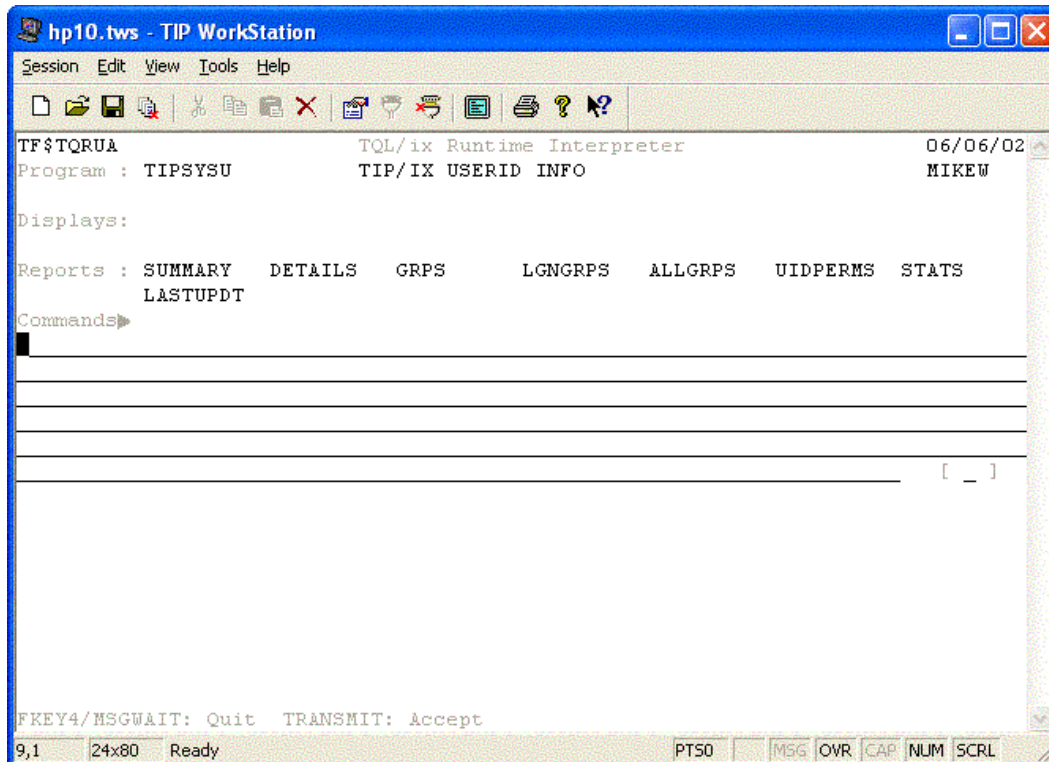
The print function is implemented by the supplied TQL program TIPSYSU. If this program is not compiled and available then the print function will not work.

## Browse user id Definitions with TQL

The **TIPSYSU** TQL program enables you to interrogate user id definitions with ad hoc TQL queries. This TQL program can be run either:

- Directly from the TIP/ix command prompt by entering "TQL TIPSYSU"
- From the **smuser** menu enter "BR" for the function to be performed (browse).

The following screen is displayed when you run **TIPSYSU**.



At this point you can make any valid TQL request including:

- Request a predefined report.
- Make an ad hoc request to list (or print) fields from the user id definitions.

See the **TIP/ix TQL Reference** for details on interacting with TQL.

The predefined reports include:

### SUMMARY

Each user id definition is summarized into a single line on the report.

### DETAILS

For each user id definition the report includes all fields from the definition.

### GRPS

For each user id the report includes the security level, groups, logon group set, and group set fields.

### LGNGRPS

For each user id the report includes the general security level at logon, all groups that the user can access upon logging on, and any security level restrictions within those groups.

### ALLGRPS

For each user id the report includes the general security level for the user and all groups that the user is eligible to access and any security level restrictions within those groups.

### UIDPERMS

For each user id the report displays the groups that the user can access and the files, queues, and programs that the user can access (based on the user's security level) from each of these groups.

### STATS

For each user id definition the report includes a one line summary of some statistics (including number of times the user logged on, and the last date and time that the user logged on).

### LASTUPDT

A report showing who last updated user id definitions (and when).

In addition to using predefined reports, you can also perform ad hoc queries based on field settings in user id definitions. The following is a list of TQL field names and the corresponding user id Definition Attribute:

| user id Definition Attribute | TQL Field Names |
|------------------------------|-----------------|
| user id                      | UR-UID          |
| Security                     | UR-SECUR        |
| Comments                     | UR-CMT          |
| Sets (1st one)               | UR-SET (1)      |
| Sets (2nd one)               | UR-SET (2)      |
| Full name                    | UR-NAME         |
| Language                     | UR-LANG         |
| Location                     | UR-LOCN         |

|                                  |                |
|----------------------------------|----------------|
| Home LOCAP                       | UR-LOCAP       |
| Phone                            | UR-PHONE       |
| Last Update                      | UR-UPD-USER    |
|                                  | UR-UPD-DATE    |
|                                  | UR-UPD-TIME    |
| Datafax                          | UR-FAX         |
| Last Logon                       | UR-LDATE       |
|                                  | UR-LTIME       |
| Max con-current logons           | UR-USERS       |
| Group set                        | UR-GRPST       |
| Logon set                        | UR-LGNST       |
| 1st elective Group name          | UR-GRP1        |
| 1st elective Group security      | UR-SEC1        |
| 2nd elective Group name          | UR-GRP2        |
| 2nd elective Group security      | UR-SEC2        |
| Group Search                     | UR-NSRCH       |
|                                  | UR-GSRCH       |
| Group searching for MCS          | UR-MSRCH       |
| Default account                  | UR-DACCT       |
| Valid accounts                   | UR-ACCT (1-16) |
| Startup Program                  | UR-TCP         |
| Prompt Format                    | UR-MENU        |
| user id Expires                  | UR-XPIRY       |
| Allowed to execute UNIX programs | UR-OPSYS       |
| Use 'Enter' key as Transmit      | UR-ENTER       |
| user id is to run in Test mode   | UR-TESTM       |
| UTS style keyboard handling      | UR-UTSKEYS     |

**Examples:**

List user id that can run UNIX programs from the TIP/ix shell.

```
LIST (UR-UID UR-CMT UR-OPSYS)
IF UR-OPSYS = "Y"
```

List user id definitions that have a security level less than 10

```
SUMMARY IF UR-SECUR < 10
```

Print a summary report showing groups and group sets assigned to users for those users that have some groups assigned. List

```
GRPS IF UR-GRP1 > " " OR IF UR-GRP2 > " "
OR UR-LGNST > " " OR IF UR-GRPST > " "
ON AUX1
```

Print a report showing all files, queues, and programs that the user id "FRED" can access.

```
UIDPERMS FROM "FRED" TO "FRED" ON AUX1
```

Display a report of users that have not logged on since Jan 1, 1999.

```
STATS IF UR-LDATE < "960101"
```

---

## sorter - Sort TIP/ix Edit Buffer

The **sorter** transaction is a utility that sorts the contents of a TIP/ix edit buffer.

Data is sorted according to the standard ASCII collating sequence (according to the internal binary representation of each character).

This program is utilized by other TIP/ix transaction programs and is not often used directly from the TIP/ix command line. User programs may TIPSUB to **sorter**, passing the necessary command line parameters in the CDA.

### Syntax:

```
sorter[/Q][H|h] grp,buffer [,begin] [,end] [,col] [,dir]
```

### Where:

- /Q** Quiet. Suppress output messages.
- [H|h]** This option will invoke the usage information.
- grp** The name of the group to which the edit buffer belongs.
- buffer** The name of the edit buffer.
- begin** The line number where sorting is to begin (inclusively).  
Default: line 1.
- end** The line number where sorting is to end (inclusively).  
Default: last line of buffer.
- col** The starting column of the data to use as a sort key.  
Default: column 1.

**dir** The direction of sorting. May be either "A" (ascending) or "D" (descending). Default: "A" (ascending).

**Example:**

Sort an edit buffer named "table" in the group "edp" into ascending sequence according to data starting in column 10.

```
sorter edp/table,,,10
```

**Additional Considerations:**

Records with identical sort keys will not necessarily remain in their original order. The **sorter** program may unpredictably interpret data that is in packed or binary format. For example, positive numeric packed fields may have different signs depending on which compiler you use.

## STARTUP - Startup Processing

The STARTUP transaction is a mechanism whereby the system administrator can schedule one or more transactions that are to be run when the TIP/ix system is initially started.

The STARTUP transaction is simply a TIP/ix script file named *STARTUP* which is stored in the TIPROOT directory. This script file may contain several transactions to be scheduled for execution when the TIP/ix system is started. Each line in the script file will be executed as if the line was a standard TIP/ix command line.

If the STARTUP script file exists, it will be executed after the TIP/ix system has been started, and will execute in parallel with other transactions which may be started by users.

All transaction programs executed in the script file will be running in the background via TIPFORK. Thus any transaction that the STARTUP script performs *must* be capable of running in the background (that is, it should not solicit screen input). STARTUP uses the user id STARTUP.

## status - Display System Statistics

The **status** program displays information about the running TIP/ix system.

**Syntax:**

```
status cmd [-p printdest] [parameters]
```

**cmd** is the status command to be performed for the following list:

- a** Print the following reports to stdout: system status, transactions, files, and active keys.



**c [[\*]prtdest1] ... [[\*]prtdest7]**

Print a report of TIP/ix print destinations (as defined with smprint).

If no filters are supplied, all TIP/ix print destinations are included in the report.

**d [[\*]user] [[\*]catalog] [[\*]file]**

Print dynamic-file report. The parameters specify the dynamic files to be included in the report.

If no filters are supplied, all dynamic files are listed.

**e [[\*]user [[\*]buf]]**

Print edit-buffer report. The parameters specify the edit buffers to be included in the report.

If no filters are supplied, all edit buffers are listed.

**f[mod] [[\*]file1]...[[\*]file7]**

Print file report. The parameters specify the files to be included in the report.

**Note:** The file definition names specified here are defined with smfile. The TIPFCS names (defined by smsec) which applications use to access the files may be different.

The following modifiers are supported:

- b** Sort by usage (most I/O first).
- n** Sort by name.
- d** Sort by path.
- s** Sort by the TIPFCS server number. Only files that have activity will be included.
- c** Display only closed files.
- h** Print help for command syntax.

**i[mod] [[\*]file1]...[[\*]file7] [-s#]**

Print abbreviated file (I/O) report. The parameters specify the files to be included in the report.

**Note:** The file definition names specified here are defined with smfile. The TIPFCS names (defined by smsec) which applications use to access the files may be different.

The following modifiers are supported:

- u** Sort by usage (most I/O first).
- n** Sort by name.
- d** Sort by path.
- s** Sort by the TIPFCS server number. Only files that have activity will be included.
- c** Display only closed files.

**k [f=filename | u=user | s=segment ]**

Print active record Key report.

**l** Print LOCAP information report.

**m** Print Memory report.

**q** Print Queue report. Shows the status of the TIP queues.

- r[n]** Print information about transaction programs and database interface server processes that have been Run since the TIP/ix system was last started up. Report average response time per transaction. If n is specified, the programs are sorted by name. Otherwise, the programs are sorted by usage (most used first).
- ra** Print information about transaction programs and database interface server processes that have been Run since the TIP/ix system was last started up. Report average I/O per transaction processed.
- s** Print System Status report.
- t** Print Transaction response-Time log. This is a rolling log of response times for the last 24 hours (less — if you just started TIP/ix). You can specify the interval size with the RESPLOG= parameter in the tipix.conf file.
- u** Shows the status of TIP/ix UNIX message queues.
- x[n]** Same as r, except that only programs and database server processes with active or waiting threads are displayed.

**[\*]file1...[\*]file7**

Optional parameters specifying (when applicable) up to 7 prefixes or identifiers for use by the command. For example, "status l \*TIP" displays the abbreviated I/O report for all files with names beginning with "TIP" (the "l" command treats the additional parameters as file name prefixes). If no prefixes or identifiers are supplied, the command includes information about all items within the scope of the command (for example, all files or all LOCAPs).

**-p printfil**

Print the report to the specified file. The filename is translated to lowercase and created in the current directory. The filename is limited to 8 characters.

If this parameter is not specified, the report goes to stdout.

If this option is used from the SHUTDOWN or STARTUP files (see SHUTDOWN and STARTUP in this manual), the file will be created in the \$TIPROOT directory.

The status utility has the "set group ID" attribute set so the \$TIPROOT directory must have group permission for both reading and writing, and the group ownership of \$TIPROOT must be the TIP/ix administrative group (see tipinstal in Installations and Operations manual).

- s#** Only include files for FCS server specified by #. Only files that have activity will be included.

|

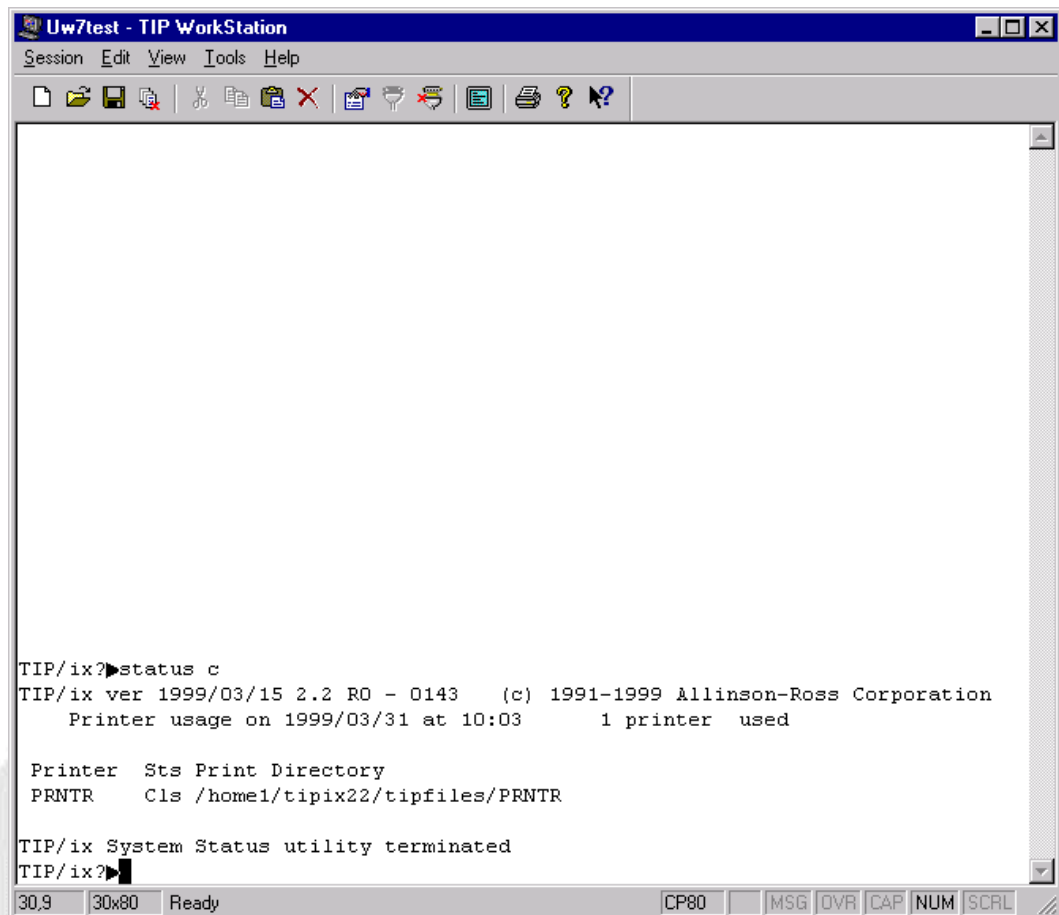
If the status utility is run from the Unix shell, the \* must be escaped. For example, when using the Korn Shell, precede the asterisk with a backslash to prevent the shell from processing the asterisk.

```
status e \<*ACT
```

## c - Printer file report

The **status "c"** command displays information about TIP/ix printer files.

### Example of status c output



```
Uw7test - TIP WorkStation
Session Edit View Tools Help
[Icons]

TIP/ix?▶status c
TIP/ix ver 1999/03/15 2.2 R0 - 0143 (c) 1991-1999 Allinson-Ross Corporation
Printer usage on 1999/03/31 at 10:03 1 printer used

Printer Sts Print Directory
PRNTR Cls /home1/tipix22/tipfiles/PRNTR

TIP/ix System Status utility terminated
TIP/ix?▶
30,9 30x80 Ready CP80 MSG DVR CAP NUM SCRL
```

## d - Dynamic file report

The **status "d"** command displays information about TIP/ix dynamic files. Dynamic files may be created on demand by TIP/ix transaction programs.

TIP/ix dynamic file names have three components of 8 bytes (user, catalog, and file). You can specify TIP/ix style prefixes for any component to restrict which dynamic files are listed.

### Example: status d

```

Uw7test - TIP WorkStation
Session Edit View Tools Help
[Icons]

TIP/ix?▶status c
TIP/ix ver 1999/03/15 2.2 R0 - 0143 (c) 1991-1999 Allinson-Ross Corporation
Printer usage on 1999/03/31 at 10:03 1 printer used

Printer Sts Print Directory
PRNTR Cls /home1/tipix22/tipfiles/PRNTR

TIP/ix System Status utility terminated
TIP/ix?▶status d
TIP/ix ver 1999/03/15 2.2 R0 - 0143 (c) 1991-1999 Allinson-Ross Corporation
TIP/ix System Dynamic Files

User-Id Catalog Filename #chars Last Modified
MAILY MAILBOX MICHAELH 2048 Feb 29 12:28:47 2000
MAILY MAILBOX TIPIXUSR 2048 Mar 1 23:00:25 1998
MAILY MAILBOX SCOTTC 2048 Mar 30 09:43:56 1999
MAILY MAILBOX JDOE 4608 Mar 30 10:41:48 1999
MAILY MAILBOX BJONES 6144 Mar 30 10:44:01 1999
MAILY MAILLIST MAILLIST 2048 Mar 30 10:52:27 1999

TIP/ix System Status utility terminated
TIP/ix?▶
30.9 30x80 Ready CP80 MSG OVR CAP NUM SCRL

```

### Example:

To display all dynamic files where the third component of the dynamic file name is "KEYS":

```
status d * * *KEYS
```

## e - Edit buffer report

The **status "e"** command displays information about TIP/ix edit buffers. Edit buffers may be created on demand by TIP/ix transaction programs; the TIP/ix editor, **fse**, is usually the program that creates and disposes of edit buffers.

TIP/ix edit buffer names have two components of 8 bytes: user (or group), and buffer. You can specify TIP/ix style prefixes for any component to restrict which edit buffers are listed.

**Example: status e**

```

TIP/ix?>status e
TIP/ix ver 1999/03/15 2.2 R0 - 0143 (c) 1991-1999 Allinson-Ross Corporation
TIP/ix System Edit Buffers

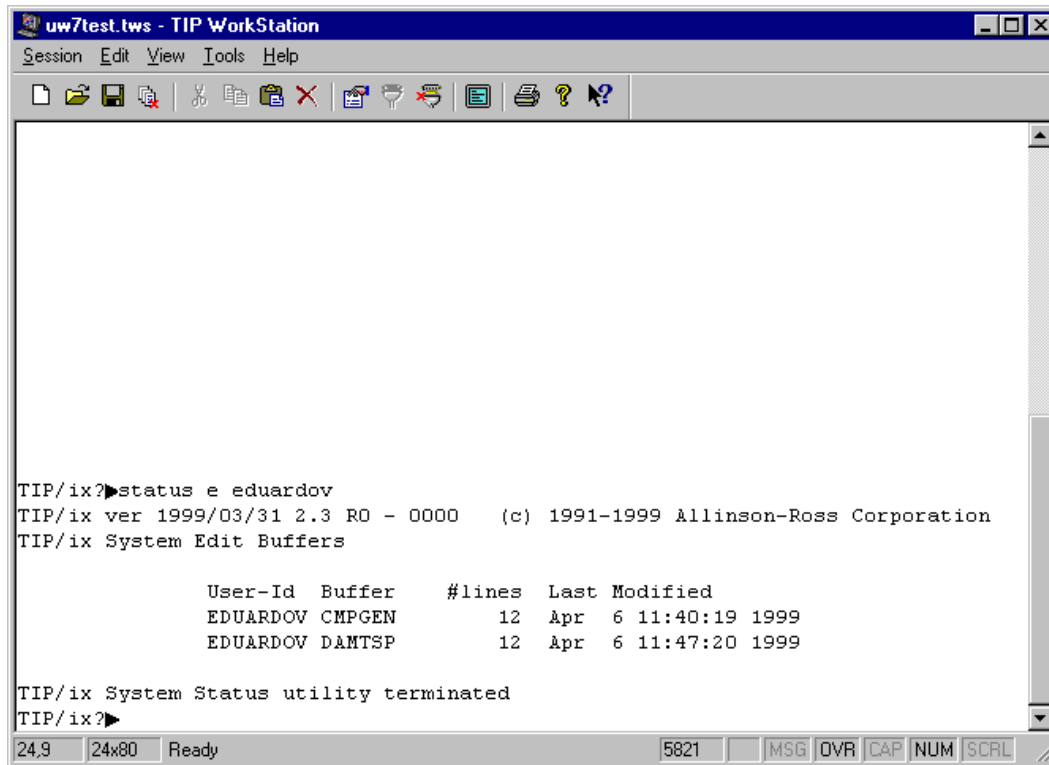
 User-Id Buffer #lines Last Modified
MICHAELH EDITBUF 31 Mar 4 10:39:41 2000
MICHAELH TQLTSP 48 Mar 6 16:29:25 2000
SCOTTC SCOTT 1 Mar 29 09:25:28 1999
SCOTTC X-REFRNC 324 Mar 29 09:54:39 1999
SCOTTC P9551 1337 Mar 30 05:52:55 1999
SCOTTC CORE 409 Mar 29 10:43:05 1999
SCOTTC LOG 180 Mar 29 11:19:35 1999
SCOTTC PASSWD 28 Mar 30 09:05:37 1999
SCOTTC TEST 1 Mar 31 06:54:53 1999

TIP/ix System Status utility terminated
TIP/ix?>

```

30.9 30x80 Ready MSG OVR CAP NUM SCRL

### Example: for specific groups/users



```

uw7test.tws - TIP WorkStation
Session Edit View Tools Help

TIP/ix?▶status e eduardov
TIP/ix ver 1999/03/31 2.3 R0 - 0000 (c) 1991-1999 Allinson-Ross Corporation
TIP/ix System Edit Buffers

 User-Id Buffer #lines Last Modified
 EDUARDOV CMPGEN 12 Apr 6 11:40:19 1999
 EDUARDOV DAMTSP 12 Apr 6 11:47:20 1999

TIP/ix System Status utility terminated
TIP/ix?▶
24.9 24x80 Ready 5821 MSG DVR CAP NUM SCRL

```

### Example: for specific edit buffers

To display all edit buffers with a name beginning with "FOO" (in any group):

```
status e * *FOO
```

## f - File report

The **status "f"** command displays detail information about files accessed through TIP/ix. An abbreviated version of this report can be obtained by using the status "i" command.

### Example of status f output

```

Uw7test - TIP WorkStation
Session Edit View Tools Help
[Icons]

TIP/ix?▶status f
TIP/ix ver 1999/03/15 2.2 R0 - 0143 (c) 1991-1999 Allinson-Ross Corporation

File usage on 1999/03/31 at 10:06 30 files used

File Type Recsz #I/Os #write Usr Sts Srv Complete file name
BMKACCT ISAM 34 0 0 0 0 Opn 2 /home1/tipix22/tipfiles/inv
 Key Loc Len Dup Chg #read %
 1 0 4 N N 0 0
 2 4 16 Y Y 0 0
 3 20 2 Y Y 0 0

 I/O Q Server SETL swaps
 46 tipfcs 0

DAMINV DAM 64 0 0 0 0 Cls 2 /home1/tipix22/tipfiles/daminv
 I/O Q Server SETL swaps
 46 tipfcs 0

INV ISAM 34 0 0 0 0 Cls 2 /home1/tipix22/tipfiles/inv
 Key Loc Len Dup Chg #read %
 1 0 4 N N 0 0
 2 4 16 Y Y 0 0
 3 20 2 Y Y 0 0

 I/O Q Server SETL swaps
 46 tipfcs 0

Continue? ▶Yes ▶No
30.16 30x80 Ready [MSG] [OVR] [CAP] [NUM] [SCRL]

```

## h - Command Help

The **status "Help"** command displays a summary of command syntax for the **status** program.

### Example of status help output

```

Uw7test - TIP WorkStation
Session Edit View Tools Help

TIP/ix?>status help
TIP/ix ver 1999/03/15 2.2 R0 - 0143 (c) 1991-1999 Allinson-Ross Corporation
Usage: status cmd [-p dest] [file1 [file2 [...]]]

Command To display
A - TIP/ix all reports
C - TIP/ix printers
D - TIP/ix dynamic files
E - TIP/ix edit buffers
F[N|U|D|S]
 - TIP/ix detailed file I/O statistics
H - this Help information
I[N|U|D|S]
 - TIP/ix file I/O statistics
K - the FCS Key Holding table
L - TIP/ix locap information
M - TIP/ix user memory information
Q - TIP/ix queue information
R - TIP/ix transaction program information
S - TIP/ix system information
T - TIP/ix system response time log
U - TIP/ix message queue usage information
X - active reusable transactions & TIP/dbi servers

commands F and I accept an optional letter N|U|D|S which
allows the information to be sorted by name(default),
number of times used, device and by server number.

Continue? >Yes No
30.16 30x80 Ready MSG OVR CAP NUM SCRL

```

### i - Abbreviated I/O report

The **status "i"** command displays an abbreviated File I/O report.



### Example of status i output

```

Uw7test - TIP WorkStation
Session Edit View Tools Help
TIP/ix?▶status i
TIP/ix ver 1999/03/15 2.2 R0 - 0143 (c) 1991-1999 Allinson-Ross Corporation

File usage on 1999/03/31 at 10:06 30 files used

File Type Recsz #I/Os #write Usr Sts Srv Complete file name
BMKACCT ISAM 34 0 0 0 Opn 2 /home1/tipix22/tipfiles/inv
DAMINV DAM 64 0 0 0 Cls 2 /home1/tipix22/tipfiles/daminv
INV ISAM 34 0 0 0 Cls 2 /home1/tipix22/tipfiles/inv
INVEN ISAM 64 0 0 0 Off - /home1/tipix22/tipfiles/inven
ORD ISAM 28 0 0 0 Cls 2 /home1/tipix22/tipfiles/ord
QAFILE1 SAM 0 0 0 0 Cls 2 /home1/tipix22/tipfiles/qa1111
QAFILE2 SAM 0 0 0 0 Cls 2 /home1/tipix22/tipfiles/QAFILE2
QAFILE3 SAM 0 0 0 0 Cls 2 /tmp/qafile3
QAFILE4 SAM 0 0 0 0 Cls 2 /home1/tipix22/tipfiles/QAFILE4
QAINVEN RDBMS 0 0 0 0 Cls 0 QAINVEN
QALIB LIB 0 0 0 0 2 /usr/local/qatools/help/
QATQLDNL ISAM 44 0 0 0 Cls 2 ~home1/tipix22/tipfiles/QATQLDNL
QATQLHDR ISAM 16 0 0 0 Cls 2 ~home1/tipix22/tipfiles/qatqlhdr
QATQLTST ISAM 156 0 0 0 Cls 2 ~home1/tipix22/tipfiles/qatqltst
SAMINV SAM 64 0 0 0 Cls 2 /home1/tipix22/tipfiles/saminv
TIP LIB 0 0 0 0 2 ~e1/tipix22/tipfiles/./include/
TIP$MCS ISAM 2628 0 0 0 Opn 1 /home1/tipix22/tipfiles/tipmcs
TIP$MNU ISAM 511 0 0 0 Cls 2 /home1/tipix22/tipfiles/tipmnu
TIP$MSG ISAM 127 774 0 0 Opn 1 /home1/tipix22/tipfiles/tipmsg
TIP$QUE DAM 2048 292 16 0 Opn 2 /home1/tipix22/tipfiles/tipque
TIP$SEC ISAM 80 1340 8 0 Opn 1 /home1/tipix22/tipfiles/tipsec

Continue? ▶Yes ▶No
30.16 30x80 Ready MSG OVR CAP NUM SCRL

```

The "Sts" (status) column may contain the following values:

| Value | Description                                                                                                     |
|-------|-----------------------------------------------------------------------------------------------------------------|
| Cls   | Closed. TIP/ix has not established access to the file. No one is using the file.                                |
| Off   | Off-line. The file is currently FCLOSEd and unavailable to TIP/ix users.                                        |
| Opn   | Open. TIP/ix has established access to the file, but this doesn't imply that anyone is actually using the file. |
| Pnd   | Pending. The file is pending off-line status. As soon as there are no users, the file will be set to "Off".     |

### k - Active record key report

The **status** "k" command displays information about record locks that are currently being maintained by TIP/ix.

### Example of status k output

```

Allross.tws - TIP WorkStation
Session Edit View Tools Help

TS22?>status k
TIP/ix ver 1999/05/18 2.3 R0 - 0008 (c) 1991-1999 Allinson-Ross Corporation

TIP/ix Key Holding table contents on 1999/05/28 at 16:24.
Transactions = 2, Most Transactions = 3
Keys Held = 4, Most Keys Held = 20

User Tran Seg * Server QBL Transaction ID
 File Lock Adrs Rec # Key (1st 16 bytes)

ALLANR KEY1 33 * tipfcs TS22 121:0000085
 TSPFILE 0 6 FOR00000
 TSPFILE 0 4 DELO0000

ALLANR KEY1 25 * tipfcs TS22 121:0000082
 TSPFILE 0 3 BET00000
 TSPFILE 0 2 ALP00000

TIP/ix System Status utility terminated
TS22?>
24.7 24x80 Ready MSG DVR CAP NUM SCRL

```

The f, u, and s switches allow filtering on the basis of the filename, user name, and segment number.

If name is specified without a switch, name is taken to be the file name. For example: **status k name**

The u switch must be specified to filter data on a user basis.

The s switch must be specified to filter data on a segment number basis.

## I - LOCAP information report

The **status "I"** command displays information about TIP/ix LOCAP connections (that is: connections with other TIP systems).

### Example of status I output

```

TIP/ix?▶status I
TIP/ix ver 1999/03/15 2.2 R0 - 0143 (c) 1991-1999 Allinson-Ross Corporation
Locap usage on 1999/03/31 at 10:07

Current Locap: [UW7TEST]

Open sessions
Locap State Type In Out Sec Current Most Messages Remote File
 Open Max In Out In Out In Out Rcvd Sent Rcvd Sent
TS22 Disconnect TIP/ix Y Y TECH 0 0 0 0 1
UW7TEST Connected SELF Y Y TECH 0 0 0 0

TIP/ix System Status utility terminated
TIP/ix?▶

```

#### Messages Rcvd

Count of messages received via TCP/IP from the other system.

#### Messages Sent

Count of messages sent via TCP/IP to the other system.

#### Remote File Rcvd

Count of remote TIPFCS requests received from the other system

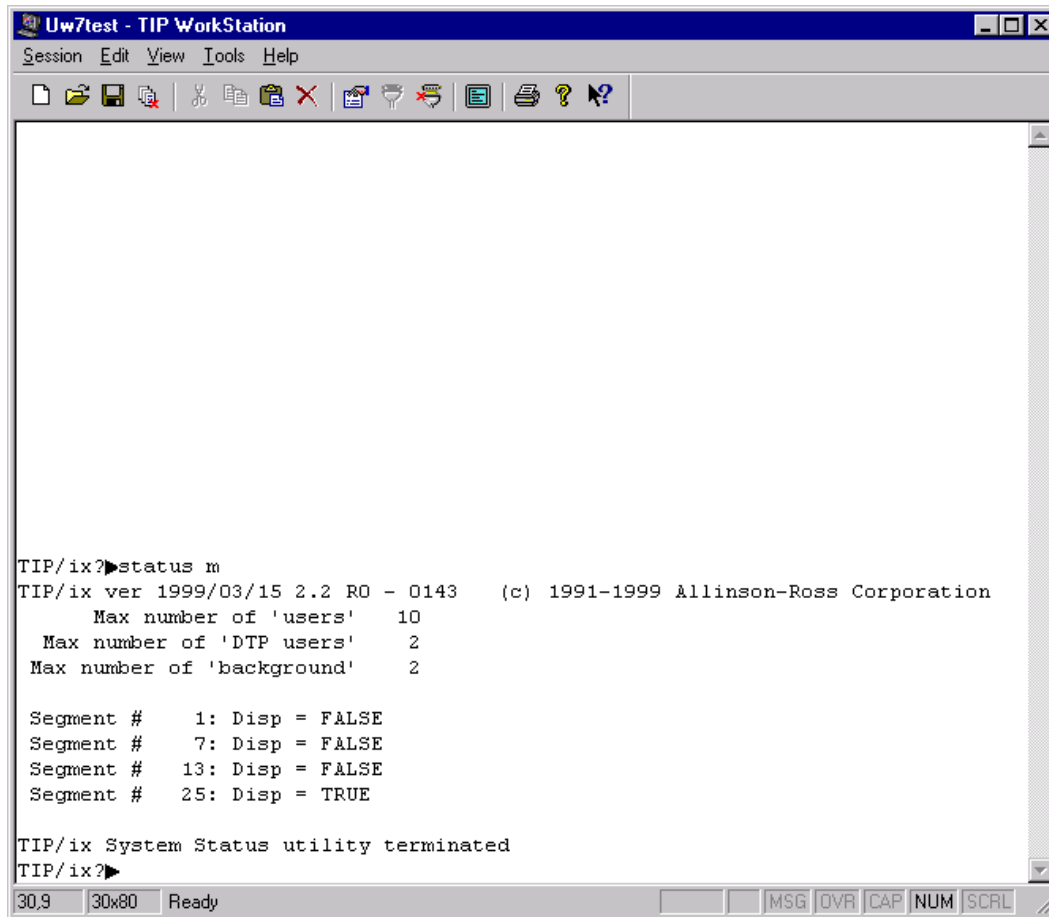
#### Remote File Sent

Count of remote TIPFCS requests sent to the other system.

### m - Memory information report

The **status "m"** command displays information about TIP/ix memory use.

### Example of status m output



```
TIP/ix?▶status m
TIP/ix ver 1999/03/15 2.2 R0 - 0143 (c) 1991-1999 Allinson-Ross Corporation
 Max number of 'users' 10
 Max number of 'DTP users' 2
 Max number of 'background' 2

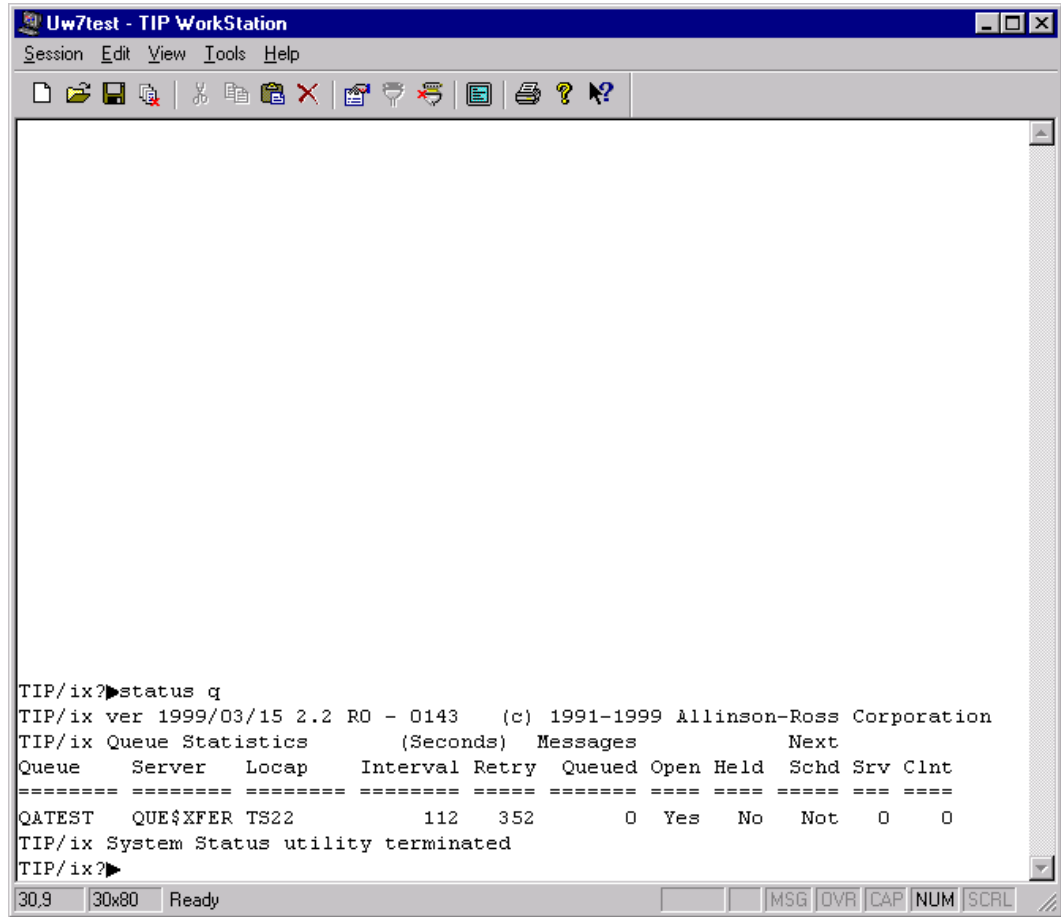
Segment # 1: Disp = FALSE
Segment # 7: Disp = FALSE
Segment # 13: Disp = FALSE
Segment # 25: Disp = TRUE

TIP/ix System Status utility terminated
TIP/ix?▶
```

### q - Queue information report

The **status "q"** command displays information about TIP/ix queue use.

**Example of status q output:**



**r - Run report**

Display information transaction programs and database interface server processes that have been Run since the TIP/ix system was last started up.

### Example of status r output:

```

Uw7test - TIP WorkStation
Session Edit View Tools Help
TIP/ix?▶status r
TIP/ix ver 1999/03/15 2.2 R0 - 0143 (c) 1991-1999 Allinson-Ross Corporation
TIP/ix Transaction programs at 10:08:36

Program Used Msgin Msgout Binary name Most Most Average
 Min Max Cur Active Waiting Response
SMSEC 15 159 173 smsec 0.24
STATUS 14 17 205 status 0.04
OPEN 5 5 10 tqlrn 0.14
FOPEN 4 0 0 fopen
SMFILE 4 95 108 smfile 0.21
SMPROG 2 22 25 smprog 0.19
SMTERM 2 37 37 smterm 0.20
SMUSER 2 16 16 smuser 0.38
FIN 1 0 0 fin
FSE 1 7 12 fse 0.14
SCRATCH 1 1 1 scratch 0.16
SMGRP 1 21 22 smgrp 0.20
SMLCAP 1 24 24 smlcap 0.12
SMPRINT 1 25 27 smprint 0.21
SMQUE 1 37 38 smque 0.18
TFD 1 6 0 tfd

 16 transactions listed
Total messages: Input 472 Output 698; Avg response 0.21 sec
TIP/ix System Status utility terminated
TIP/ix?▶
30.9 30x80 Ready MSG OVR CAP NUM SCRL

```

## s - System status report

The **status "s"** command displays general information about the TIP/ix system itself. This information includes items that are primarily of interest for debugging problems.

**Example of status s output:**

```

Uw7test - TIP WorkStation
Session Edit View Tools Help
TIP/ix?>status s
TIP/ix ver 1999/03/15 2.2 R0 - 0143 (c) 1991-1999 Allinson-Ross Corporation
TIP/ix System Status on 1999/03/31 at 10:08 Session # 31
Global semaphore is 99 System started by 'ScottC' 1999/03/31 at 06:27
Most concurrent users was 2 of possible 12, 17%
Total messages: Input 500 Output 785; Avg response 0.20 sec
Of 24 data files, 2 were accessed and 1 were updated
Total file I/O: 312 Total updates: 16
Avg file I/O: 0.6 avg updates: 0.0 per transaction
Avg transactions per second: 0.04 over 3 hours 41 minutes

Server # Server Type Process Id Queue Id Active Files Program Name

Tran monitor 468 47 N/A tipmon
MCS I/O 471 44 N/A tipmcsio
Thread Mgr 472 45 N/A tippcstm
0 DTP 469 49 0 tipdtp
1 Primary FCS 470 48 5 tipfcs
2 FCS1 474 46 24 tipfcs
3 QUEUE 473 43 0 tipque

Global Data Area size: 2048 bytes, Seg 1, Address 0
Memory -M 148K: Now free: 117K Most used: 32K
-L 600K: Now free: 500K
TIP/ix System Status utility terminated
TIP/ix?>
30.9 30x80 Ready MSG OVR CAP NUM SCRL

```

**t - Transaction Response Time**

The status “t” command displays a rolling log of response times for the last 24 hours (less — if you just started TIP/ix). You can specify the interval size with the RESPLOG= parameter in the tipix.conf file.

### Example of status t output:

```

Uw7test - TIP WorkStation
Session Edit View Tools Help
TIP/ix?▶status t
TIP/ix ver 1999/03/15 2.2 R0 - 0143 (c) 1991-1999 Allinson-Ross Corporation

Time Users #Input #Output #I/Os #Updates Response

7:00 1 10 18 18 1 0.14
7:30 1 0 2 0 0 0.00
8:15 1 7 8 49 0 0.21
8:30 1 30 30 105 0 0.15
8:45 2 66 81 519 7 0.21
9:00 2 68 84 326 1 0.17
9:15 2 62 72 677 10 0.19
9:30 1 139 143 1986 13 0.25
9:45 1 65 69 539 6 0.22
=====
Cur: 1 54 314 - - 0.20
Avg: 1.2 44.7 50.7 421.9 3.8 0.21
TIP/ix System Status utility terminated
TIP/ix?▶
30.9 30x80 Ready MSG OVR CAP NUM SCRL

```

## STOP - Shutdown TIP/ix Immediately

This command causes TIP/ix to shut down immediately. It does not wait for all users to log off.

### Syntax:

STOP

No parameters are required.

### Additional Considerations

The system SHUTDOWN script (if it exists in the TIPROOT directory) is *not* executed.

Under normal conditions, EOJ is the preferred method of system shutdown. A STOP command may be necessary to force off users that are running programs which do not recognize that system shutdown was requested. See the description of the PIB-SYSTEM field in the documentation of the PIB (Process Information Block).



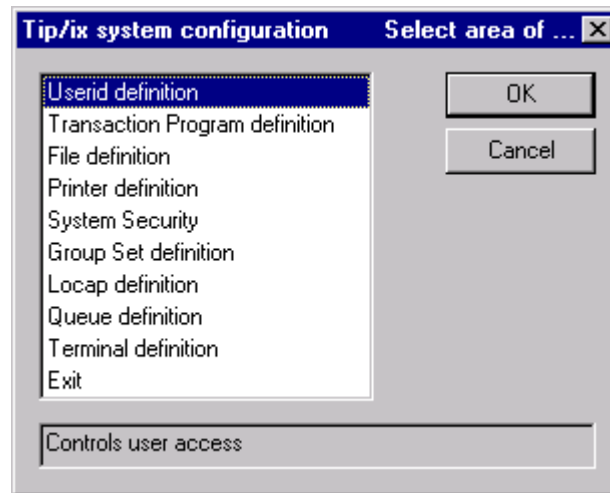
## tcm - TIP/ix Configuration Menu

The **tcm** utility provides a menu-driven front end for the TIP/ix system maintenance programs ([smuser](#), [smprog](#), [smfile](#), [smprint](#), [smsec](#), [smgrpset](#), [smlocap](#), [smqueue](#), and [smterm](#)).

### Syntax:

```
tcm
```

When you enter the above, you will receive the following menu screen:



Select an item by using the arrow keys (or by pressing the first letter of the intended choice).

For each selection, except "Exit", **tcm** calls the appropriate utility program. When control returns from a utility, **tcm** displays the menu selection again.

### **User account maintenance**

Program smuser is called.

### **Program maintenance**

Program smprog is called.

### **File maintenance**

Program smfile is called.

### **Printer maintenance**

Program smprint is called.

### **Security maintenance**

Program smsec is called.

### **Group set maintenance**

Program smgrpset is called.

**LOCAP maintenance**

Program smlocap is called.

**Queue maintenance**

Program smqueue is called.

**Terminal maintenance**

Program smterm is called.

**Exit** The tcm program terminates.

---

## tfd - Screen Format Definition Utility

The **tfd** utility defines and maintains screen formats used by TIP/ix transaction programs. This utility is like the TIP/30 **TFD/TFU** screen format definition utility which defines and maintains TIP/30 screen formats. The screen formats generated (by either utility) are portable between TIP/30 and TIP/ix.

### Features of tfd

The **tfd** utility provides a fast and simple way of creating and maintaining screen formats. Features of the program include:

- Compatibility with existing TIP/30 screen formats.
- A fully window-oriented user interface.
- Single pass definition of formats.
- Automatic syntax checking of fields as they are defined.
- "Point and Shoot" capability for many operations.
- Line and column marking, cutting, and pasting.
- Easy definition of context-sensitive help.
- SFS indicator handling.
- DPS 1100 option handling.

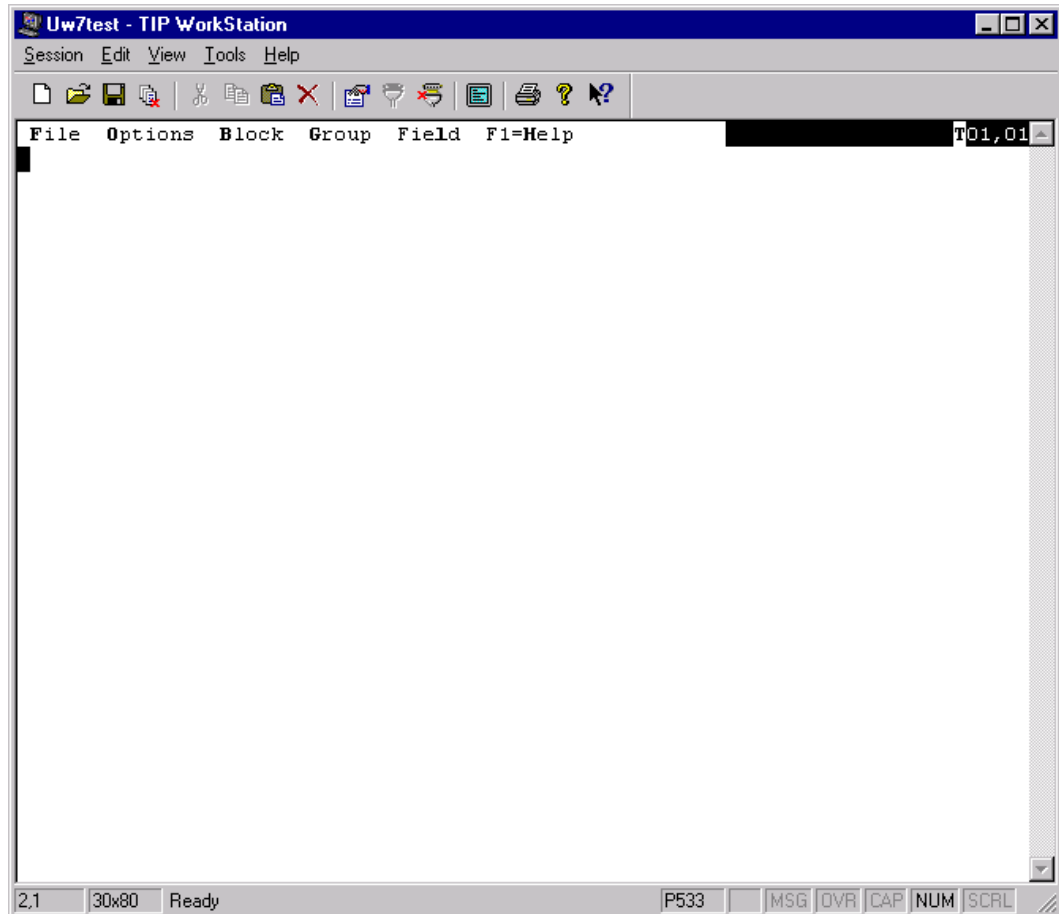
These features make it easier to design screen formats, yet provide compatibility with existing formats.

### Starting tfd

You can run **tfd** directly from the TIP/ix prompt or as a standalone UNIX program. In either case, execute this command:

```
tfd
```

**tfd** displays the following screen:



You can also load a screen format directly and skip the tfd - Main Screen.

Screen formats are referenced by a **format name** which consists of a **group name** and an **element** name. Both of these names must begin with an alphabetic character and cannot contain imbedded spaces or commas. You must precede the element name with the group name, if specified, and separate them with a forward slash ( / ) character.

### Syntax

Use the following syntax to invoke tfd from the command line:

```
tfd [[group/]element]
```

### Where:

#### group

The group name of the screen format. If you do not specify the group, the default is "TIP\$Y\$".

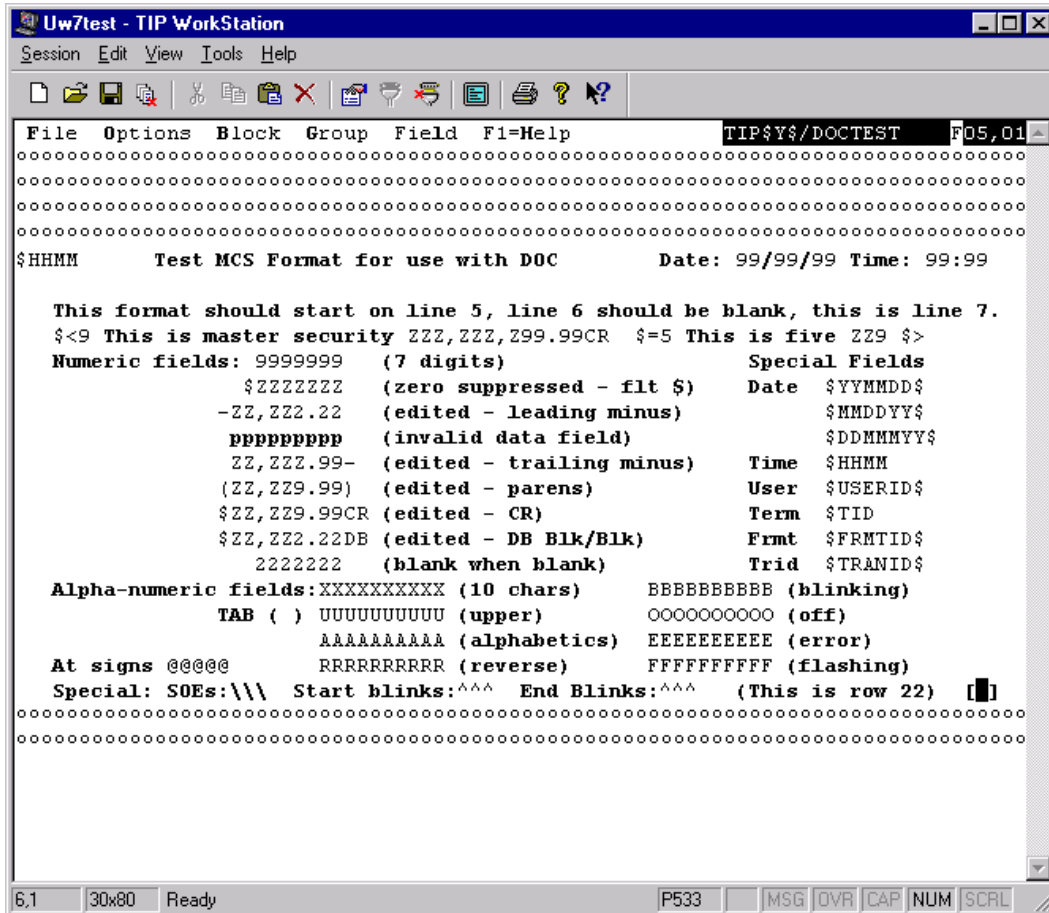
#### element

The element name of the screen format.

### Example 1:

```
tfd doctest
```

In this example, tfd will load the predefined format **TIP\$\$/DOCTEST** and will display the following screen of the loaded format:



This sample screen format contains many of the available field types.

**Example 2:**

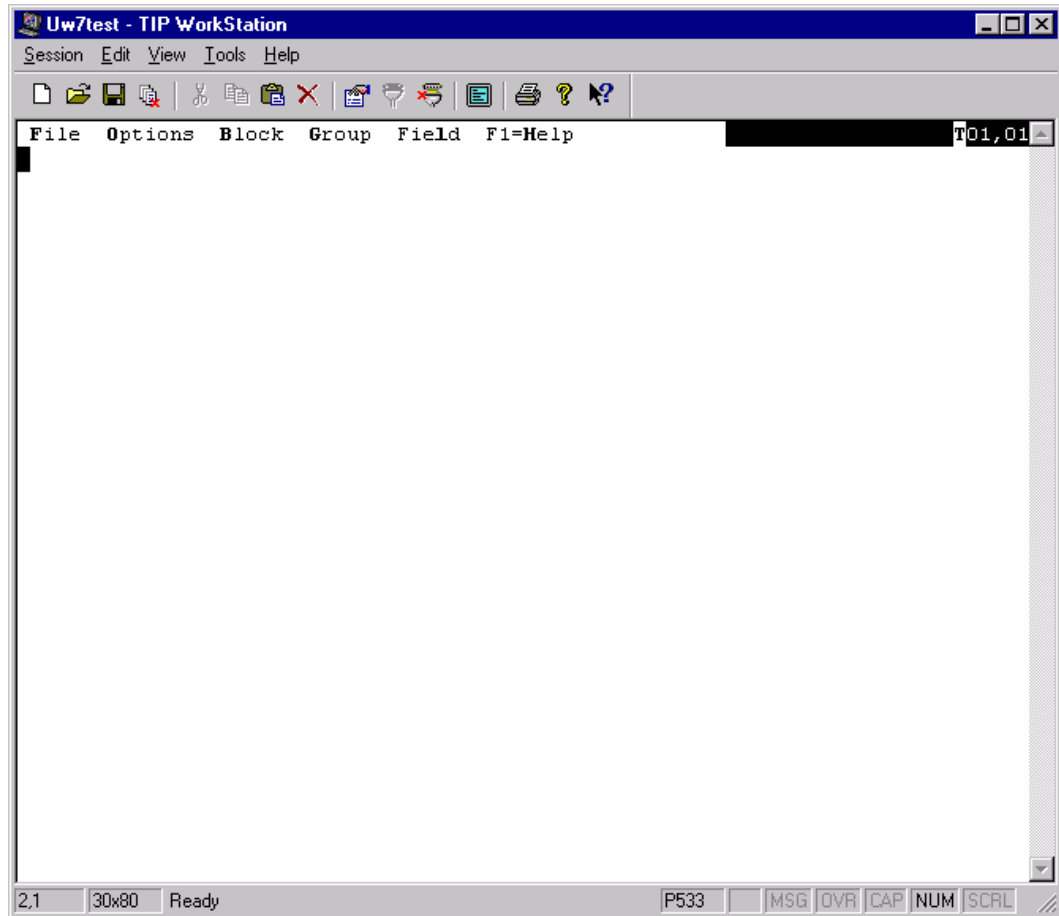
```
tfd GRP1/FRM1
```

In this example, tfd will attempt to load the format **GRP1/FRM1**.

If tfd is unable to find the specified format, you will be notified and presented with a blank editing area to begin definition of the screen format.

**Layout of the tfd Main Screen**

tfd uses a window-oriented user interface to create and edit screen formats. You can edit screen formats in the tfd screen editing area, and you can access various other options by using pull down windows. The pull downs are specified on the highlight bar, or horizontal menu. The layout of the tfd - Main Screen is shown below with identification numbers linking the areas of importance to their descriptions below.



To access the highlight bar from the editing area, press **Ctrl-\** or the **Esc** key. These keys act as a toggle between the highlight bar and the editing area.

### File Selection

This pull down menu allows access to **tfd**'s file operations which can be used to:

- Load Format**
- Save Format**
- Rename Format**
- Delete Format**
- Summaries**
- Catalogue**
- Name Format**
- Quit**

The file operation functions support the use of a *point and shoot* interface for simplifying the various file operations. See [Performing File Operations](#) for more information.

### Options Selection

This pull down window allows you to access various format options for the current screen format including: the cursor resting location, various color attributes, the starting and ending row of the format.

### Block Selection

This pull down allows you to access tfd's **text Blocking** options on lines or columns of text. These options include the ability to cut or copy blocks of text to scrap, and the ability to delete or blank out blocks of text. After text has been copied or cut to scrap, the scrap can be inserted or overlaid where desired and as many times as desired. See [Performing Block Operations](#) for more information.

### Group Selection

This pull down allows you to define and maintain group information as well as indicating and generating any copybook information. See [Group Operations](#) for more information.

### Field Selection

This pull down lets you view and edit the characteristics of the field in which the cursor is currently positioned. If the cursor is not currently positioned within a field, tfd will display an appropriate error message. See [Changing Field Characteristics](#) for more information.

### F1=Help Selection

This pull down allows you to access tfd help information as well as access the tfd Help Definition System. This pull down can be activated by pressing the **F1** or **ALT-H** keys or from the main horizontal menu. For further information on the TIP/ix help system, see [The Help System](#).

### Format Name Area

The name of the screen currently being edited is displayed in this area. In the screen example, the group and format for the current screen is **TIP\$\$/TEST1**. This area is blank if you have not named the format.

### Edit Mode Indicator

This indicator informs you of the editing mode that tfd is currently using. The indicator will be updated to keep you informed of the edit mode. For a description of the tfd editing modes see [Editing Modes](#).

### Row and Column Indicator

This indicator displays the current cursor row and column position. This is useful for determining the starting position and length of fields.

## Screen Editing Area

Screen format editing takes place here.

## Window Interface

Using **tfd**, you may perform editing operations using the **editing area** of the **tfd** screen. You access **tfd** options with pull down menus from the **highlight bar** (horizontal menu) at the top of the screen, or with "**fastpath**" key commands directly from the edit area.

### Using the Highlight Bar

To access the highlight bar from the editing area, press **Ctrl-\** or the **Esc** key. These keys act as a toggle between the highlight bar and the editing area. When you activate the highlight bar, **tfd** will highlight your last selection.

#### Example:

To access the "**Options**" pull down from the editing area press **Ctrl-\** then press **O**.

There are two types of **tfd** pull downs:

1. The first type is a pull down "**menu**".
2. The second type is a "**data-entry**" window. These windows consist of several **fields** into which you may enter data. On exiting a field, **tfd** performs an edit check to ensure that the data entered is correct. If the data is incorrect, **tfd** will *not* permit the cursor to leave the field until the error is corrected.

The following sections describe the functionality of menu and data-entry windows.

## Menus

**tfd** uses both vertical and horizontal menus:

- Horizontal menus consist of a single line or "highlight bar" which contains the available menu selections.
- A vertical pull down menu consists of a box in which selections are displayed in a vertical column.

**tfd**'s pull down menus allow you to select from a list of available operations. A help message is associated with each item in a vertical pull down and is located at the bottom of the menu to assist you in your selection.

Each item in a pull down menu contains one character that stands out from the other characters. For example, the "Options" item on the highlight bar appears as **O**ptions. You may use this character (**O**) to access the item. Therefore, while positioned on the highlight bar, pressing one of these **special characters** will cause that item to be selected. For

example, while positioned on the highlight bar, you would press **F** to access the "**File**" selection.

You can use several other keys to make selections while on a pull down menu:

- |← On a horizontal menu, this causes the item to the left of the currently highlighted item to be highlighted. On a vertical menu screen, this causes the item above the current item to be highlighted.
- | On a horizontal menu, this causes the item to the right of the currently highlighted item to be highlighted. On a vertical menu screen, this causes the item below the current item to be highlighted.
- ↓ On a horizontal menu, this selects the currently highlighted item. On a vertical menu, this causes the item below the current item to be highlighted.
- ↑ On a horizontal menu, this selects the currently highlighted item. On a vertical menu, this causes the item above the current item to be highlighted.
- ← On a horizontal menu, this causes the item to the left of the currently highlighted item to be highlighted. On a vertical menu, this selects the currently highlighted item.
- On a horizontal menu, this causes the item to the right of the currently highlighted item to be highlighted. On a vertical menu, this selects the currently highlighted item.

#### **ENTER or RETURN**

Causes the highlighted item to be selected.

- Esc** Exits the current horizontal or vertical menu. On the main **tfd** horizontal menu, this key toggles control back to the editing area.

### **Data-Entry Windows**

**tfd**'s data-entry windows provide a fast, simple way to enter and validate data. For each data item to be entered, **tfd** displays a help message at the bottom of the window to assist you in choosing the correct response. As mentioned above, **tfd** performs edit checking on exiting a field, to ensure the correctness of the data entered. The method used to access fields on a data-entry pull down is similar to that used on a pull down menu. The keys used in this process are outlined below:

- | Moves the cursor to the next unprotected field.
- |← Moves the cursor to the previous unprotected field.

#### **ENTER or RETURN**

Moves the cursor to the next unprotected field.



- ↑ Moves the cursor to the previous unprotected field.
- ↓ Moves the cursor to the next unprotected field.
- Esc** Exits the current data-entry window and returns control to the place where the window was activated. This causes the window to vanish, restoring the area it had overlaid.

### Wildcard Characters

**tfd** supports the use of wildcard characters for all data-entry windows for both screen formats and help text identifiers. The wildcard characters supported are the asterisk "\*" and the question mark "?" and are handled as follows:

- \* matches zero or more occurrences of any character.
- ? matches one occurrence of any character.

For example, if **SCR?** is entered, then any one of the following would be matched:

**SCRA**  
**SCR0**  
**SCR-**

However, the following would not be matched:

**SCRAP**  
**SMC**

Also, if **S\*** is entered, then all of the above identifiers would be matched.

### Editing Modes

**tfd** can operate in any of three editing modes: 1. Text Mode, 2. Unprotected Text Mode, 3. Field Definition Mode.

#### Text Mode

The **text mode** of **tfd** allows you to type in protected text. This text is referred to as "protected" or "heading" text. Heading text is useful for such things as titles, prompts, and other display only information.

#### Unprotected Text Mode

A technique often employed by application programs is the inclusion of **unprotected** (heading) information in a screen format, usually with a leading **SOE** (?) character. When this information is explicitly unprotected, the user of the screen format can simply place the cursor at the end of the

area and press **XMIT**. This is a simple way to imbed commands in a screen normally used to display data.

### Field Definition Mode

**Field Definition Mode** allows you to enter **data fields** which are used to display and accept information from the screen format. A data field is entered by specifying a **field mask** for that field on the screen at the desired position. Unlike heading text or unprotected text, which may consist of virtually any upper or lower case characters, data field masks are restricted to a certain set of characters. For a description of valid field masks, see [Field Definition Information](#).

The editing mode of **tfd** is displayed by the **Edit Mode Indicator** (see [Layout of the tfd Main Screen](#)). The editing modes and the character displayed by the Edit Mode Indicator are summarized in the table below:

| Editing Mode          | Indicator Reading |
|-----------------------|-------------------|
| Text Mode             | T                 |
| Unprotected Text Mode | U                 |
| Field Definition Mode | F                 |

**tfd** can switch between Text Mode and Field Definition Mode or Unprotected Text Mode and Field Definition Mode without being explicitly instructed to do so.

If **tfd** is editing in Text Mode or Unprotected Text Mode (the Edit Mode Indicator will read "**T**" or "**U**") and the cursor moves into a *previously defined field*, **tfd** will automatically switch to Field Definition Mode to allow the field to be updated.

Switching between modes is explained in the following sections.

### Entering Protected Text

Protected text can be entered whenever the Edit Mode Indicator displays a "**T**", for protected text mode.

If the Edit Mode Indicator displays a "**U**", for unprotected text mode, then pressing **Ctrl-T** will toggle the Edit Mode Indicator to a "**T**" and protected text can be entered.

If the Edit Mode Indicator displays a "**F**", for field mode, then you must finish defining the field and then move out of the field before entering protected text.

### Entering Unprotected Text

Unprotected text can be entered whenever the Edit Mode Indicator displays a "**U**", for unprotected text mode.

If the Edit Mode Indicator displays a "T", for protected text mode, then pressing **Ctrl-T** will toggle the Edit Mode Indicator to a "U" and unprotected text can be entered.

If the Edit Mode Indicator displays a "F", for field mode, then you must finish defining the field and then move out of the field before entering unprotected text.

## Defining A Field

You can only define or update fields while in **Field Definition Mode**. To begin definition of a new field you must press **Ctrl-F**. The Edit Mode Indicator will display the "F" character to inform you of the change to field definition mode.

If the cursor enters a previously defined field, tfd will automatically switch to Field Definition Mode to allow update of the field.

While in Field Definition Mode, any input will be converted to upper case and checked against the set of valid field mask characters (see *Field Definition Information* below). If an invalid character is entered, tfd will beep to indicate the error and the input character will *not* be displayed.

When you have finished entering or updating a field mask, and attempt to exit the field, the mask is then automatically edit checked. If this edit check fails, the portion of the mask that was in error is highlighted, and tfd will *not* allow the cursor to exit the field until the error is corrected.

Since tfd must ensure the integrity of all fields, fields are edit checked before access to **pull downs** is permitted. This prevents you from performing operations using the **pull downs**, when there is a data field which is in error. A tfd field is *not* actually defined until the field mask has been entered, and the field has passed the edit check.

You can enter an edit mask using short form notation. For example, typing **X(10)** in field definition mode is equivalent to typing **XXXXXXXXXX**. This short form notation can be used with normal notation (for example: **Z(5).99** is equivalent to typing **ZZZZZ.99**).

The sections which follow outline a complete listing of valid edit masks and the rules for their use.

## Field Definition Information

You may define data fields to tfd by using combinations of certain characters. For example, you may define a numeric field as **ZZZ99**. This is intentionally similar to COBOL formatted picture clauses.

Both numeric and alphanumeric fields may contain editing codes to specify certain automatic editing that is to be handled transparently.

## Field Definition Codes

**A...A** Define upper case **Alphabetic** field.

On both input and output any alphabetic character will be *automatically* translated to the equivalent upper case character.

**B...B** Define upper case field — **Display Blinking**.

Treated the same as a "U" field, except that the field will be displayed as a **blinking** field. A field which is all spaces cannot blink.

**E...E** Define **error field**.

Error fields are (by definition) output-only areas that are typically used to display **error** or **informational** messages. These are not real data fields.

**F...F** Define upper case field — **Flashing**.

Treated the same as a "U" field, except that the field will be displayed in **reverse video** on a background that alternates from **bright** to **off** intensity.

**G...G** Define upper case field — **Flashing**.

Treated the same as a "U" field, except that the field will be displayed in **reverse video** on a background that alternates from **bright** to **low** intensity. This intensity is also known as **grotesque**.

**H...H** Define upper case field — **Flashing**.

Treated the same as a "U" field, except that the field will be displayed in **reverse video** on a background that alternates from **low** to **off** intensity. This intensity is also known as '**hideous**'.

**L...L** Define upper case field — **Low intensity**.

Treated the same as a "U" field, except that the field will be displayed in **low intensity**.

**N...N** Define upper case field — **Normal intensity**.

Treated the same as a "U" field, except that the field will be displayed in **normal intensity**.

**O...O** Define upper case field — **display off**.

Treated the same as a "U" field, except that the field will be hidden.

**R...R** Define upper case field — **reverse video**.

Treated the same as a "U" field, except that the field will be displayed in **reverse video**.

**S...S** Define upper case field — **Shaded**.

Treated the same as a "U" field, except that the field will be displayed in **reverse video** on a **low intensity** background.

**U...U** Define upper case field.

On both input and output, any alphabetic character will be automatically translated to the equivalent upper case character.

**X...X** Define alphanumeric field.

Any printable characters may be input or output in this format. *No translation or checking is performed.*

**Z...Z** Define numeric digit with zero suppression.

A "Z" field definition code represents a digit of a numeric field which will be displayed as a **space** on output if the digit is a leading zero. The digit will be forced to a valid digit on input (space becomes a zero).

**9...9** Define numeric digit.

A "9" field definition code represents a digit of a numeric field which will be unconditionally displayed on output. The digit will be forced to a valid digit on input (space becomes a zero).

**2...2** Define numeric digit — Blank if not input.

A "2" field definition code represents a digit of a numeric field which will be unconditionally displayed on output. The digit is input as a space if not entered.

If a "2" definition is used in a field, the entire field will be returned as **spaces** if it is not input. This allows the combination **ZZZ,ZZ2.22** to be edited as **ZZZ,ZZ9.99** with consideration that the field will be blank if not input.

**Examples:**

**UUUUUUU**

defines a 7 character upper case field.

**EEEE**

defines a 4 character error field.

**UUUXXX**

defines two adjacent alphanumeric fields.

**ZZZ999**

defines a 6 digit numeric field with leading zero suppression.

**Field Editing Codes**

You may define **numeric** data fields to include various types of editing. The editing that occurs is transparent to the application program that uses the screen format.

**Comma Insertion**

Place commas in a numeric field to specify comma insertion.

*Example: 99,99*

**Decimal Point**

Place a decimal point in a numeric field to indicate decimal point alignment.

*Example: ZZ9.99*

**Leading Minus**

Specify a leading minus sign to cause a (floating) leading minus sign to appear for negative numeric values.

*Example: -ZZ,ZZ9*

**Trailing Minus**

Specify a trailing minus sign to cause negative numeric values to appear with a minus sign following the last digit.

*Example: ZZ,ZZ9-*

**Parentheses**

Place parentheses around a numeric field to cause negative values to be surrounded by a **floating left** and a **fixed right** parenthesis.

*Example: (ZZ,ZZ9)*

**CR Symbol**

Append the two characters "**CR**" to a numeric field to cause "CR" to follow a negative value.

*Example: ZZ,ZZ9CR*

**DB Symbol**

Append the two characters "**DB**" to a numeric field to cause the symbol "DB" to follow a negative value.

*Example: ZZ,ZZ9DB*

**'/' Insertion**

Edit a numeric field with imbedded slash characters ("**/**"). This is often specified for **date** fields.

*Example: 99/99/99*

### **':' Insertion**

Edit a numeric field with imbedded colon characters (":"). This is often specified for **time** fields.

*Example: 99:99:99*

### **Floating Currency**

Specify a floating currency symbol for a numeric field. TIP/ix will float the currency symbol "\$" in front of the first digit displayed.

*Example: \$ZZ,ZZ9.99*

## **Special Definition Codes**

This section documents some special definition codes that you may include in a format to represent data to be supplied by the **MCS** output routines. This reduces the amount of effort required by the application program.

### **\$MMDDYY\$**

This string of characters will be replaced by the current date in the format: **month day year**.

*Example: Date: \$MMDDYY\$ — Date: 2/7/95*

### **\$DDMMYY\$**

This string of characters will be replaced by the current date in the format: **day month year**, where the **MMM** field is the English abbreviation of the month.

*Example: Date: \$DDMMYY\$ — Date: 7 FEB 95*

### **\$YYMMDD\$**

This string of characters will be replaced by the current date in the format: **year month day**.

*Example: Date: \$YYMMDD\$ — Date: 95/2/7*

### **\$HHMM**

This string of characters will be replaced by the current time in the format: **hours minutes**.

*Example: Time: \$HHMM — Time: 1:52*

### **\$user id\$**

This string of characters will be replaced by the eight character user id of the user that is logged on the terminal that is using the screen format.

*Example: user id: \$user id\$ — user id: ALLINSON*

**\$TID** This string of characters will be replaced by the name of the terminal where the screen format is being used (**four** characters).

*Example:* Term: **\$TID** — Term: **TD69**

### **\$FRMTID\$**

This string of characters will be replaced by the name of the screen format (**eight** characters).

*Example:* Format: **\$FRMTID\$** — Format: **DOCTEST**

### **\$TRANID\$**

This string of characters will be replaced by the **transaction code** (**eight** characters) of the program currently displaying the screen format.

*Example:* Trans: **\$TRANID\$** — Trans: **PAYROLL**

## Language Considerations

TIP/ix and tfd handle different interpretations for different languages for the three numeric field symbols: the comma, the decimal point and the currency symbol.

These symbols are configured at installation time of TIP/ix to set the decimal point symbol to be either the period or the comma and to set the currency symbol to the specific symbol for the local language. The **default** decimal point symbol is the period “.” and the **default** currency symbol is the dollar sign “\$”.

See the discussion of the **TIPINSTALL** program for information on configuring these symbols at installation time.

Whatever symbols have been configured to TIP/ix for the comma, decimal point and currency symbol must be used in tfd when defining numeric data fields.

## Special Heading Codes

This section explains some special codes that you can include in a tfd screen format as heading information.

### **Underscore '\_'**

Enter an underscore character in unprotected text mode (see [Editing Modes](#) and [Entering Unprotected Text](#)) to designate a cursor resting location. TIP/ix will set this character (by **default**) to an unprotected underscore that is not a data field.

*Example:* [ \_ ]

### **Blink Character**

Use the blink character in heading information to blink the



characters '<' and '>'. For more information, see *Options Selection* above.

*Example: <<< This is important >>>*

### SOE Character

Use the **SOE** (?) character in heading information to represent a real SOE character. For more information, see *Options Selection* above.

*Example: ? whoson*

## tfd Security

You will often need a program to display certain information only if the user has a particular **security clearance**. For example, a payroll inquiry program may wish to display salary information only if the user has a certain security level.

To facilitate this, **tfd** allows the designer of a screen format to specify that an area of the screen is not to be displayed unless the user has a specific level of security. **tfd** considers security specifications to be fields; you must therefore enter them in field definition mode.

Once **tfd** encounters a security specification, it applies it to *all* heading and data information *until* it encounters another security specification.

### **\$<nnn**

This string consists of a dollar sign followed by a less-than symbol followed by a one, two or three digit number representing the security level.

**tfd** will not display the heading and data information from this point on unless the user's security level is numerically less than or equal to the value specified.

*Example: \$<21* will allow the display of following headings and data only if the user has a security of **1** through **21**.

### **\$=nnn**

This string consists of a dollar sign followed by an equal symbol followed by a one, two or three digit number representing the security level.

**tfd** will not display the heading and data information from this point on unless the user's security level is exactly equal to the value specified.

*Example: \$=21* will allow the display of following headings and data only if the user has a security of **21**.

### **\$>**

This string consists of a dollar sign followed by a greater-than symbol. This is used to "turn off" the security mechanism.

**Note:** For this feature to work efficiently you must increase the security on the **tfd** program via [smsec](#).

#### **Additional Considerations:**

These security specifications may be placed throughout the screen format. The screen positions occupied by these strings are turned to **protected spaces**, and are therefore "wasted".

The heading and data fields that are "concealed" by the action of a security specification string are not "squeezed" out of the format. The screen space they would normally occupy is filled with protected spaces.

## **Performing File Operations**

**tfd** provides a complete set of file operations to allow you to perform a variety of file related commands. The user interface to operate each of the file operations is essentially the same.

When you select any file operation, you are prompted for the format name on which the operation is to be performed. See [Starting tfd](#) for information on the syntax of **tfd** format names.

**tfd** supports the use of wildcard characters to simplify entry of format names. When using file operations, the inclusion of a wildcard character in a format name or entering of a blank format name will cause **tfd** to enter "point and shoot" mode. See [Wildcard Characters](#) for more information on the how wildcard characters are handled.

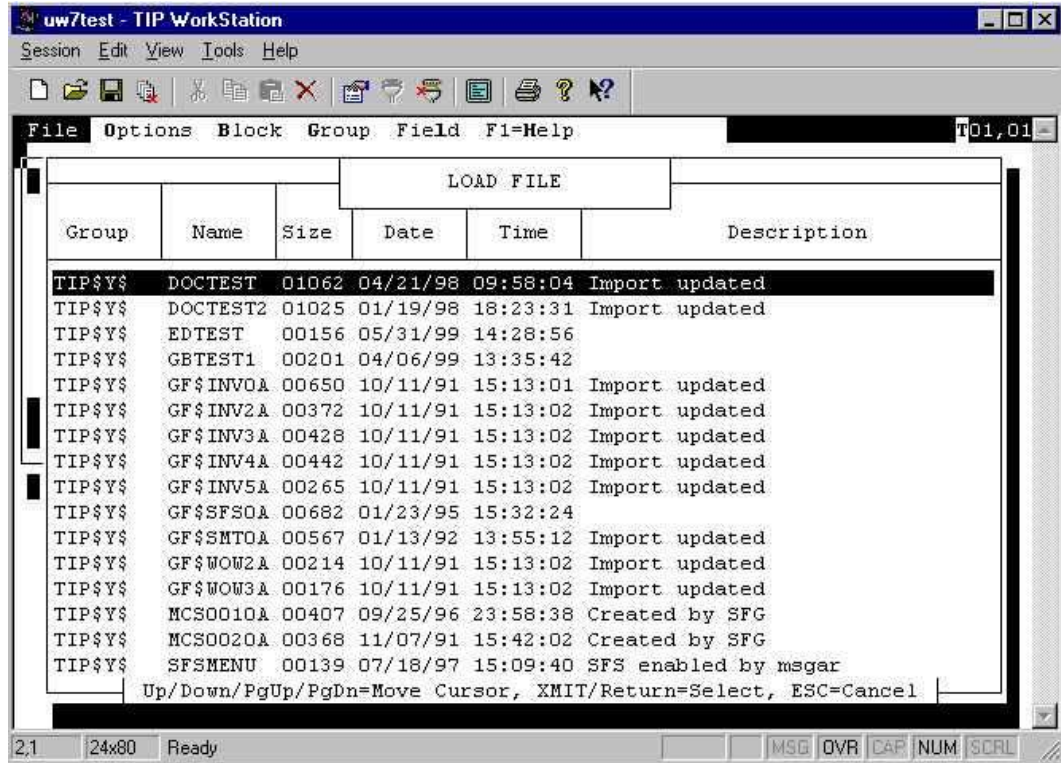
### **Point and Shoot Interface**

In point and shoot mode, **tfd** will display a window containing the names of all formats which correspond to the pattern entered.

**tfd** converts a blank format name to **"\*/\*"**.

You can use point and shoot mode with all file operations except for the save operation (see *Save Format* below).

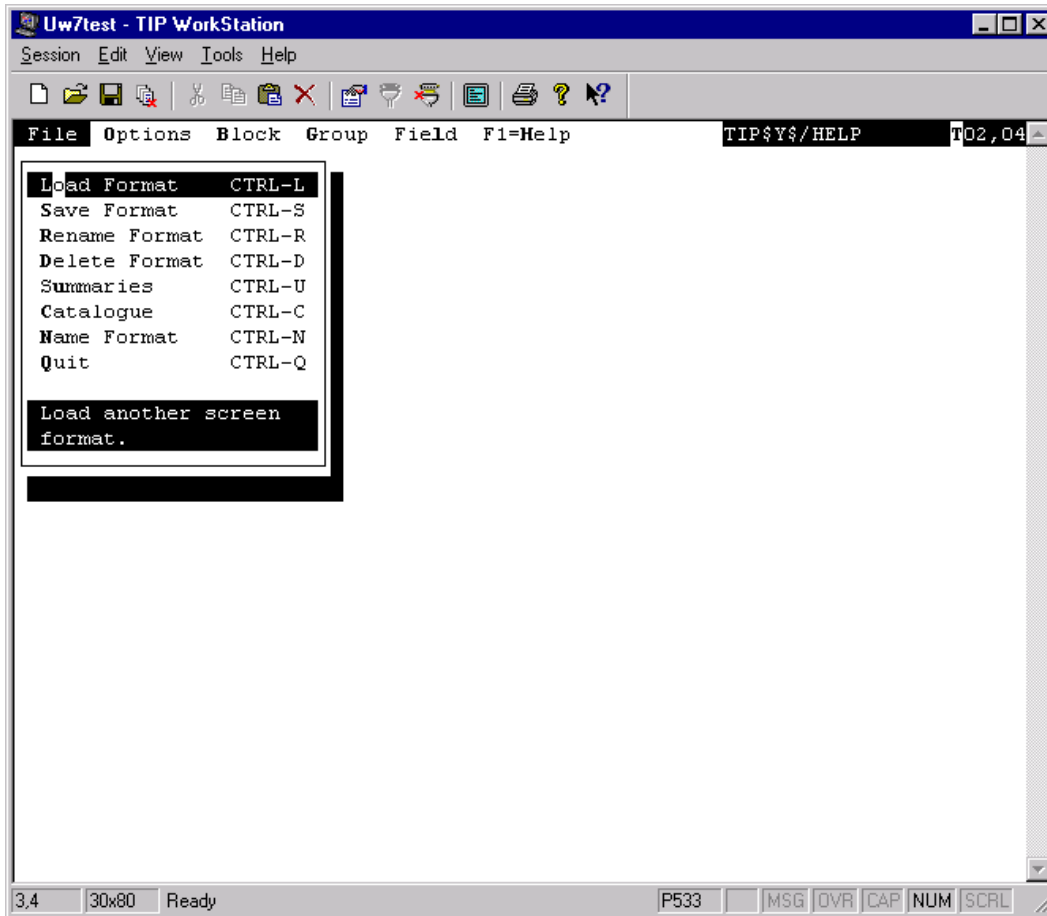
If you select a point and shoot option, **tfd** will display a screen similar to the following **tfd** — Point and Shoot Screen:



You can use the **Up** and **Down** keys and the **Pgup** and **Pgdn** keys to position the highlight bar (shown as a shaded bar above) over the desired format name. Press **Esc** at any time to cancel a file operation.

When the desired format name has been highlighted, press the **ENTER** or **XMIT** key to cause the file operation to be performed on the selected format, just as if you had explicitly typed the name in.

If you select the File option from the tfd main horizontal menu, the following tfd — File **pull down** menu will be displayed:



The file options available on the above screen are as follows:

### Load Format

The **Load Format** option will accept a format name either explicitly, or by using the point and shoot interface, and then attempts to load the format. If the current format being edited has not been saved prior to a load attempt, tfd will prompt you to save the current format *before* the load is performed.

If the specified format cannot be found, tfd begins with a blank editing area, and informs you that this is a *new* format. The format name area will display this name.

### Save Format

The **Save Format** option prompts you to enter the name with which to save the format. If the format already has a name, this name is displayed as the default and can be changed if desired. The save command does *not* operate in point and shoot mode. If you attempt to save a format with a name already present in the format library, tfd prompts you to overwrite it. This prevents formats from being accidentally overwritten. After a format is saved, its name is displayed in the format name area.

## Rename Format

The **Rename Format** option accepts a format name either explicitly, or by using the point and shoot interface, and then attempts to locate the specified format. If this format cannot be found, tfd displays an error message and the rename operation is aborted. If the specified format is found, tfd prompts you for the *new* format name and performs the rename operation. If a format with the new name already exists in the format library, tfd displays an error message and the operation aborted.

## Delete Format

The **Delete Format** option accepts a format name either explicitly, or by using the point and shoot interface. For delete, the point and shoot mode functions slightly differently. tfd allows the deletion of multiple formats at a time and the point and shoot interface allows you to "mark" any number of formats by pressing the **SPACE BAR** to highlight your choices. When you press the **ENTER** key, tfd opens a window and begins to delete each marked format until the operation is completed, or you press **Esc** to abort.

When you use the mark feature, the position of the highlight when you press **ENTER** is irrelevant. If you wish to delete a single format using the "point and shoot" interface, simply position the highlight bar on the name of the desired format and press the **SPACE BAR** to highlight that entry and press **ENTER** to delete that format.

## Summaries

The **Summaries** option accepts a format name either explicitly or by using the point and shoot interface. If the specified format cannot be found, then tfd will display an appropriate error message and abort the operation.

A tfd format summary provides a pop-up window which contains useful information about the format. This information includes the format size and date, the number of data bytes, and other useful information.

## Catalogue

The **Catalogue** option provides a method of displaying the formats present in the tfd format library. tfd prompts you for a search pattern and displays formats matching this pattern in a window format. Use the **Pgup** and **Pgdn** keys to view the various format names in the catalogue.

## Name Format

This **Name Format** option allows you to name a tfd format. The new name is displayed in the format name area at the top of the screen. This naming has no immediate effect on the format library, it simply tells tfd what name to use when the format is to be saved. This is analogous to leaving the name blank and specifying the name when prompted by the **save** command. A description field is provided that allows you to enter a short description of the current screen format.

## Quit

The **Quit** option is used to terminate **tfd**. If you attempt to quit and the current format has not been saved, **tfd** will prompt you to save the format before quitting. If a format is being edited that is not in the format library, you will be prompted to save it before exiting.

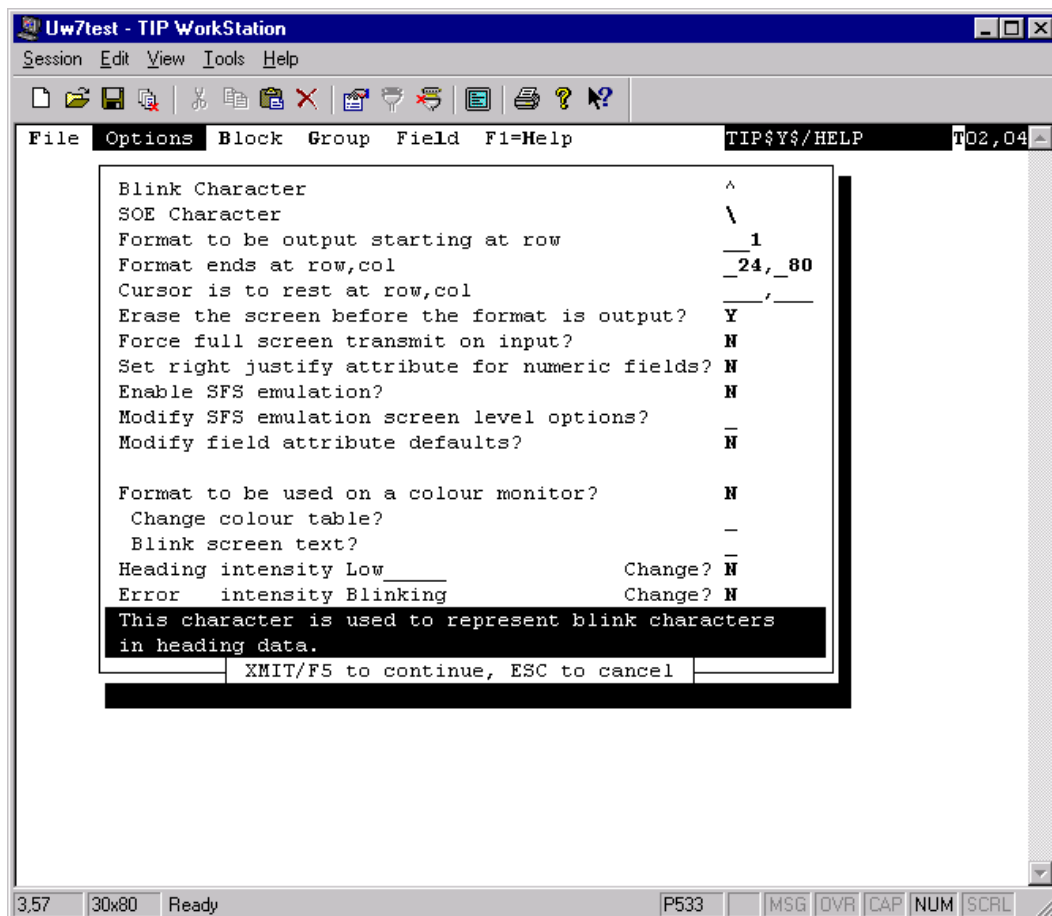
You can also quit **tfd** by pressing **Ctrl-Q** while in the **tfd** main screen editing area.

## Setting Options

While editing a screen format, you may wish to change a variety of screen and field options. You can easily accomplish this by using the **Options** and **Defaults pull downs**. These **pull downs** and their contents are explained in the sections which follow.

### Options Pull down

This **pull down** allows you to access various options for the current screen format including the cursor resting location, various color attributes and the starting and ending row of the format. If you select the "Options" selection from the **tfd** main horizontal menu, the following **tfd** — Options data-entry window will be displayed:



An explanation of the layout, and of the selections it contains, is given below:

### Blink Character

This character represents blink characters in heading data. This character will be replaced by either a left or right blink character (X'1C or X'1D'). tfd begins by using the left blink character and alternates with a right blink. For two or more adjacent blink characters, no alternation is done (two or more adjacent blinks will result in the same type of blink character).

Default: ^ (circumflex, carrot)

### SOE Character

The character is used whenever a real **SOE** character is desired.

Default: \ (backslash)

### Format to be output starting at row

This governs the starting row of the format. This value must be between 1 and 255 inclusive and be less than the first non-blank row of the screen format.

Default: Row 1

**Format ends at row**

This governs the ending row of the format. This value must be between **1** and **255** inclusive and be greater than the last non-blank row of the screen format.

Default: Row 24

By setting the starting and ending row of a format, you may make some areas on the edit screen inaccessible. tfd will display a fill character ('o') in these areas.

**Cursor is to rest at row,col**

These fields specify the row and column coordinates (respectively) where the cursor is to be placed when the format is output. The row value must be between **1** and **255** inclusive, and the column value between **1** and **255** inclusive.

Default: No values are specified.

If you do not specify values, tfd will set these values to the row and column of the first character of the first unprotected data field.

**Erase the screen before the format is output?**

Choose **"Y"** or **"N"** to indicate whether the screen is to be erased before the format is output.

Default: 'Y'

You should choose the default unless you are defining a portion of a split screen application.

**Force full screen transmit on input?**

Choose **"Y"** or **"N"** to indicate whether the entire screen is to be transmitted on input. With this enabled, MCS-FUNCTION will be set to 'A' for full transmit.

Default: 'N'

**Set right justify attribute for numeric field?**

Chose **"Y"** (or **"N"**) to have numeric fields set with the right-justified attribute ON (or OFF). In a numeric field:

**Y (on)** This acts like a field on a real UTS terminal with the numeric right-justified attribute set.

**cursor is leftmost**

If the cursor is in the first (left most) position of the field, then each new digit enters at the right, and the other digits shift left.

In the following examples, a lowercase 'b' represents a blank, bold represents the cursor



position, and an underline represents the data entry position. For example, type "123".  
"bb123".

#### cursor not leftmost

If the cursor is not in the first position of the field, the cursor is the data-entry point.

For example, starting with the result of the previous example, type "Y".

The cursor moves to the third character. "bb123"

Now, type "4". The result depends on the mode the keyboard is in:

|                 |         |
|-----------------|---------|
| overstrike mode | "bb423" |
| insert mode     | "bb412" |

**N (off)** With the attribute OFF, data characters enter into the current cursor location. Starting at the leftmost position, type "123"

"123bb"

Default: 'N'

#### Enable SFS emulation?

Use this item to indicate whether the format contains fields which use **SFS** (Screen Format Services) emulation.

Default: 'N'

#### Modify SFS screen level options?

Enter 'Y' to bring up a menu which allows for modifying the screen level options used under SFS emulation. The options available to be modified are for the error field information and for modifying the non-displayed constant. See *SFS Screen Level Options* on page 625 for more information.

Default: 'N'

#### Modify field attribute defaults?

Enter 'Y' to bring up a menu which allows for modifying the default field attributes. See [Default Field Characteristics](#) for more information.

Default: 'N'

#### Format to be used on a color monitor?

This option is only intended for use with non-remote MCS (non-smart mode TIP/fe) sessions. To configure remote MCS sessions the user must use TIP/fe format/color configuration menu option.

Choose "Y" or "N" to indicate whether the format is to be used on a color or monochrome monitor.

Default: 'N'

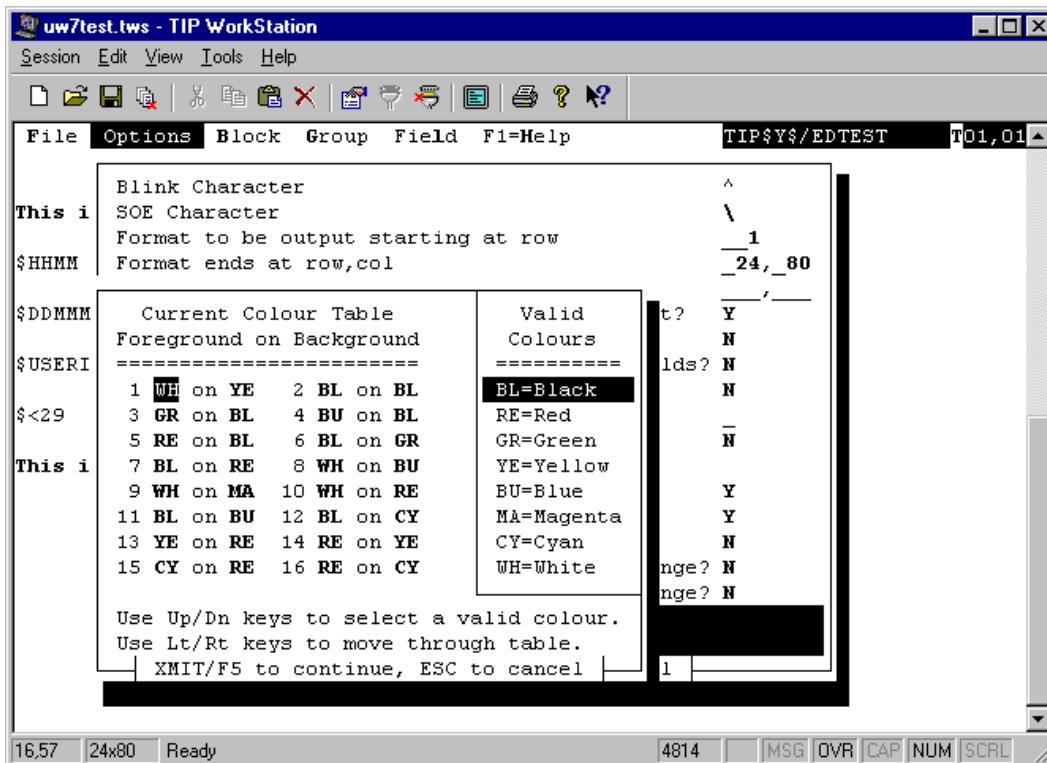
### Change color table?

Choose "Y" to indicate that you wish to change the current color table.

Default: 'N'

If, in the preceding option, you specified a non-color monitor, this field will be blank and protected.

If, in the preceding option, you specified a color monitor, and enter "Y" in this field, tfd will display the following tfd — Color Options Pulldown Screen:



Use the and keys to select a valid color from the list. Use the and keys to move through the foreground and background colors for the 16 available color combinations.

When you have selected colors for both foreground and background, press **ENTER** and your selection will be placed in the color table.

The color combinations you define here are the *only* ones that may be used in this screen format.

### Blink screen text?

This field specifies whether screen text is to blink. Only valid for formats used on a color monitor.

Default: 'N'

## Heading Intensity or Heading Color

This item specifies the foreground and background color of heading text for a color format or heading intensity for a monochrome format.

Enter "Y" to change the current heading color combination or heading intensity. If you choose "Y", tfd will present you with a **pull down** menu containing all the valid color combinations from the color table **OR** all the valid heading intensities, depending on whether you have defined the monitor as color or monochrome.

Use the **↑** and **↓** keys to move through the table and press **ENTER** to select the highlighted option. The selected option will be displayed as protected text on the Options **pull down** menu.

## Error Intensity

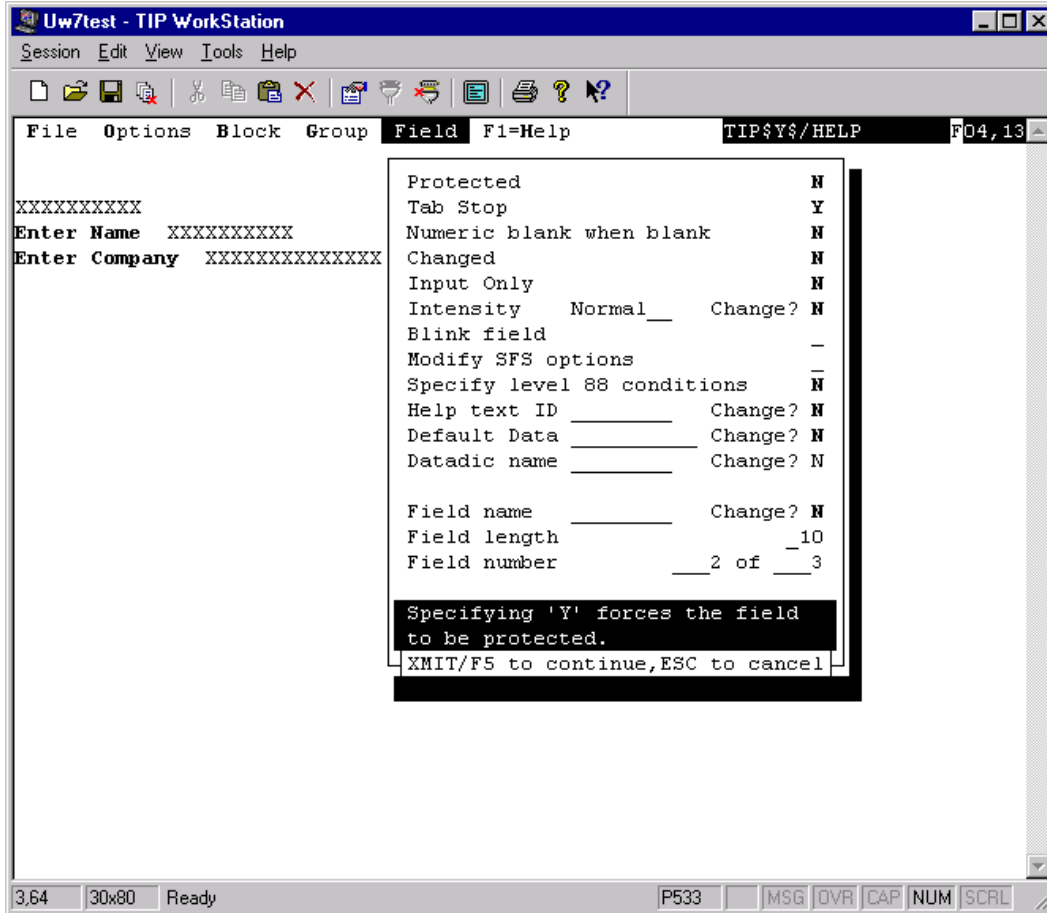
This item specifies the color or intensity for error fields.

Enter "Y" to change the current error color or intensity. If you choose "Y", you will be presented with a **pull down** menu containing all the valid color combinations from the color table **OR** all the valid heading intensities, depending on whether you have defined the monitor as color or monochrome.

Use the **↑** and **↓** keys to move through the table and press **ENTER** to select the highlighted option. The selected option will be displayed as protected text on the Options **pull down** menu.

## Default Field Characteristics

The default field characteristics are set using the Defaults **pull down** on the main tfd menu bar. Each time a new field is defined, its characteristics are defined according to the current entries in this **pull down**. Items in the default **pull down** can be changed at any time during the editing process. See [Defining A Field](#) for more information.



When attempting to modify the default field attributes, the following tfd — Defaults Pulldown Screen will be displayed and the individual items are described below:

### Protected

Choose "Y" to define a protected field. Protected fields provide for display only data.

Default: 'N'

### Tab Stop

Choose "Y" to cause the FCC for this field to be built with the automatic TAB flag set on. Specify "Y" in this field to allow you to tab to this field at run time.

Default: 'Y'

### Numeric Blank when blank

This item applies to numeric fields only and is ignored for non-numeric fields. Choose "Y" to fill the **MCS-DATA** area for this field with spaces if nothing is input for numeric fields.

Default: 'N'

### Changed

Choose "Y" to cause the FCC for this field to be built with the changed flag set on.

Default: 'N'

### Input Only

Choose "Y" to specify that this field is an input-only field. This item is only applicable to **SFS** (Screen Format Services) emulation.

Default: 'N'

### Field Intensity or Field Color

This item specifies the foreground and background color of a field for a color format or field intensity for a monochrome format.

Enter "Y" to change the current field color combination or field intensity. If you choose "Y", tfd will present you with a **pull down** menu containing all the valid color combinations from the color table **OR** a **pull down** menu containing all the valid field intensities, depending on whether you have defined the monitor as color or monochrome.

Use the **↑** and **↓** keys to move through the table and press **ENTER** to select the highlighted option. The selected color will be displayed in a protected field on the Defaults **pull down** menu.

### Blink field

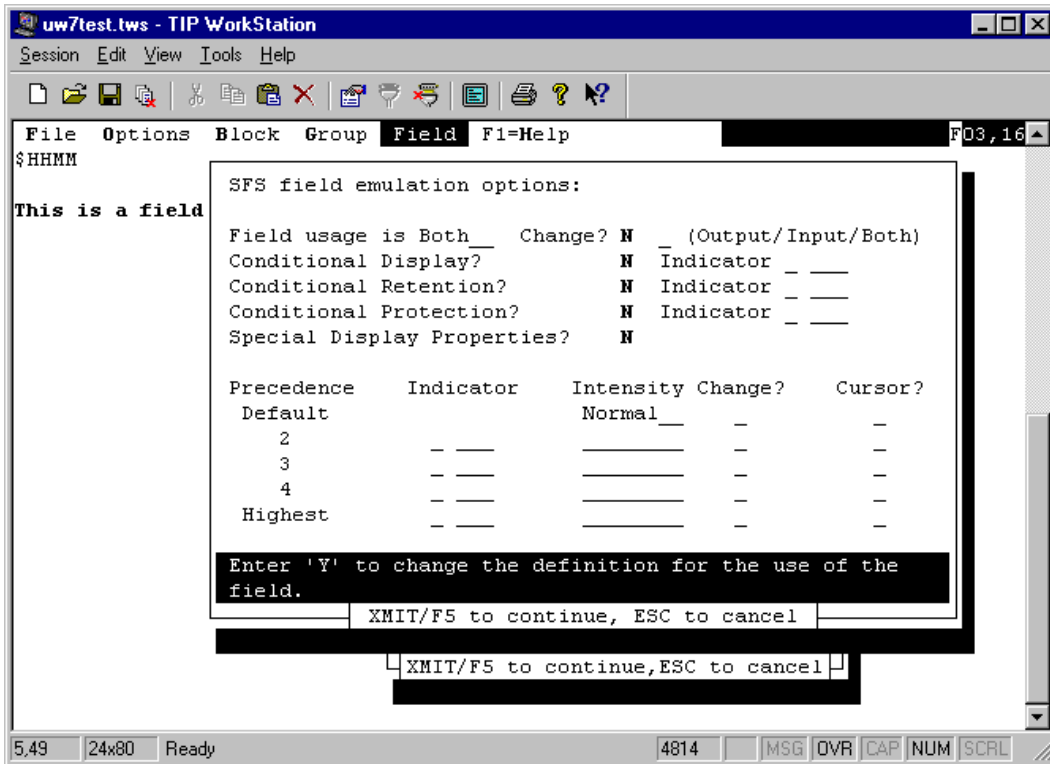
This field specifies whether the field is to blink. Only valid for formats to be used on a color monitor.

Default: 'N'

### SFS Screen Level Options

The SFS screen level options relate to the error field attributes, such as the display row, error indicator and the non-displayed constant.

When attempting to access the SFS screen level options, the following tfd — SFS Screen Level Options Screen will be displayed and the individual items are described below:



### Are error messages defined?

Choose "Y" to indicate that error messages are activated for this screen. The attribute options following this question will become unprotected to allow for entering of the appropriate options.

Default: 'N'

### Number of rows in error message

Error messages can be either 1 or 2 rows in size. Enter one of these only two options to define the number of rows in the error message.

Default: '1'

### Error message starting row

Indicate in this field the starting row at which the error message is to be located.

Default: '24'

### Error message indicator

Enter in this field the indicator which will activate the error messages at run time as well as the activation indicator to determine if the error messages are displayed when the given indicator is ON or OFF.

There is no default error message indicator and an indicator must be specified in order to define SFS error messages with the current screen.

### Error message intensity

This item specifies the intensity to be used at run time for the defined error messages.

Enter "Y" to change the current intensity and you will be presented with a **pull down** menu containing all the valid color combinations from the color table **OR** a **pull down** menu containing all the valid intensities, depending on whether you have defined the monitor as color or monochrome.

Use the **↑** and **↓** keys to move through the table and press **ENTER** to select the highlighted option. The selected intensity will then be displayed as the error message intensity.

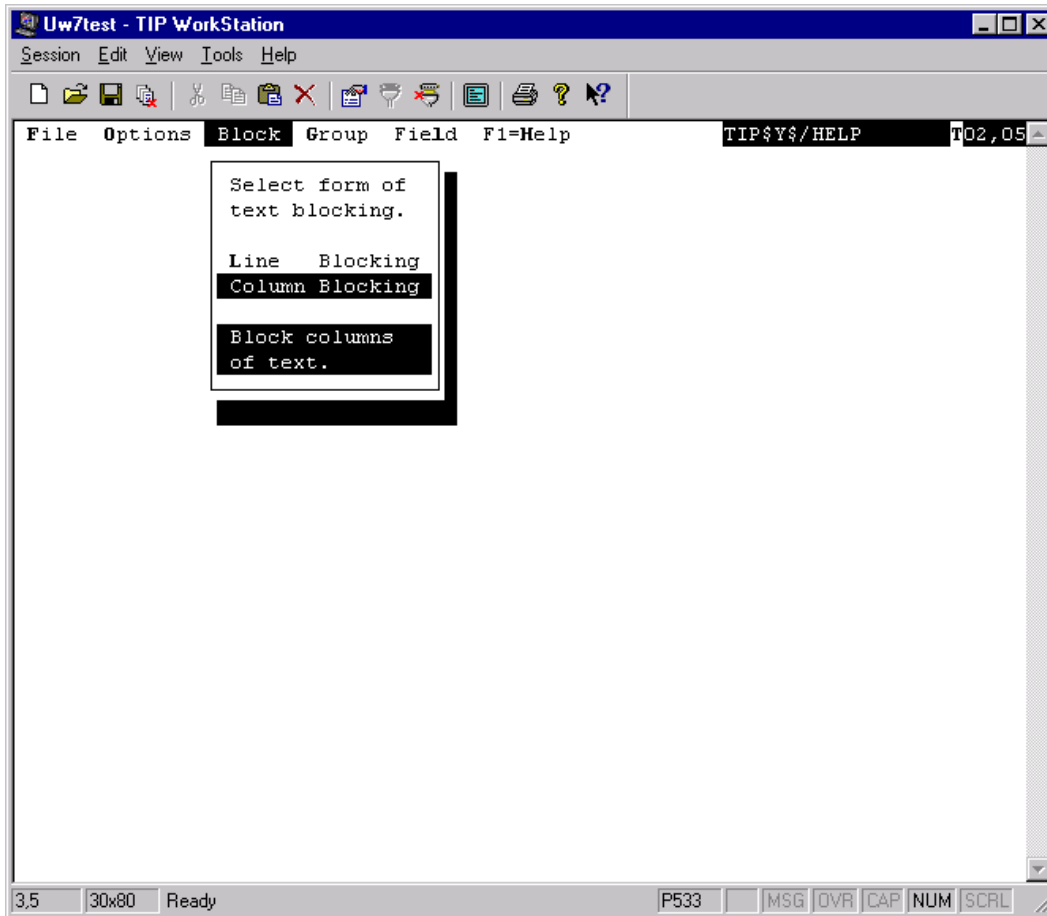
### Modify non-displayed constant

This item allows for indicating a maximum 80 character non-displayed constant with this screen.

RPG identification characters are handled by this non-displayed constant. Enter the single ID character in the first position of this constant and hit **ENTER** or **XMIT** to continue.

## Performing Block Operations

tfd supports block editing operations for both line blocking and column blocking which provides the ability to cut, copy, delete or blank out blocks of text. When copying or cutting blocks of text, the text is copied or moved to a scrap area from which you can then insert or overlay that scrap area as many times as desired. To perform tfd block operations, use the following tfd — Block Operations Pulldown Screen:



The first step is to select what type of blocking option to use; either line blocking or column blocking.

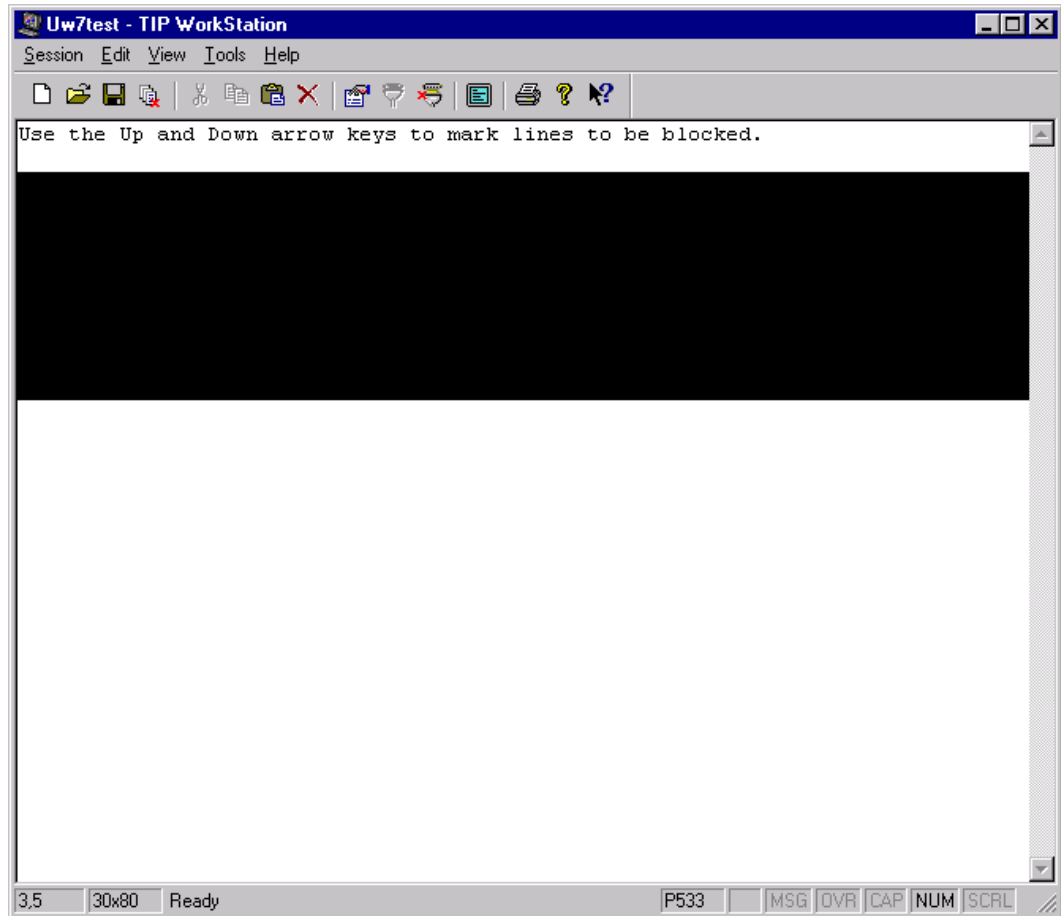
In column blocking, only entire fields can be blocked. A field cannot be partially outside of the marked block or tfd will display the following flashing error message:

**Marked columns cut through existing fields,  
strike any key to continue...**

When you strike any key, tfd will allow you to mark other columns to block. You may continue to change your block or press **Esc** to exit.

After selecting either line or column blocking, a screen similar to the following will be displayed which shows an example of line blocking:



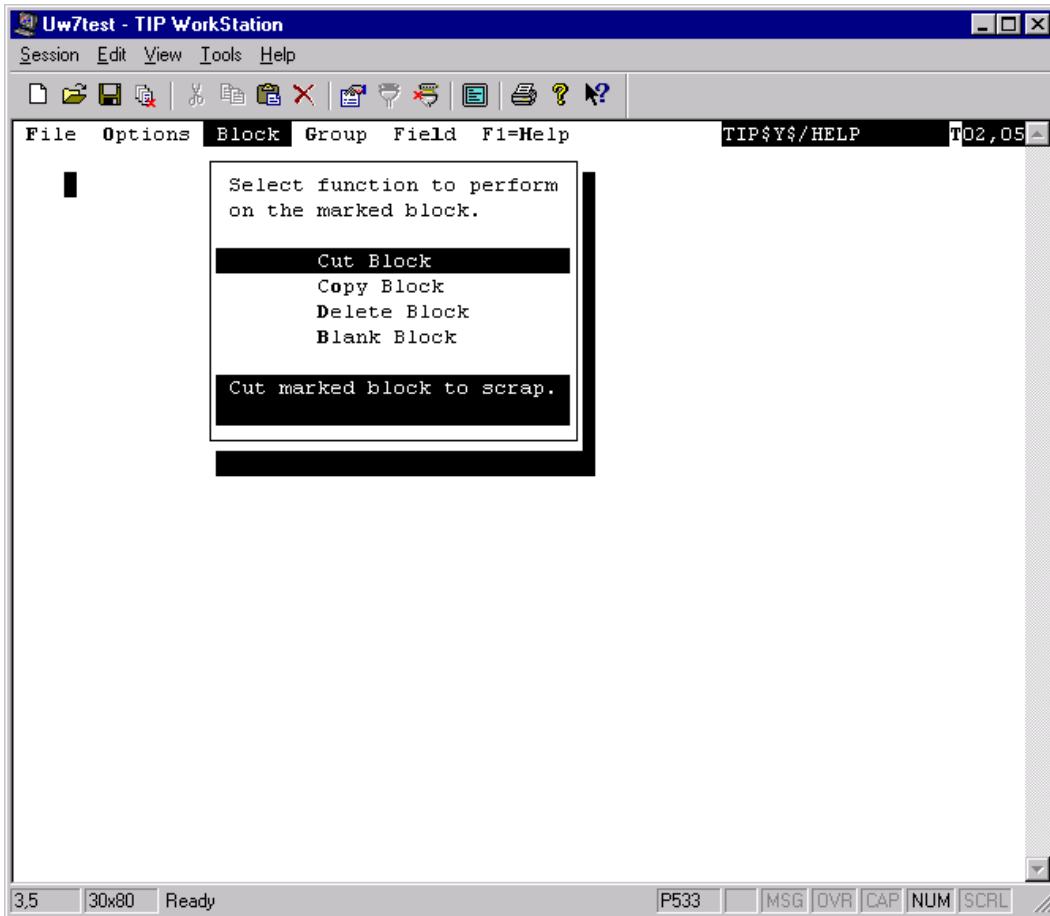


If you selected line blocking, the **pull down** menu will disappear and the line in which the cursor rested will be displayed in reverse video. A line of help will also replace the main tfd horizontal menu which will instruct you to mark the lines to be blocked.

If you selected column blocking, the **pull down** menu will disappear and the current cursor position will be displayed in reverse video. A line of help will also replace the main tfd horizontal menu which will instruct you to mark the columns to be blocked.

The next step is to use the cursor keys to mark the lines or columns you want to block. When you have marked the desired text press **ENTER** to indicate that the marked area is the desired block.

If you have marked a valid block and pressed **ENTER**, tfd will display the following tfd — Block Functions Pulldown Screen containing four possible options:



### Cut Block

This option indicates that the marked block will be removed from the screen and moved to a storage area called "scrap". The lines below the area and columns to the right of the area will be shifted to fill in the vacated area.

### Copy Block

This option indicates that the marked block will not be removed from the screen. tfd will place a copy of the marked block in a storage area referred to as "scrap".

### Delete Block

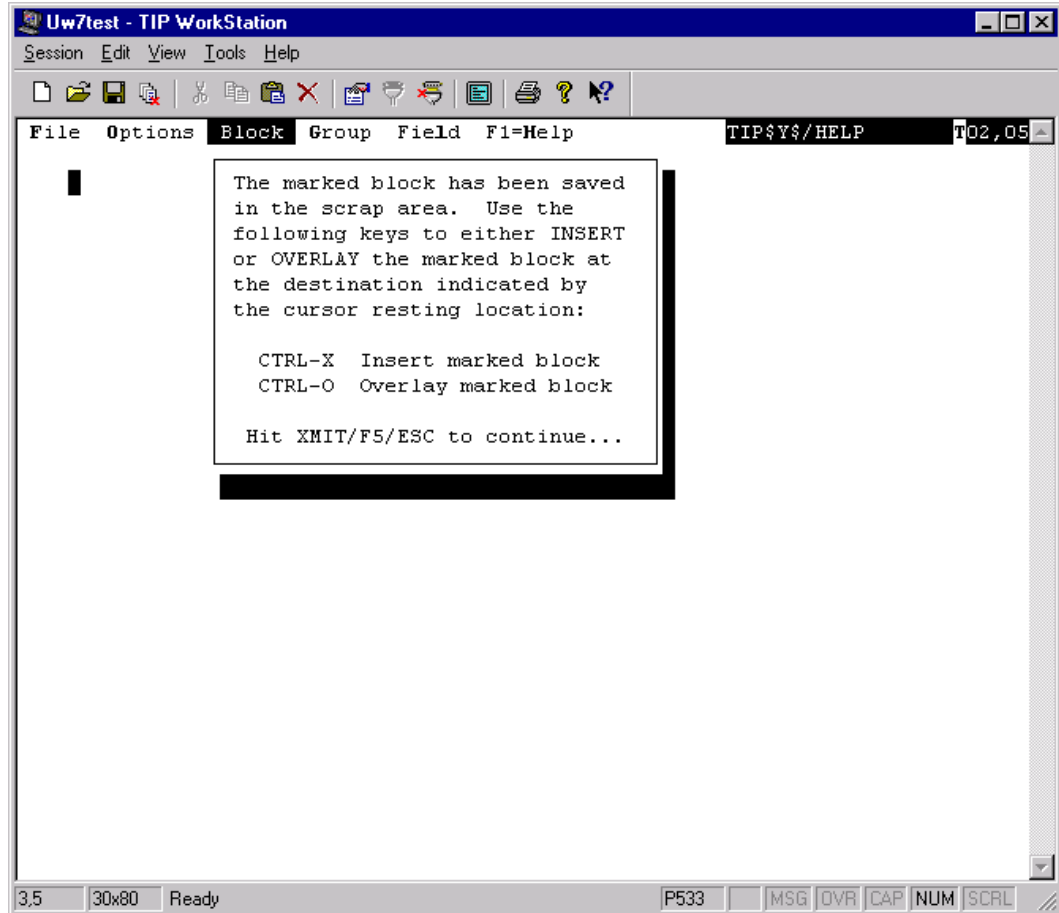
This option indicates that the marked block will be removed from the screen. The lines below the area and columns to the right of the area will be shifted to fill in the vacated area.

### Blank Block

This option indicates the marked area will be cleared and filled with spaces. The remainder of the screen will remain unaffected.

## Specifying Block Destinations

If "Cut Block" or "Copy Block" is selected from the block operations **pull down**, the following screen will be displayed informing that the marked block has been saved in the scrap area:



The marked block in the scrap area can now be either **INSERTED** or **OVERLAYED** at a selected destination and as many times as necessary.

There are two hot keys that control whether to insert or overlay the marked block. They are:

- Ctrl-X — INSERT the marked block
- Ctrl-O — OVERLAY the marked block

Wherever the cursor is at the time either of these hot keys is pressed represents the top left corner of the destination area for the marked block.

If the insert option is selected then:

- for line blocking — all lines from the specified destination to the end of the screen are shifted down to make room for the marked block. Any lines that are shifted off the screen are deleted from the screen format.

- for column blocking — all columns from the specified destination to the right of the screen are shifted right to make room for the marked block. Any columns that are shifted off the screen as well as any fields that are split by the insertion of the marked block of columns are deleted from the screen format.

If the overlay options is selected then:

- for line blocking — all lines in the overlaid area are deleted from the screen format and the marked block is dropped into the empty area.
- for column blocking — all columns in the overlaid area as well as any fields that are split by overlaying the marked block are deleted from the screen format. The marked block of columns is then dropped into the empty area.

Any scrap block of lines or columns must entirely fit on the screen at the indicated destination or the block operation will not be allowed and the following error message will be displayed:

```
Invalid destination for marked text,
strike any key to continue...
```

#### **Additional Considerations:**

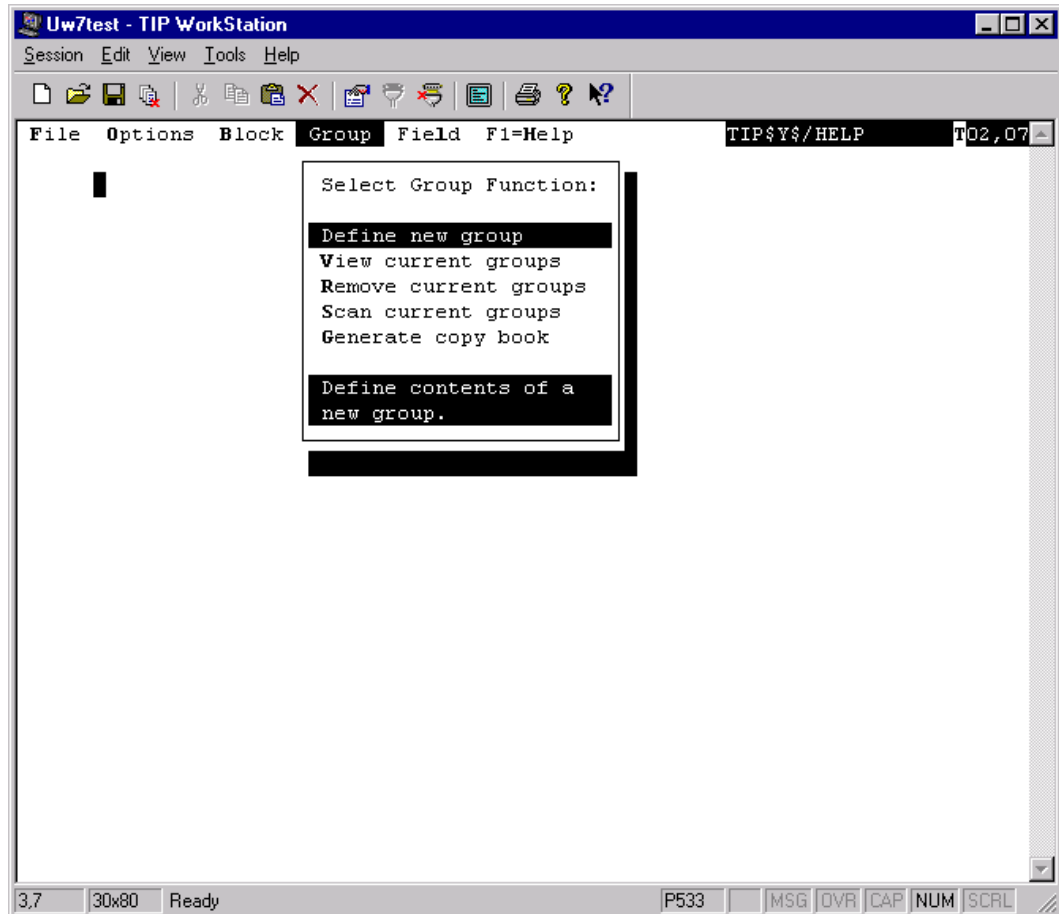
The scrap area remains saved for the duration of the tfd session and across screen formats as well. In this way, an area of one screen can be moved to another screen.

## **Group Operations**

tfd allows for definition and maintaining of groups which contain data fields. These groups can be used when generating COBOL copy books to have a group level item followed by the field names of the contained fields.

Group level names, the number of occurrences of the group, an index name for the group and the encompassing fields can all be controlled through this group operations **pull down**.

To enter into these group operations, select the group operations **pull down** from the tfd — Main Screen horizontal menu which will display the following tfd — Group Operations Pulldown Screen:



The top four selections in the above screen are used to define and maintain the groups and the fields within the group. The last selection is used to define and generate a COBOL copybook.

### Group Definition

By selecting the "Define a Group" function from the Group Operations **pull down**, a new group can be defined with group level name and indication of the encompassing fields.

The position of the cursor is important at the time this **pull down** is selected. The current cursor position should be at the first position of the first field in the group. The group being defined will start at this position and contain all fields until the end position is reached which will be defined after proceeding into the groups operations **pull down**.

Upon selecting the "Define a Group" function, the following directions will be displayed at the top of the screen:

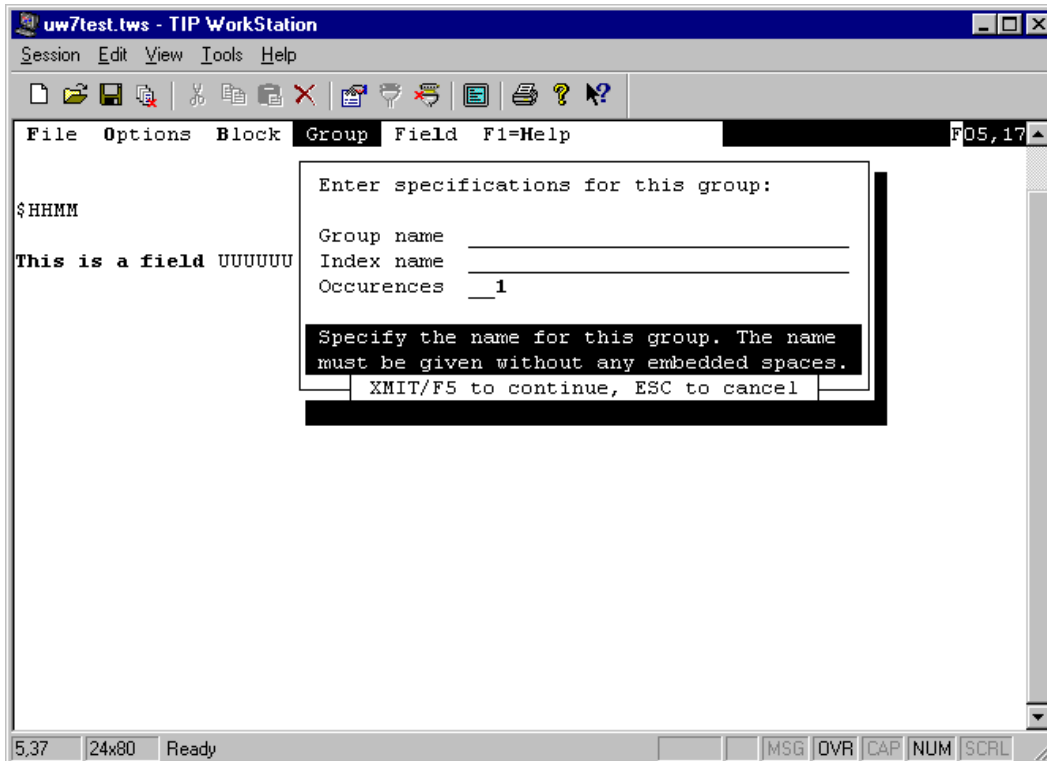
**Use the cursor keys to mark a single occurrence of the fields to be grouped.**

The cursor will also change to reverse video for marking the fields to be in the group. At this point, the cursor keys are to be used to locate the end position of a single occurrence of the group. As the cursor is moved

around the screen, the area between the initial position and the current position will all be changed to reverse video. This area is to indicate a single occurrence of the fields in the group, not the entire group containing all repeating occurrences.

If the starting location is not correct for this group, merely hit **Esc**, position the cursor at the correct starting location and re-start the group definition.

After having marked the end position for a single occurrence of the group, hit **RETURN** and the following tfd — Group Specifications Pulldown Screen will be displayed:



This screen allows you to specify the group name, index name and number of occurrences in this group.

The group name entered will be the group level name used when the copybook is generated and all fields within this group will be at a lower level under this group.

The index name entered, if desired, will be the index name associated with this group in the copybook.

The occurrences field is to indicate how many times the currently blocked single occurrence of the group is to be repeated. This number is the number of occurrences of the marked group including the currently marked single occurrence of the group.

For example, if the group name entered is "ORDER-LINES", the index name entered is "ORDER-IDX" and the number of occurrences is "12", the generated copybook would look similar to:

```
05 ORDER-LINES OCCURS 12 TIMES
INDEXED BY ORDER-IDX.
10 ... PICTURE ...
10 ... PICTURE ...
10 ... PICTURE ...
```

After entering the appropriate information on the group specifications screen, press **F5** to continue which will attempt to define the group and all the occurrences.

If an error occurs when defining a group, the following message will be displayed and group definition will be terminated.

```
Invalid group definition. Strike and key to continue...
```

When defining a group, the marked group must be immediately followed by identically matching fields as the marked group in the same order for the given number of occurrences. If any field does not match a field in the marked group in consecutive order, then group definition is aborted.

As the fields within the group will be the fields in the copybook, all fields from the start of the group to the end of the group must be in consecutive order for the entire group. Consecutive fields means that fields are ordered from the left to right on the screen and wrap around to the next line on the screen and so on.

Groups can be entirely contained within other groups, (that is, nested groups) but they can not overlap other groups or be partially contained in other groups.

Upon successful generation of a group, the following message will be displayed at the top of the screen:

```
The entire group is shown with repeating
fields blinking. Hit any key...
```

This means that the single marked occurrence defined will blink and its corresponding matching fields will be shown. The entire area including the blinking fields defines this group.

Upon pressing any key, the group is then defined and group definition is completed. To view the currently defined groups or modify them, see the following sections for group maintenance.

### Group Viewing

To view any currently defined group, select the "View current groups" function from the group operations **pull down**.

A list of currently defined groups with the number of occurrences in the group, the starting row and the starting column will be displayed. By highlighting the group to be viewed and pressing **ENTER**, that group will be displayed.

The first field in the group to the last field in the single occurrence will be displayed with the following message:

**Use the cursor keys to mark a single occurrence of the fields to be grouped.**

This is the same message as when defining a group which means that this group can now be modified. From this point on, the group definition is the same as explained above to allow for re-definition of this group.

To exit re-definition of a group, press **Esc** and the group will remain unchanged.

### Group Deletion

To delete any currently defined group, select the "Remove current groups" function from the group operations **pull down**.

A list of currently defined groups with the number of occurrences in the group, the starting row and the starting column will be displayed. By pressing the **SPACE BAR**, groups are highlighted which indicates that group is being selected for deletion. After selecting all groups to be deleted, pressing **RETURN** will delete those groups.

To exit group deletion, press **Esc** and the groups will remain unchanged.

### Group Scanning

To scan all currently defined groups, select the "Scan current groups" function from the group operations **pull down**.

The first group currently defined will be displayed with the single occurrence of the group blinking and the remainder of the group highlighted.

The following message will be displayed at the top of the screen:

**Hit the SPACE BAR to loop through existing groups or Esc to cancel.**

At this time, by repeatedly pressing **SPACE BAR**, all groups currently defined will be displayed in order. When all groups have been scanned, or at any time, pressing the **Esc** will terminate the group scanning.

The fast key for scanning groups is **Ctrl-G**.

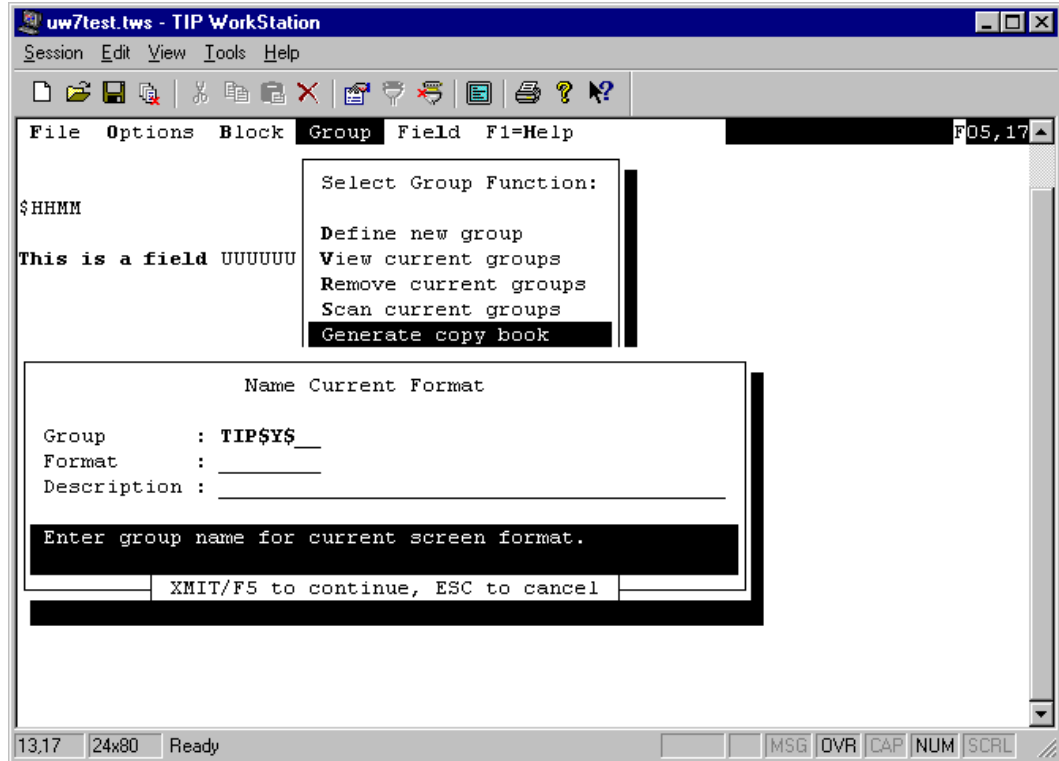
### copybook Generation

**tfd** can generate a COBOL copybook for the currently loaded screen format. To check the copybook generation parameters and generate the



copybook itself, select the "Generate copy book" function from the group operations **pull down**.

The following tfd — copybook Generation Options Screen will be displayed:



This screen allows you to specify the filename for the copybook, a prefix and suffix name for fields, options to generate FCC or cursor modification tables as well as provide the actual copybook generation.

If the current screen format is a new format or is as yet unnamed, a screen will be displayed to prompt for the new name of the current screen. After entering a name for the screen, the above screen will be displayed and processing can continue.

Each of the fields on the above screen is described individually:

### Output file name

Enter in this field the entire path and file name that will contain the generated copybook.

Example:

`/u/tip/work/cbook/payroll.cbk`

### Field name prefix

Enter in this field an optional prefix that will be attached to each field name when the copybook is generated. A dash will be inserted between the prefix and the field name.

**Field name suffix**

Enter in this field an optional suffix that will be attached to each field name when the copybook is generated. A dash will be inserted between the field name and the suffix.

**Include Cursor Modification Table?**

To have a cursor modification table generated with the copybook, enter a 'Y' in this field, otherwise 'N'. Each field in the screen will have an entry in the cursor modification table suffixed by a "-C".

**Include FCC Modification Table?**

To have a FCC modification table generated with the copybook, enter a 'Y' in this field, otherwise 'N'. Each field in the screen will have an entry in the FCC modification table suffixed by a "-F".

**copybook Generation Type**

By pressing the **Pgup** and **Pgdn** keys, the valid options for the type of copybook will be toggled.

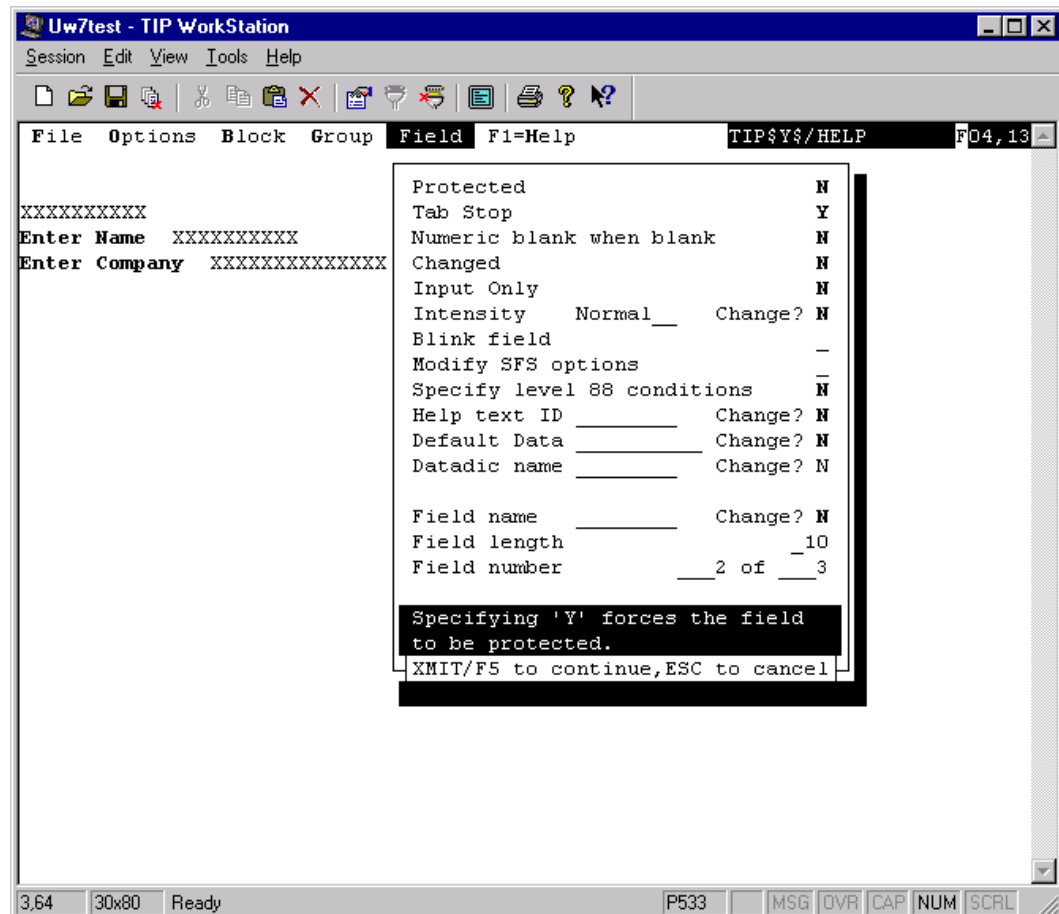
**Generate copybook**

When all of the above options are set appropriately, the copybook can be generated by entering a 'Y' in this field. Upon completion of the copybook generation, a message will be displayed indicating successful completion or any errors encountered.

**Changing Field Characteristics**

Often after a field is defined, you may want to view or change its characteristics. **tfd** allows you to view or change the characteristics of any field on the screen, at any time.

You can do this by accessing the field **pull down** from the **tfd** — Main Screen horizontal menu. **tfd** will display the following Field Pulldown:



The above screen displays many options of the current field:

- The top fields from 'Protected' to 'Blink field' in the screen allow for setting specific attributes of the current field and correspond to the default attributes defined when the field was created. See Default Field Characteristics for more information.
- The next field allows for SFS emulation options to be entered. Enter 'Y' to view and modify the current options. This field will be protected unless SFS emulation has been enabled through the options pull down.
- The next field allows for level 88 items to be entered. Enter 'Y' to bring up a menu that allows for creating, modifying and deleting of level 88 items. When the copybook is generated, these level 88 items with their values will be generated along with the current field.
- The next three fields allow a help text identifier, default data and a data dictionary name to be assigned to the current field. These fields will be protected if the field is a special definition field. See Entering Default Data below and Attaching Help to a Field for additional information.
- The next field handles the naming of the current field. Enter 'Y' to bring up a prompt to enter the field name. This name will be used with this field when the copybook is generated.

- The last three fields are protected and show further information concerning the current field such as:

**Field length:**

The length in bytes of the current field.

**Current field number:**

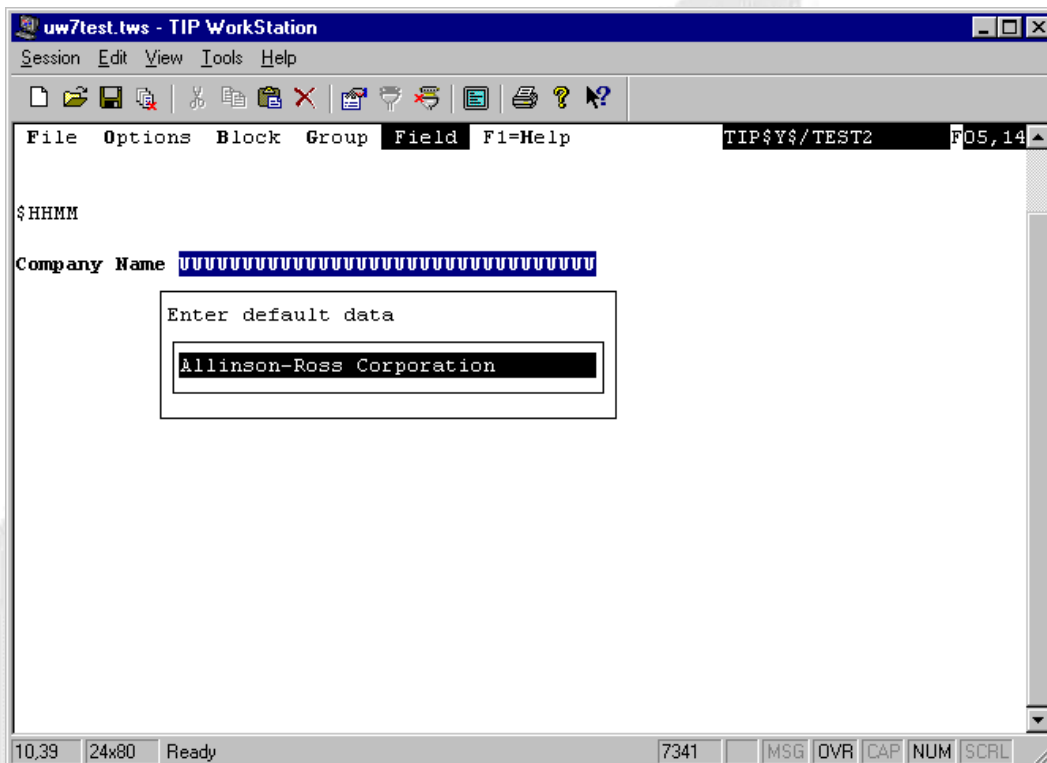
The number of the current field in relation to the other fields. Lower numbered fields start at the top left corner of the screen and increase towards the bottom right.

**Total number of fields:**

The total number of data fields defined in the current format.

### Entering Default Data

tfd allows you to enter default data into a field at any time. To enter default data, position the cursor on the field and then go to the “Field” **pull down** by using the highlight bar. You must select the “Default Data” option and enter “Y”. An example of entering default data is shown on the following tfd — Default Data Screen:



The prompt will be directly above (or below if the field is near the top of the screen) the current field and the field will be flashing.

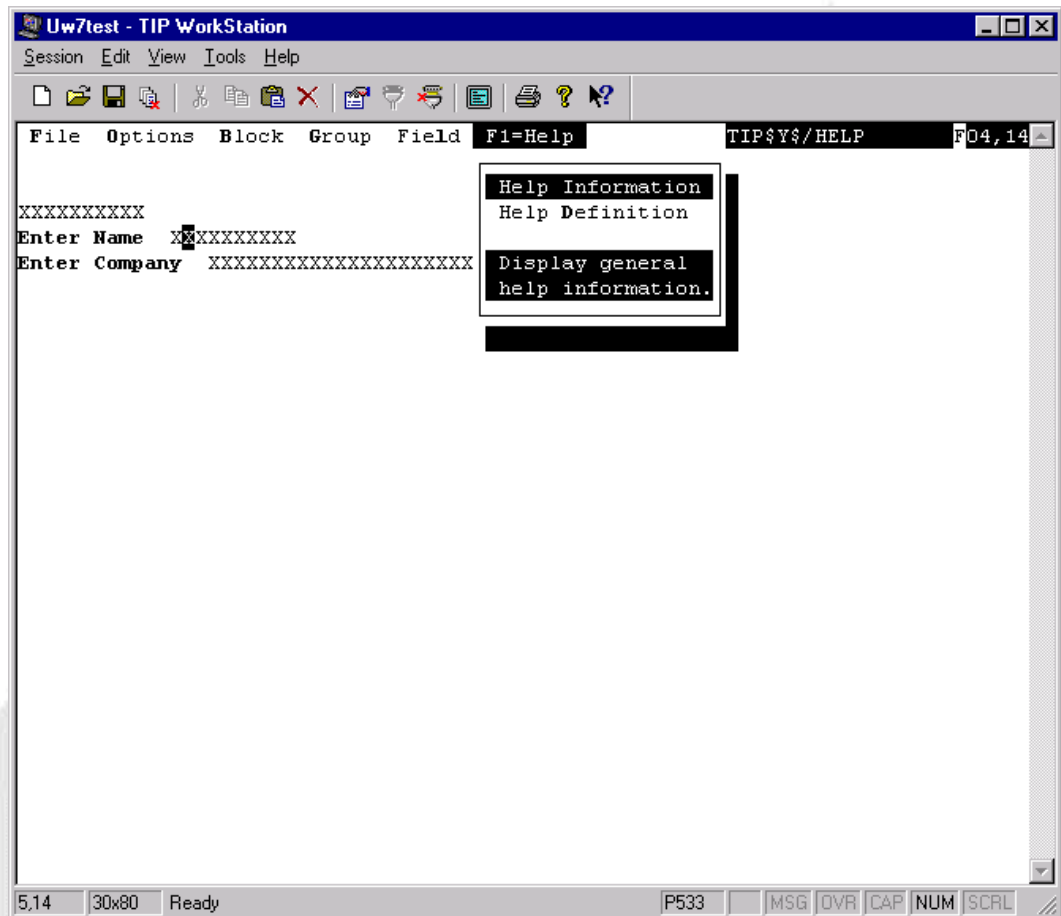
The current default data (if any) will be displayed in this window and you can view it or update it as required.

In the example illustrated above, you are viewing the contents of the field, which now contains the default data "Allinson-Ross Corporation". This data can be viewed and/or updated by using this default data window.

## The Help System

The following sections describe the TIP/ix Context-Sensitive Help System.

To bring up the help system definition and configuration screens from tfd, select the "F1=Help" selection from the main tfd menu bar. By selecting this option, either from the menu bar or by pressing the **F1** key from the screen editing area, the following tfd — Help System Selection Screen will be displayed:

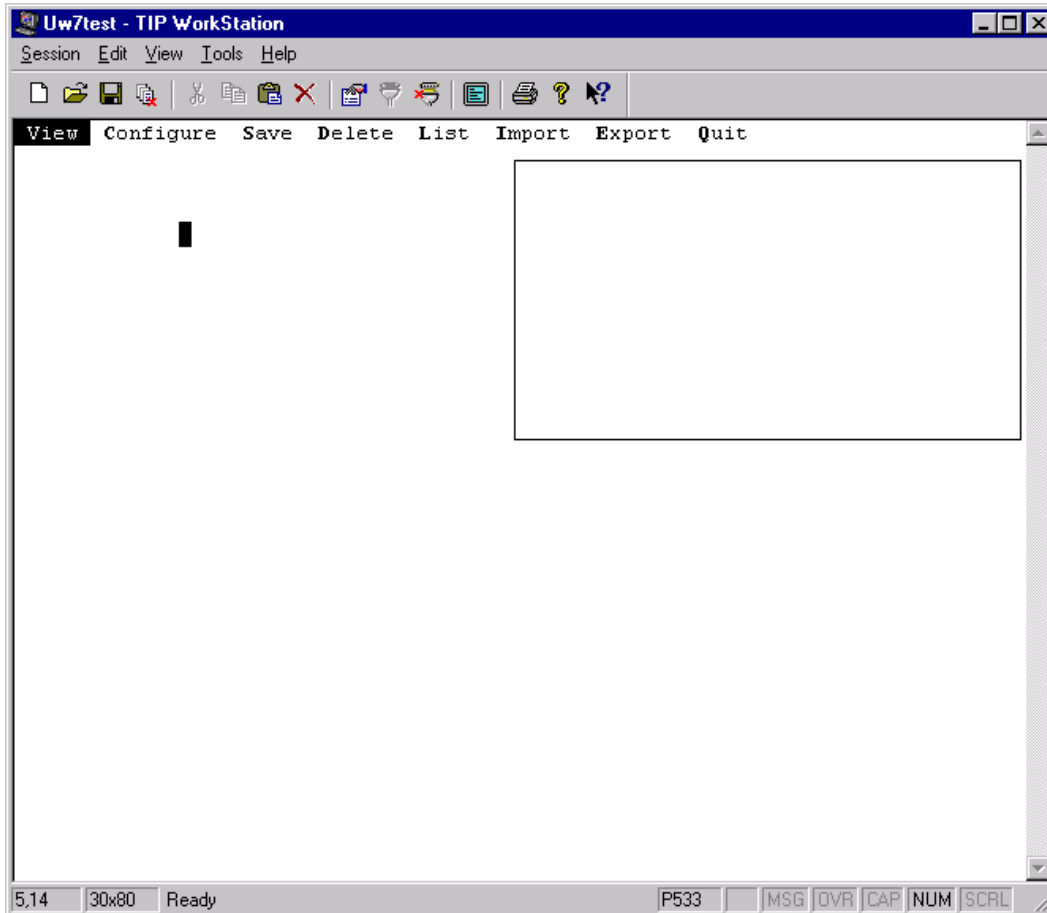


The above screen provides a selection to display general help information and a selection to bring up the tfd — Help Definition Screen as described below.

### Help Definition Screen

This utility screen allows you to define, configure and manipulate help text. The various tools you need to deal with the help text are presented

in a horizontal menu at the top of the following tfd — Help Definition Screen:



The following sections describe the tools available from the horizontal menu at the top of this screen to perform various tasks related to manipulating and maintaining help text.

### Defining Help Text

Before the help system can display help text, the text must be defined to the context-sensitive help system. To do this, perform the following:

1. Select the **Import** option from the main help menu bar and then tfd will prompt you to enter the name of a UNIX text file.
2. Enter the name of a file that contains the ASCII help text and press **ENTER**.
3. tfd will display the text from the file in a window and highlight the first line of text. This window displays the help text as it will appear from an application. You may configure the location, size, border type, and colors used to display this window as described in the upcoming sections.

4. At this point, pressing the **Esc** key will toggle you from the main menu bar to the help text and back.
5. You must now save the imported help in the screen library. Use the **Save** option available from the main help menu bar to perform this task. tfd will prompt you for the name that the text is to be saved under and will allow you to enter a description for the help text. The help name can be up to eight characters in length and any alphanumeric characters can be used.

After the name and description have been entered, press **Esc** to complete the definition of help text. This name is referred to in this manual as the "Help Text Identifier". The help text identifier is used to identify the help text to tfd, and to the Help System.

### Viewing Help Text

You can view any help text that has been saved in the screen library by using the **View** option from the main help menu bar. If you select this option, you will be prompted for a Help Text Identifier. Enter a Help Text Identifier if you know the name or press the **ENTER** key to select from a list of already defined valid identifiers.

If you entered a defined Identifier name, the Help text associated with the identifier is displayed in the window. If the identifier is not defined, an appropriate informational message is displayed.

If you simply pressed **ENTER** in response to the prompt for a help identifier, tfd will present you with a "point and shoot" list of valid identifiers to choose from. Use the **,Pgup** or **Pgdn** keys to scroll through the list until you find the identifier you want.

Select the desired identifier name by placing the cursor on its line and pressing **ENTER**. The help system will display the help text associated with the identifier name in the help window.

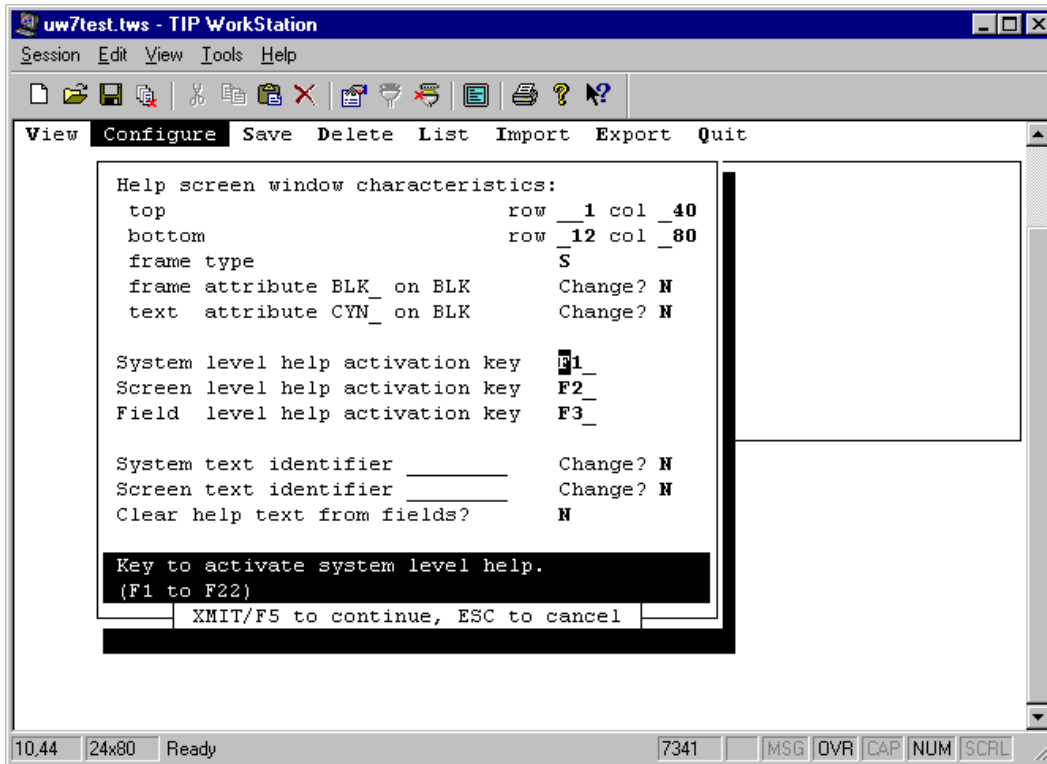
### Attaching Help to a Field

Once you define a field, you can view and edit its various characteristics by using the Field **pull down** of the tfd — Main Screen.

You can enter a Help Text Identifier (see *Defining Help Text* above) by putting the cursor, on the tfd — Main Screen, in the desired field, and then activating the **Field pull down**. If you then move the cursor to the "Change" field beside the "Help text ID" field and enter "Y", you can specify the help text identifier directly, or press **Enter**, and tfd will display a list of valid Help IDs you can select from.

### Configuring the Help System

The attributes of the help window can be configured using the screen presented (menu will be different for a monochrome monitor [see below](#)).



The following describes each of the fields:

#### Window top row \_\_ col \_\_

Top left hand corner of help window; the row must be in the range of 1 — 253 and the column must be in the range of 1 — 253.

#### Window bottom row \_\_ col \_\_

Bottom right hand corner of help window; the row must be in the range of 3 — 255 and the column must be in the range of 3 — 255.

#### Window frame type

'S' for single line border, or 'D' for double line border.

#### Window frame attribute \_\_ on \_\_ Change? N

This field allows you to set the help window frame attribute.

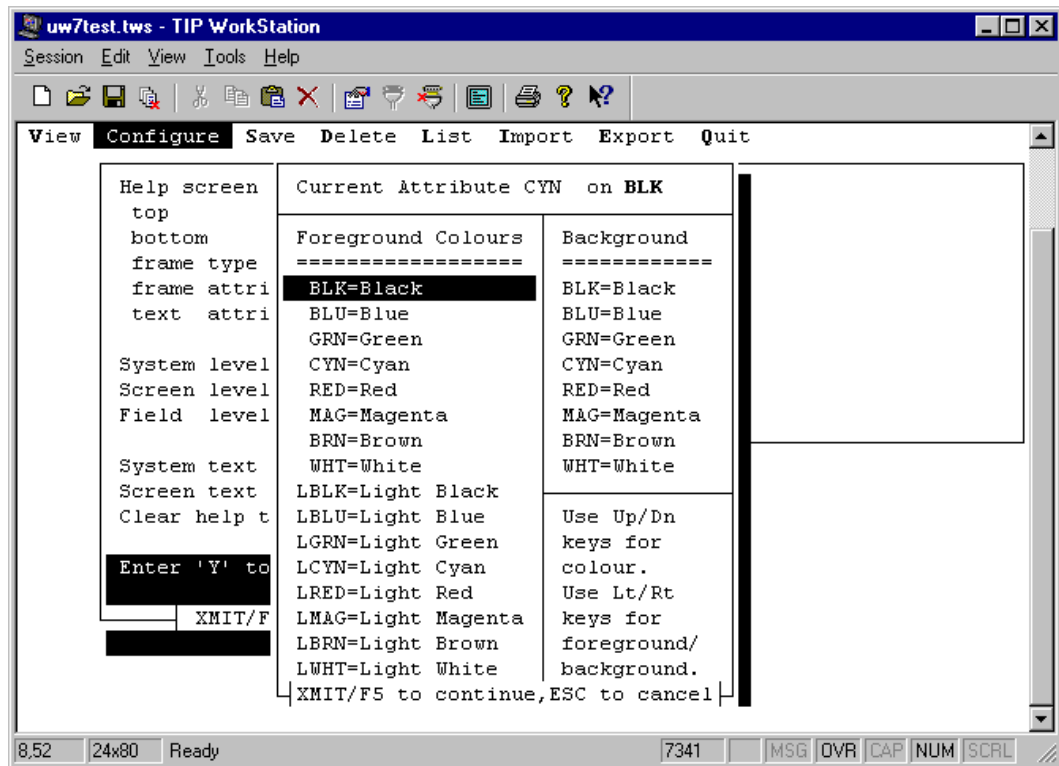
#### Window text attribute \_\_ on \_\_ Change? N

This field allows you to set the help window text attribute.

If you enter "Y" to change either the window frame attribute or the window text attribute, tfd will display one of two possible screens, depending on whether you have defined the format for a color or monochrome monitor.

If the format is being defined for a color monitor, the following tfd — Help Configuration (Color Attribute Screen) is presented:

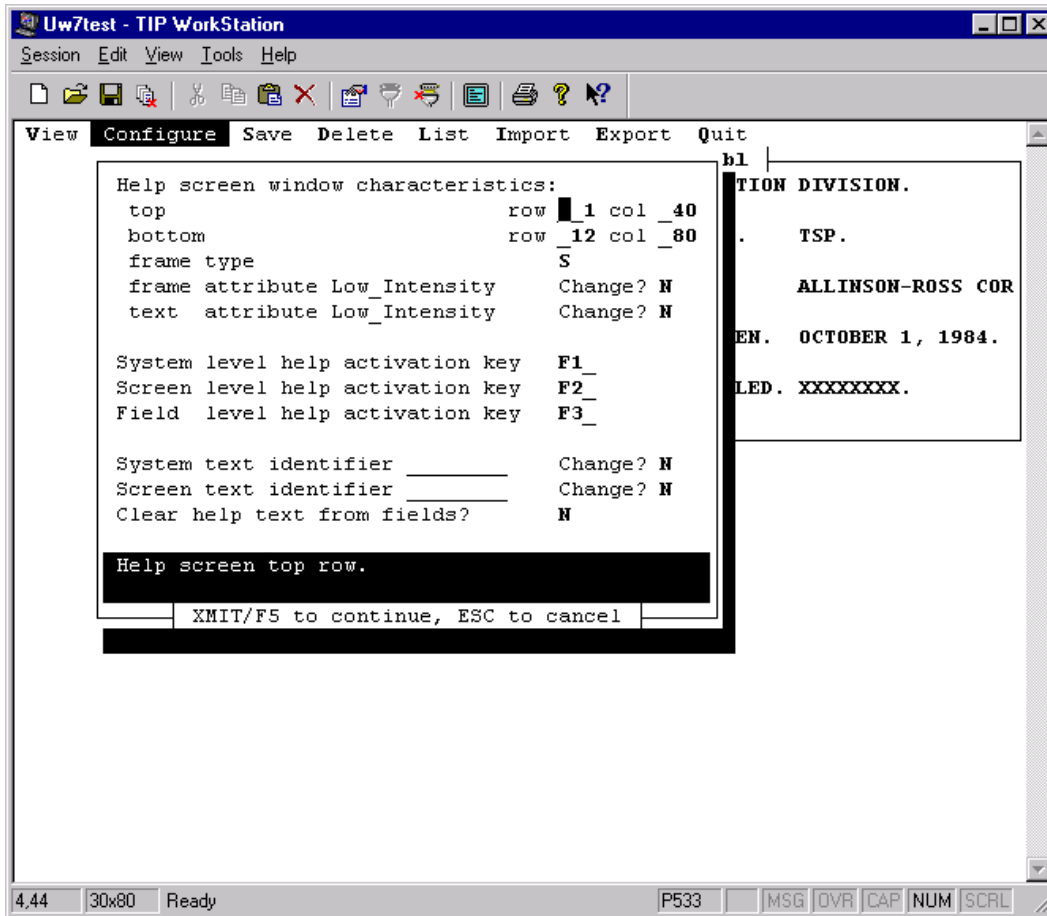




The above screen presents a list of valid foreground colors and a list of valid background colors from which to choose. Use the **↑** and **↓** keys to highlight one of the colors and press **ENTER** to select that color. Use the **←** and **→** keys to select from either the foreground color table or the background color table.

The current attribute is displayed at the top of the screen and updated when a color combination is selected. Press **XMIT** or **F5** to complete your changes.

If the format is not to be used on a color monitor, the following tfd — Help Configuration (Monochrome Attribute Screen) is presented:



The above screen presents a list of valid monochrome attributes to choose from. Use the and keys to highlight one of the three options and press **ENTER** to select that monochrome option.

#### System help activation key \_\_\_\_

This function key activates the system, or application level help. The possible values for the function key are **F1** through **F22**.

#### Screen help activation key \_\_\_\_

This function key activates screen level help. The possible values for the function keys are **F1** through **F22**.

#### Field help activation key \_\_\_\_

This function key activates field level help. The possible values for the function keys are **F1** through **F22**.

#### System text identifier \_\_\_\_\_ Change? N

This field allows you to set help text for system or application level help.

If you enter "Y" in this field tfd will display a list of valid help identifiers, from which you can select one. See [Defining Help Text](#) for more information.

**Screen text identifier \_\_\_\_\_ Change? N**

This field allows you to set help text for screen level help.

If you enter "Y" in this field tfd will display a list of valid help identifiers, from which you can select one. See [Defining Help Text](#) for more information.

**Clear help text from fields? Y / N**

This field allows you to clear all help text identifiers already associated with fields in the currently loaded screen format.

**Saving Help Text**

Once help text has been loaded into the display window and configured as necessary, selecting the **Save** option from the main help definition screen will store the help text in the help system.

You will be prompted for a help text identifier naming the stored help text and for an optional description of the help text.

**Deleting Help Text**

You can delete help text from the screen library by using the **Delete** option at the top of the main help definition screen. After you select the **Delete** option, the help system will prompt you for a Help Text Identifier. You may enter a Help Identifier and press the **ENTER** key, or just press the **ENTER** key.

If you enter a defined Identifier name and press the **ENTER** key, the help system will delete the Help text associated with this identifier. If the identifier has not been previously defined, the help system will display an appropriate message.

If you press **ENTER** in response to the prompt for a help identifier, the help system will display a list of all currently defined help identifiers. You can scroll through this list using the **and** keys, or **Pgup** and **Pgdn**.

To delete a particular help text item, place the cursor on the line with the appropriate identifier and press **ENTER**. To delete multiple help items, flag them by moving the cursor to the line that contains the appropriate help identifier name and press the **SPACE BAR**.

The **SPACE BAR** acts as a toggle to flag or unflag help identifiers. When you have flagged all the desired help items, delete them by pressing **ENTER**.

**Listing Help Text**

tfd can produce a list of help identifiers. If you select the **List** option on the main help definition screen, the help system will prompt you for a help identifier name. You can type in an actual identifier name or one that contains wildcard characters.

If you just press **ENTER** when prompted for the help identifier name, then `tfd` will list all help text identifiers.

### Importing Help Text

This option allows you to import a UNIX ASCII text file into the help system. Selecting this option will produce a prompt into which you can enter the name of the file to be imported. Once the file is imported, it will be displayed in the help definition window.

### Exporting Help Text

To copy a piece of Help text into a UNIX text file, use the **Export** option on the main help definition screen. To export help text, you must first display the text in the help window on the *Help Definition Screen*. To display help text in the help window, use the **View** option (See *Viewing Help Text* on page 646.) Once the help text is displayed in the window, you can export it to a file by selecting the **Export** option.

When you select this option, the help system will prompt you for the file name that the help text is to be copied to. Enter the file name and press **ENTER** to copy the help text into the named file.

### Exiting the Help Definition Screen

You can return to the main `tfd` screen by selecting **Quit** on the main help definition screen.

---

## tipctl - System Startup/Shutdown

The **tipctl** program is used to startup or shutdown the TIP/ix system. This program is normally run by the system administrator (or as part of the computer boot procedures). Users of the TIP/ix system seldom need to make use of **tipctl**.

#### Syntax:

```
tipctl [a | d | e]
tipctl apb "message"
tipctl b[oot] [cold | init | warm |
TIPDATE=CCYYMMDDJJJ]
tipctl eoj [timeout] [wait time-to-eoj]
tipctl k [user id]
tipctl msg dest "message"
tipctl r[estart]
tipctl s[hutdown]
tipctl stop
```

**Where:**

- a** To display TIP/ix system status.
- apb** To broadcast a message to all users.
- b[oot]** To start (boot) the TIP/ix system.
  - cold** Initialize the QBL files, with *no recovery*, then continue running.
  - init** Initialize the QBL and JRN files, with *no recovery*, then continue running.
  - warm** Do any pending rollback recovery, *then shut down*.

**TIPDATE =CCYYMMDDJJJ**

This is used for Y2K testing. When this variable is set, all tipix screens and data will use this date as default. If not set, TIP/ix will use the system date.

- CC** Century
- YY** Year
- MM** Month
- DD** Day
- JJJ** Julian date

Example:

```
tipctl boot TIPDATE=20000112012
```

The default is to do any pending rollback recovery, then continue running.

- d** To disable the running TIP/ix system (do not allow logins).
- e** To enable a disabled TIP/ix system (allow logins).

**ej [option]**

To schedule a TIP/ix shutdown (at a later time). If you do not specify a *timeout* or *time-to-ej* parameter, **ej** works like **s[hutdown]**.

To use this parameter, you must meet the shutdown security requirements (described below).

**k [user id]**

Forcibly terminate (kill) users, but leave TIP/ix running.

If user id is specified, all TIP/ix sessions for that user are terminated.

If user id is not specified, all TIP/ix sessions are terminated.

To use this parameter, you must meet the shutdown security requirements (described below).

**msg** To send a message to a user.

**dest** Username or terminal name.

**r[estart]**

To do a shutdown then a **tipctl boot**.

To use this parameter, you must meet the shutdown security requirements (described below).

**s[hutdown]**

Shutdown the TIP/ix system. Check for active users.

To use this parameter, you must meet the shutdown security requirements (described below).

**stop** Shutdown TIP/ix immediately. Do not wait for users to log off.

To use this parameter, you must meet the shutdown security requirements (described below).

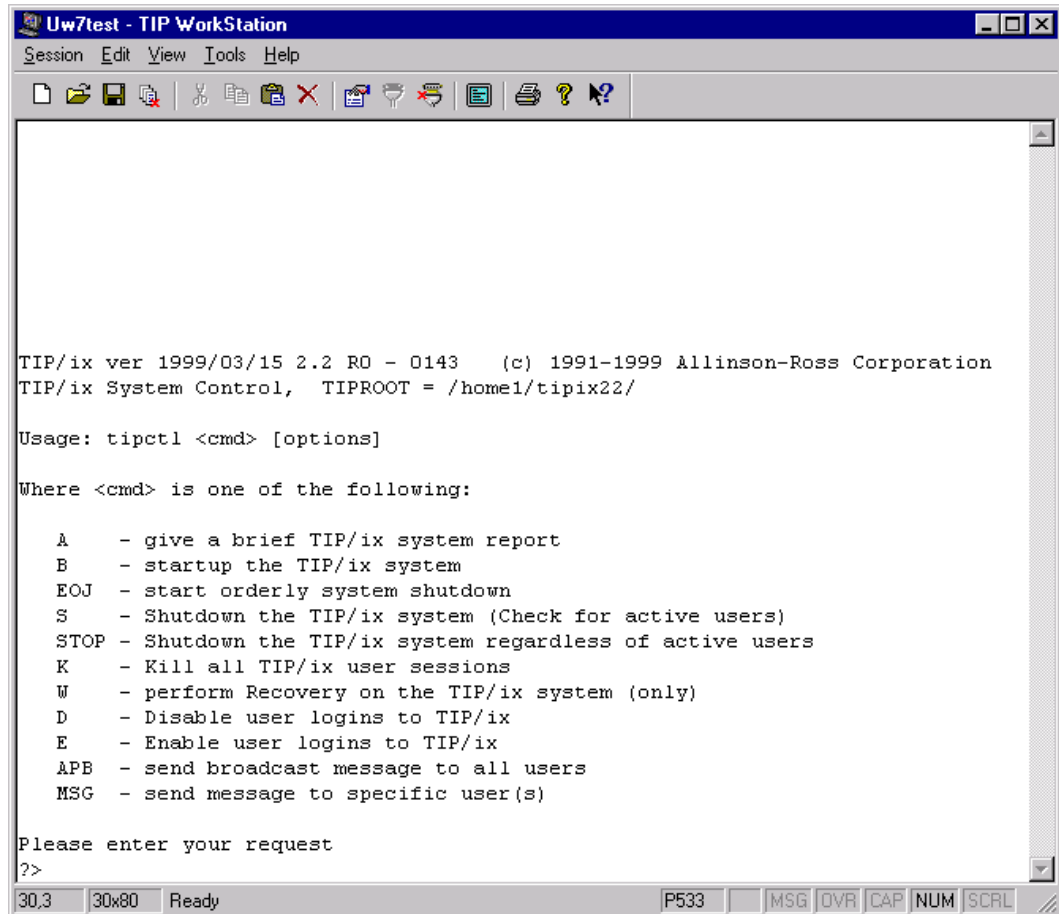
## Shutdown Security Requirements

To shutdown the TIP/ix system, you must be logged onto Unix:

- as a user id that has at least TIP/ix MAST level security, or
- as root, or
- as the TIP/ix administrative user id (usually tipixusr).

## Prompt Screen

If **tipctl** is run without command line parameters, the following prompt screen is displayed; a command may be entered at that time:



```
Uw7test - TIP WorkStation
Session Edit View Tools Help

TIP/ix ver 1999/03/15 2.2 R0 - 0143 (c) 1991-1999 Allinson-Ross Corporation
TIP/ix System Control, TIPROOT = /home1/tipix22/

Usage: tipctl <cmd> [options]

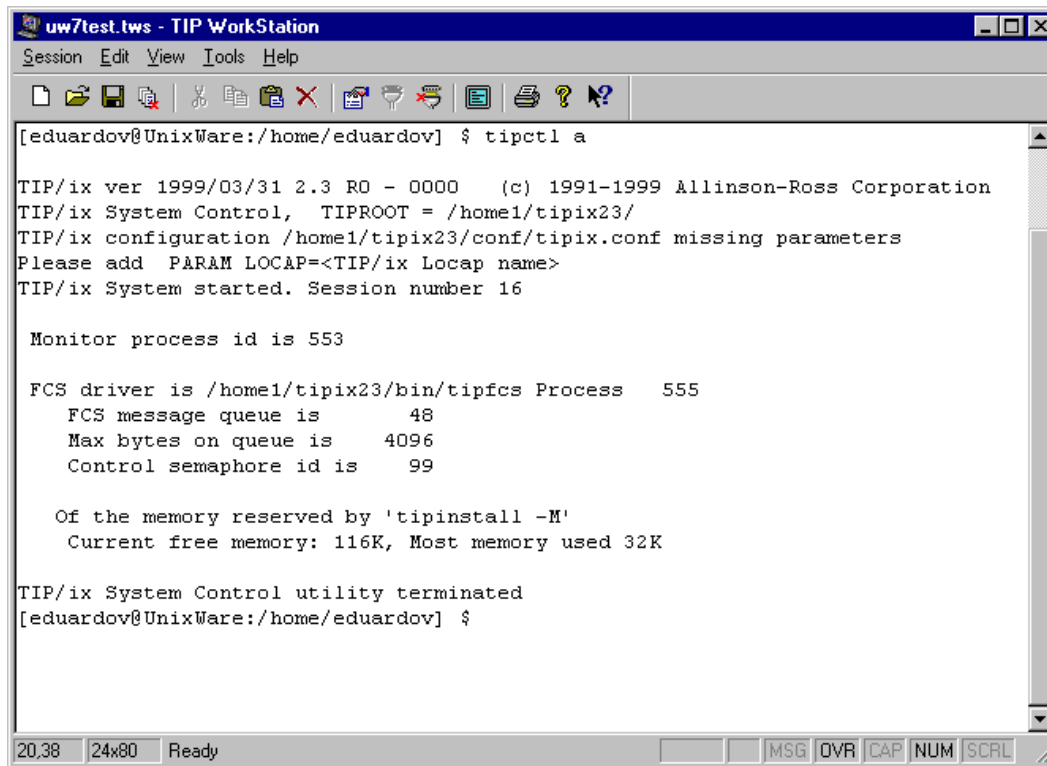
Where <cmd> is one of the following:

A - give a brief TIP/ix system report
B - startup the TIP/ix system
EOJ - start orderly system shutdown
S - Shutdown the TIP/ix system (Check for active users)
STOP - Shutdown the TIP/ix system regardless of active users
K - Kill all TIP/ix user sessions
W - perform Recovery on the TIP/ix system (only)
D - Disable user logins to TIP/ix
E - Enable user logins to TIP/ix
APB - send broadcast message to all users
MSG - send message to specific user(s)

Please enter your request
?>
```

## a - system status

To obtain a report on the TIP/ix system status, use the tipctl "a" command. The "a" command displays a report similar to this:



```
uw7test.tws - TIP WorkStation
Session Edit View Tools Help
[eduardov@UnixWare:/home/eduardov] $ tipctl a

TIP/ix ver 1999/03/31 2.3 RO - 0000 (c) 1991-1999 Allinson-Ross Corporation
TIP/ix System Control, TIPROOT = /home1/tipix23/
TIP/ix configuration /home1/tipix23/conf/tipix.conf missing parameters
Please add PARAM LOCAP=<TIP/ix Locap name>
TIP/ix System started. Session number 16

Monitor process id is 553

FCS driver is /home1/tipix23/bin/tipfcs Process 555
FCS message queue is 48
Max bytes on queue is 4096
Control semaphore id is 99

Of the memory reserved by 'tipinstall -M'
Current free memory: 116K, Most memory used 32K

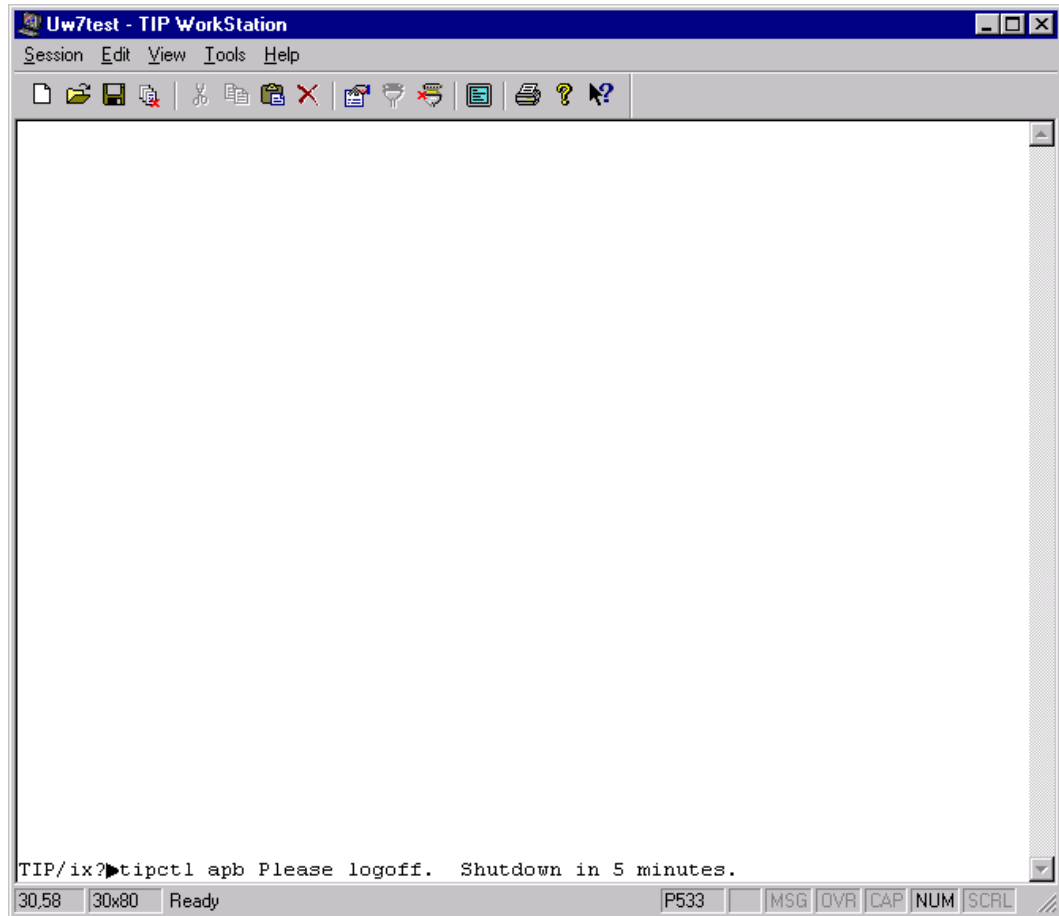
TIP/ix System Control utility terminated
[eduardov@UnixWare:/home/eduardov] $

20:38 24x80 Ready MSG OVR CAP NUM SCRL
```

## apb - Broadcast All Points Bulletin

To send a message to all users that are logged on, use the **tipctl apb** command.





## b - boot TIP/ix

Before users can run the TIP/ix command shell (named "tipix"), the TIP/ix system itself must first be started (booted):

As the system is starting, messages similar to the following are displayed:

```

Uw7test - TIP WorkStation
Session Edit View Tools Help
[ScottC@UnixWare:/home/ScottC] $ tipctl b

TIP/ix ver 1999/03/15 2.2 R0 - 0143 (c) 1991-1999 Allinson-Ross Corporation
TIP/ix System Control, TIPROOT = /home1/tipix22/
Allinson-Ross Registered for the following products
 Concurrent Serial
Product Users Expiry Number
TIP/ix 10 2000/04 01388231
TIP/dbi 10 2000/04 01388231
HSP/80 10 2000/04 01388231
JRN file /home1/tipix22/tipfiles/tipix.jrn0 is now 161 K bytes

Starting system process ... monitor . FCS . Ready at 10:35 on 1999/03/31

Waiting for FCS to complete any recovery:
#*** TIP/ix FCS server tipfcs0 as Process 939 on Q98 ***
Read 0 before images from QBL file 1
TIP/ix local LOCAP name is UW7TEST o

```

Important points to notice are the directory information on the first output line, the TIP/ix serial number and the revision level of TIP/ix.

At startup TIP/ix displays messages regarding any configuration options that may need changing or that must be changed. If an option must be changed and TIP/ix will not start up then you should follow the advice from the startup messages and try again.

Restarting TIP/ix does **not** cause files that have been marked as offline (with [fclose](#)) to be brought online. This is to prevent files from coming back online due to an unscheduled shutdown and startup of TIP/ix. Once a file is closed with [fclose](#), it remains offline until a subsequent [fopen](#).

This is different from TIP/30 where all files are marked as online when TIP/30 starts up. If you want to bring all TIP/ix files back online when TIP/ix starts up (TIP/30 style), do this:

```

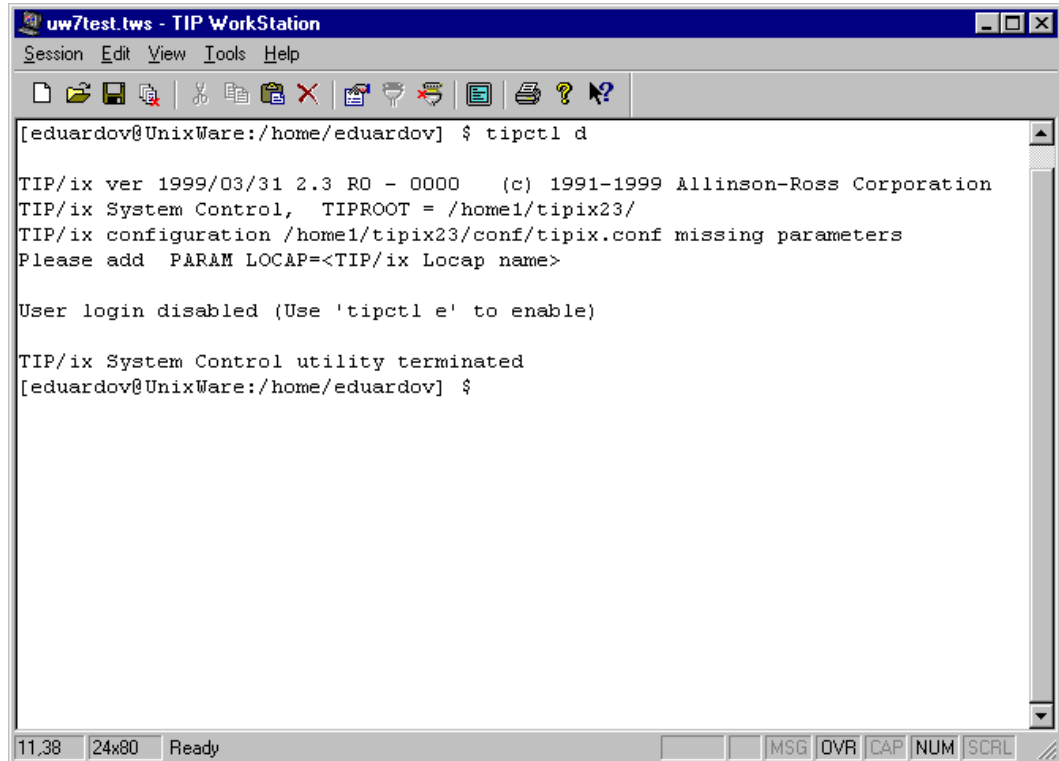
fopen *
tipctl b

```

These two commands could be combined into a startup script. Such a script could include any other housekeeping activities associated with starting up TIP/ix (perhaps removing or creating temporary files).

## d - disable logins

The tipctl "d" command can be used to disable logins; that is, once the "d" command is issued, no new use of the TIP/ix command shell is permitted until a tipctl "e" command is executed.



```
uw7test.tws - TIP WorkStation
Session Edit View Tools Help
[eduardov@UnixWare:/home/eduardov] $ tipctl d
TIP/ix ver 1999/03/31 2.3 RO - 0000 (c) 1991-1999 Allinson-Ross Corporation
TIP/ix System Control, TIPROOT = /home1/tipix23/
TIP/ix configuration /home1/tipix23/conf/tipix.conf missing parameters
Please add PARAM LOCAP=<TIP/ix Locap name>

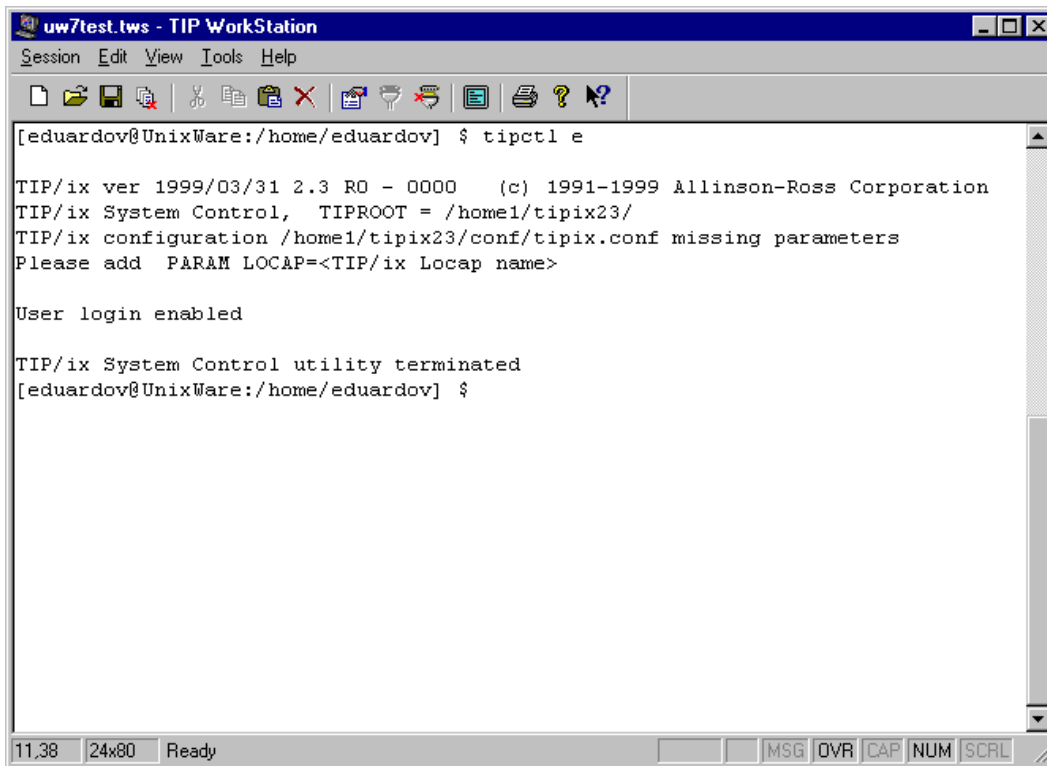
User login disabled (Use 'tipctl e' to enable)

TIP/ix System Control utility terminated
[eduardov@UnixWare:/home/eduardov] $
```

This command can be useful to inhibit users from using the TIP/ix system while some debugging or maintenance activity is completed.

## e - enable logins

The tipctl "e" command can be used to enable logins that were previously prohibited by the use of the tipctl "d" command.



```

uw7test.tws - TIP WorkStation
Session Edit View Tools Help
[eduardov@UnixWare:/home/eduardov] $ tipctl e

TIP/ix ver 1999/03/31 2.3 RO - 0000 (c) 1991-1999 Allinson-Ross Corporation
TIP/ix System Control, TIPROOT = /home1/tipix23/
TIP/ix configuration /home1/tipix23/conf/tipix.conf missing parameters
Please add PARAM LOCAP=<TIP/ix Locap name>

User login enabled

TIP/ix System Control utility terminated
[eduardov@UnixWare:/home/eduardov] $

```

## ej - Schedule a TIP/ix Shutdown

The **tipctl ej** command starts end-of-job processing for the TIP/ix system. TIP/ix does not allow any new logins to occur and waits for transactions that are executing to finish. The **tipctl ej** command also causes TIP/ix to set the status code **PIB-EOJ-PENDING** in the Program Information Block (PIB) of all TIP/ix programs that are executing. *Well-behaved TIP/ix programs should periodically check this flag to determine whether system shutdown has been requested.*

When there are no more users remaining on the system, the **SHUTDOWN** script (if one exists in the TIPROOT directory) will be executed. When the SHUTDOWN script has finished, all files are closed and TIP/ix terminates normally.

### Syntax:

```
tipctl ej [timeout] [WAIT time-to-ej]
```

### Where:

#### timeout

The number of minutes to wait for transactions to complete. If a TIP/ix program is waiting (via TIPTIMER), the wait time will be adjusted to be the lesser of the actual time remaining and this value.

**WAIT** This indicates that the EOJ program is to take effect on a delayed basis. That is, it will start after time-to-eoj minutes.

**time-to-eoj**

The number of minutes to defer EOJ processing.

This command runs under the UNIX prompt. If WAIT is specified, then EOJ will run in the background.

If EOJ is specified without any arguments, then a forced shutdown of TIP/ix will occur.

**Examples**

**EOJ** Force system shutdown immediately (the user is prompted).

**EOJ 1**

Shut down the system immediately; reduce all TIPTIMERS to 1 minute.

**EOJ WAIT 30**

Force system shutdown in 30 minutes.

**EOJ 2 WAIT 15**

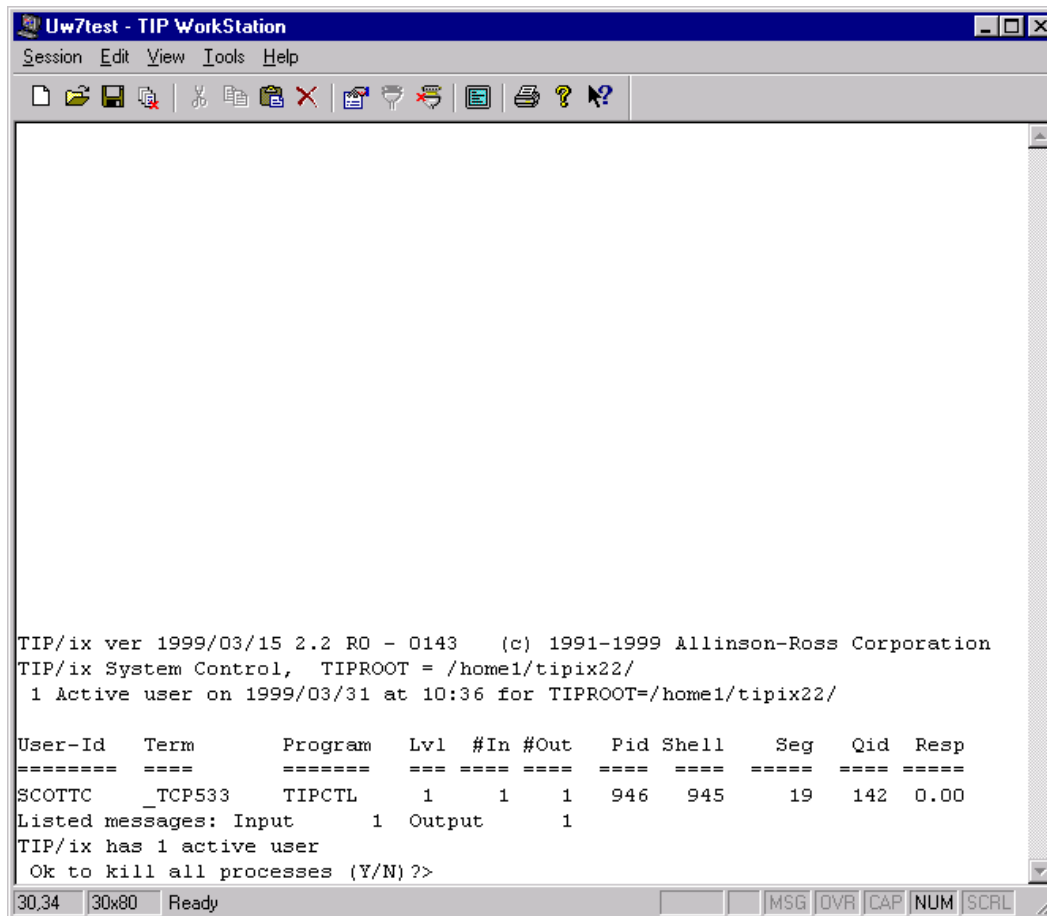
Start system shutdown in 15 minutes; reduce all TIPTIMERS to 2 minutes.

## **k - kill user sessions**

Forcibly terminate (kill) users, but leave TIP/ix running.

If user id is specified, all TIP/ix sessions for that user are terminated.

If user id is not specified, all TIP/ix sessions are terminated. This command can be useful to remove all users from the TIP/ix system. Most often, it is preceded by a "tipctl d" command to prevent any new users from accessing the TIP/ix system.



```

Uw7test - TIP WorkStation
Session Edit View Tools Help
TIP/ix ver 1999/03/15 2.2 R0 - 0143 (c) 1991-1999 Allinson-Ross Corporation
TIP/ix System Control, TIPROOT = /home1/tipix22/
 1 Active user on 1999/03/31 at 10:36 for TIPROOT=/home1/tipix22/

User-Id Term Program Lvl #In #Out Pid Shell Seg Qid Resp
===== =====
SCOTTTC _TCP533 TIPCTL 1 1 1 946 945 19 142 0.00
Listed messages: Input 1 Output 1
TIP/ix has 1 active user
Ok to kill all processes (Y/N)?>
30.34 30x80 Ready MSG DVR CAP NUM SCRL

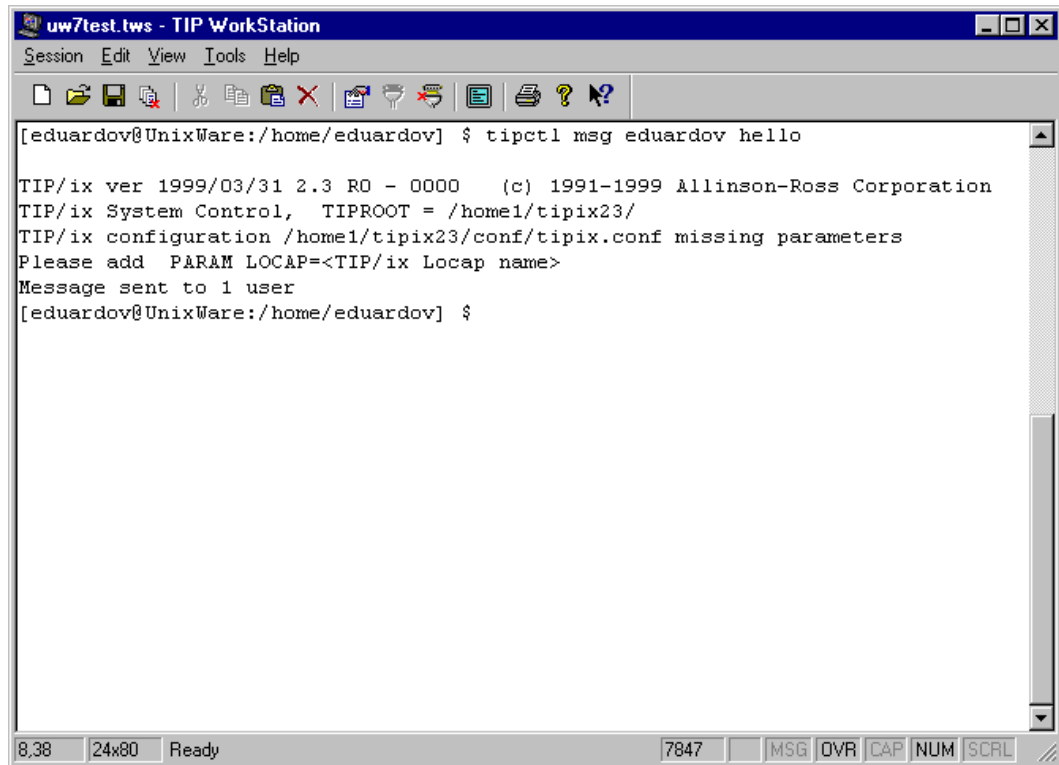
```

The **tipctl** utility requests confirmation before killing all users sessions.

## msg - Send a Message

To send a message to a user, enter the following command:

```
tipctl msg [user] message
```

**Example:**

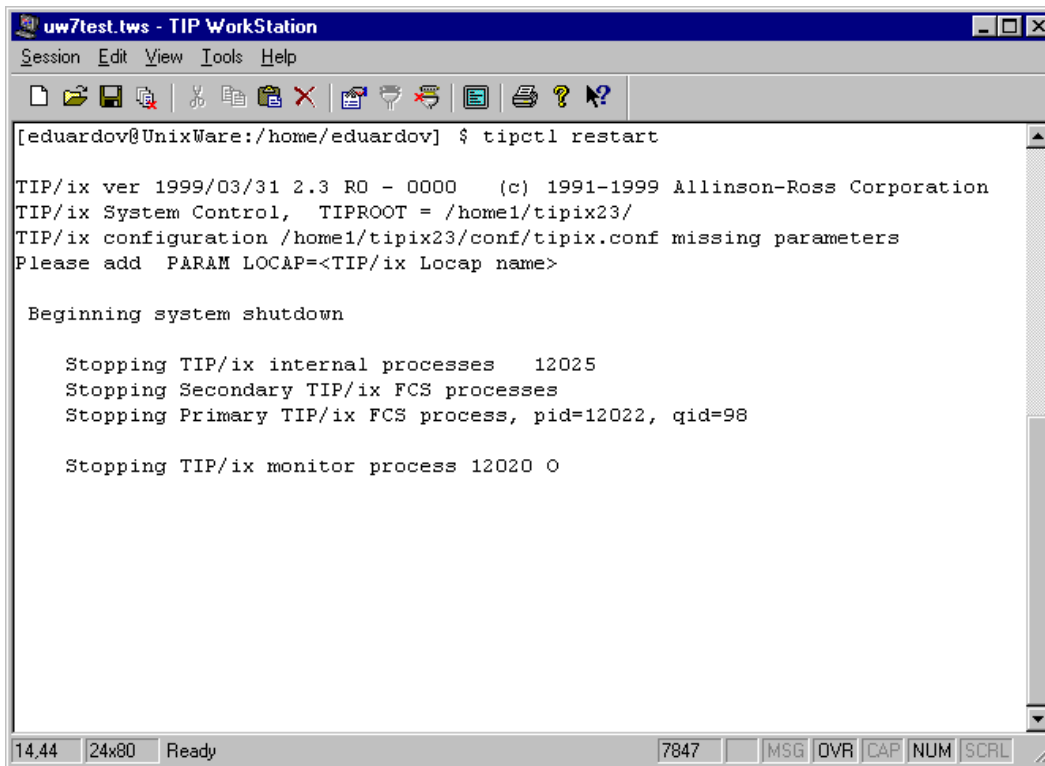
```
uw7test.tws - TIP WorkStation
Session Edit View Tools Help
[eduardov@UnixWare:/home/eduardov] $ tipctl msg eduardov hello

TIP/ix ver 1999/03/31 2.3 RO - 0000 (c) 1991-1999 Allinson-Ross Corporation
TIP/ix System Control, TIPROOT = /home1/tipix23/
TIP/ix configuration /home1/tipix23/conf/tipix.conf missing parameters
Please add PARAM LOCAP=<TIP/ix Locap name>
Message sent to 1 user
[eduardov@UnixWare:/home/eduardov] $
```

**restart - Shutdown then Boot**

Shutdown then do a **tipctl boot**. This option will shutdown the system and then bring it back up.

```
tipctl restart
tipctl r
```



```
uw7test.tws - TIP WorkStation
Session Edit View Tools Help
[eduardov@UnixWare:/home/eduardov] $ tipctl restart

TIP/ix ver 1999/03/31 2.3 RO - 0000 (c) 1991-1999 Allinson-Ross Corporation
TIP/ix System Control, TIPROOT = /home1/tipix23/
TIP/ix configuration /home1/tipix23/conf/tipix.conf missing parameters
Please add PARAM LOCAP=<TIP/ix Locap name>

Beginning system shutdown

Stopping TIP/ix internal processes 12025
Stopping Secondary TIP/ix FCS processes
Stopping Primary TIP/ix FCS process, pid=12022, qid=98

Stopping TIP/ix monitor process 12020 O

14.44 24x80 Ready 7847 MSG OVR CAP NUM SCRL
```

## s - shutdown TIP/ix

To shut down the TIP/ix system, enter the following command:

```
tipctl s
```

If any users are logged on, tipctl prompts you for confirmation.



```

uw7test.tws - TIP WorkStation
Session Edit View Tools Help
[eduardov@UnixWare:/home/eduardov] $ tipctl s
TIP/ix ver 1999/03/31 2.3 RO - 0000 (c) 1991-1999 Allinson-Ross Corporation
TIP/ix System Control, TIPROOT = /home1/tipix23/
TIP/ix configuration /home1/tipix23/conf/tipix.conf missing parameters
Please add PARAM LOCAP=<TIP/ix Locap name>
 1 Active user on 1999/06/01 at 19:37 for TIPROOT=/home1/tipix23/

User-Id Term Program Lvl #In #Out Pid Shell Seg Qid Resp
===== ===== ===== == == == == == == == ==
EDUARDOV 17935 TIP/ix 0 0 1 0 12073 19 2192 0.00
Listed messages: Input 0 Output 1
TIP/ix has 1 active user
Ok to kill them and shut system down (Y/N)?>y

 Stopping TIP/ix shell processes 12073 o

```

## stop - Shutdown Immediately

To shutdown the TIP/ix system immediately, without waiting for users to log off, enter the following command:

```
tipctl stop
```

The system SHUTDOWN script (if it exists in the TIPROOT directory) is *not* executed.

Under normal conditions, **tipctl EOJ** is the best way to shut down the system.

A **tipctl stop** command may be necessary to force off users that are running programs that do not recognize that system shutdown was requested. See the description of the PIB-SYSTEM field in the documentation of the PIB (Process Information Block).

## w - Recover then Shutdown

To perform recovery for TIP/ix system then shut it down, enter the following command:

```
tipctl w
```

## tipdump - Report Shared TIP/ix Memory

Create a report on shared TIP/ix memory in the file tipix.dump in the current directory. Tipdump can be executed from UNIX as well as TIP/ix.

### Syntax:

```
tipdump [-g|-h|-t|-f tipfile] [-o outfile]
```

### Where:

- f Prints fcs status report for the tip\_file only
- g Prints global memory information in terms of offsets etc.
- h Dumps the TIP/ix shared memory in hex format in addition to default information (mentioned below).  
-h option is turned off by default.
- t Prints transaction and key hold information only.
- o outfile  
When specified, tipdump writes to outfile instead of tipix.dump.

## tipflg - Flag Manipulation

The TIP/ix system has 32 flags (or semaphores) that are accessible by all TIP/ix online programs. These flags may be used by online programs for whatever purpose; typical uses are as "semaphores" to queue or control access to resources.

The **tipflg** utility interrogates or changes the setting of any of the flags. The flags may also be manipulated by an online native mode program (see "TIPFLAG" in the *TIP/ix Programming Reference*.)

Before using this transaction in a cavalier fashion, the user is advised to check with the installation administrator. Some of the 32 flags may be used for specific scheduling purposes and should not be modified without careful consideration.

### Syntax 1 - Online:

```
tipflg [command] [, f1, f2, f3, f4, f5, f6, f7]
```

### Syntax 2 - UNIX:

```
tipflg [command] [f1 f2 f3 f4 f5 f6 f7]
```

**Where:****command**

The tipflg commands appear in the following list. If no command is specified, the FLAGS command (shown below) is taken as the default.

**WANYS**

Wait for specified flags to be on.

**WALLS**

Wait for all to be on.

**WSETC**

Wait for specified flags to be on then set them off.

**WANYC**

Wait for specified flags to be off.

**WCLRS**

Wait for specified flags to be off then set them on.

**SET**

Set specified flags on.

**CLEAR**

Set specified flags off.

**FLAGS**

Display current status of all flag bits.

This is the default.

? Display help.

h Display help.

**f1...f7** Optional parameters where the user may specify up to 7 flags that are to be acted upon by the specified command.

Flags are numbered 0 through 31.

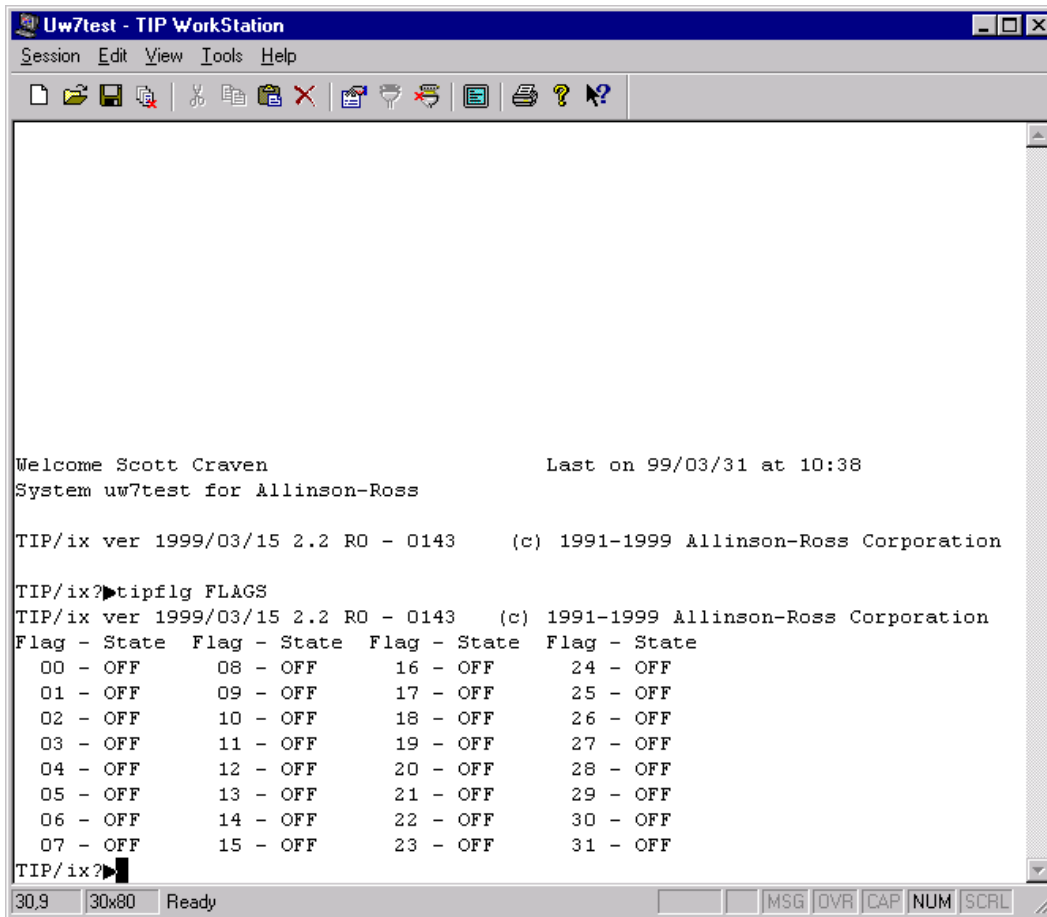
All 32 flags may be specified by using an asterisk, for example:

```
tipflg set *)
```

**Examples:**

```
tipflg FLAGS : display current flags status
tipflg CLEAR 0,1,2 : turn off flags 0, 1, 2
tipflg WANYS * : wait for any bit to be set
```

### Example output of tipflg



```

Uw7test - TIP WorkStation
Session Edit View Tools Help

Welcome Scott Craven Last on 99/03/31 at 10:38
System uw7test for Allinson-Ross

TIP/ix ver 1999/03/15 2.2 RO - 0143 (c) 1991-1999 Allinson-Ross Corporation

TIP/ix? tipflg FLAGS
TIP/ix ver 1999/03/15 2.2 RO - 0143 (c) 1991-1999 Allinson-Ross Corporation
Flag - State Flag - State Flag - State Flag - State
 00 - OFF 08 - OFF 16 - OFF 24 - OFF
 01 - OFF 09 - OFF 17 - OFF 25 - OFF
 02 - OFF 10 - OFF 18 - OFF 26 - OFF
 03 - OFF 11 - OFF 19 - OFF 27 - OFF
 04 - OFF 12 - OFF 20 - OFF 28 - OFF
 05 - OFF 13 - OFF 21 - OFF 29 - OFF
 06 - OFF 14 - OFF 22 - OFF 30 - OFF
 07 - OFF 15 - OFF 23 - OFF 31 - OFF

TIP/ix?
30.9 30x80 Ready MSG OVR CAP NUM SCRL

```

#### Additional Considerations:

The tipflg program is *not* an interactive program. Parameters must be entered on the command line.

## tipix - Command Shell

The **tipix** program is the TIP/ix "shell". You may execute all TIP/ix transaction programs in the shell by entering the transaction name and pressing **XMIT**.

From the TIP/ix shell you may be able to execute UNIX programs as well as TIP/ix transaction programs. Permission to run UNIX programs from the TIP/ix shell is controlled via the TIP/ix user id definition (see [smuser](#)). If you have been granted permission to run UNIX programs then you can invoke Unix utilities (like **vi** or **make**). However, this does not include the ability to use alias names for programs that you may have established in your UNIX shell.

When you execute the **tipix** command, a TIP/ix session is established. If no user id is supplied with the **tipix** command (see **-u** option below) then your UNIX user id is used to establish the session. TIP/ix will convert your UNIX user id to upper case and look up the TIP/ix definition for the resulting user id (see [smuser](#)). If the user id is not defined to TIP/ix then the definition for the user id "DEFAULT" is used. If neither your user id or the user id "DEFAULT" is defined then you are denied access to TIP/ix.

If you wish to restrict access to the TIP/ix system to the users defined to TIP/ix then the TIP/ix user definition for "DEFAULT" must be removed.

The TIP/ix user id that is used to establish the TIP/ix session determines the access permissions granted to the user for the duration of the session. Security level and group membership (elective groups, Group set, and Logon set) are the attributes of the user id definition that determine a user's ability to access files, programs, and queues.

When starting **tipix** without specifying a user id you are *not* prompted for a password (because you already established your identity by specifying a password when you logged onto UNIX).

However, if you supply a user id (**tipix -u user id**) when attempting to start a TIP/ix session then you are prompted for the password for that TIP/ix user id (if a password was defined for that user id). It is a good idea to password protect all TIP/ix user ids to prevent unauthorized access. Passwords can be set for user ids with the [smuser](#) program.

#### Syntax:

```
tipix [options]
```

#### Where:

- a If logging is requested (options **d**, **D**, or **l**) then log all available debugging information. The default is to write minimal debug information to the log file(s).  
If logging is not requested then option does nothing.
- d Create debug log files for *every* transaction. If you execute the program **PAYUPD**, a file called **log.PAYUPD** is created in the HOME directory of the user executing **tipix**. The file will contain a log of all CALLs made to TIP/ix by that program.

Each invocation of a transaction (at this TIP/ix session) creates a new log file overlaying the previous log file for the transaction. Therefore, this option may **not** be suitable for debugging IMS programs that call (succeed to) themselves. Instead use the option "**-D logname**".

If the field *Log file level* is set in a program definition (see [smprog](#)) then the program log setting takes precedence over this option. Therefore, if logging is requested in a

program definition then this option is not required. As well if logging is set to *Never* in the program definition then no log information will be created for the program even if this option is specified.

**-D name**

Create a debug log file with the specified name. If the file name supplied does not begin with a period or a slash then it is created in the HOME directory of the user executing tipix. Similar to the -d option, except all debug information is written to a single file.

This is the recommended logging style as it shows the sequence in which transactions were called. This is especially useful in when several programs are invoked in the session or if a single program is invoked multiple times (TIPSUBs or IMS succession).

If the field *Log file level* is set in a program definition (see [smprog](#)) then the program log setting takes precedence over this option. Therefore, if logging is requested in a program definition then this option is not required. As well if logging is set to *Never* in the program definition then no log information will be created for the program even if this option is specified.

**-h[elp]**

Display the available options.

If this option is supplied then all other options are ignored.

**-l**

Create a debug log file for the **tipix** shell itself. The log information will be written to the file "log.tipix" in the HOME directory of the user executing tipix. This log is not normally required for debugging TIP/ix applications.

**-m tran**

The transaction **tran** is executed immediately after entering TIP/ix. When the transaction terminates you automatically exit TIP/ix. This is useful for invoking user written menu systems.

If you must supply command line parameters to the transaction, enclose the transaction name and the parameters in quotes.

Example: tipix -m "status s"

**-M**

***For problem determination by IngleNet customer support only.***

Turn on an extra level of logging to aid IngleNet in diagnosing problems related to the interaction of TIP/fe

with the TIP/ix user interface including: AUX1 printing, computer transfer, MCS, etc. Used as follows:

```
tipix -laM
```

**-s name**

Execute the script file *name* from the TIP/ix shell. The file is expected to contain commands that can to run from the TIP/ix shell. When all of the commands have been executed then the TIP/ix session is terminated.

If the script file can not be opened then an error message is displayed and the TIP/ix session continues as if the “-s” option had not been supplied.

TIP/ix does not require UNIX execute permission to the file. All that is required is read permission.

**-t tname**

Redirect transaction program output to another terminal. This frees up the original terminal for use by source level debuggers.

Before using this command line option, you should have entered the command "sleep 20000" on the terminal named ("*tname*" is the terminal name of the other terminal where you have also logged onto the same Unix system.)

**-T** Force this TIP/ix session to operate in test mode.

The TIP/ix prompt will indicate that it is running in test mode.

Test mode prevents the user from updating data files that are defined with the “Record hold” field set to “T” which means hold for transaction (see [smfile](#)).

If test mode is active then whenever a transaction end (commit point) is reached TIP/ix will roll back any updates for files defined as hold for transaction. It is as if all TIP/ix transactions set the PIB-LOCK-INDICATOR to PIB-ROLLBACK at transaction end.

This option lets you test the flow of control in programs, and train new users, *without updating files that have “Record Hold” set to “T”*.

Be careful — test mode does not prevent updates (adds, and deletes) to files that are defined with “Record hold” set to “U” (update) or “Y” (yes).

**-u user id**

Login to TIP/ix as the user specified. If that TIP/ix user id requires a password, you will be prompted to supply the

password. This option can be useful to login as another user to test their security capabilities or try to recreate a problem they are reporting.

#### **-x tran**

The transaction *tran* is executed as an initial command upon entering TIP/ix. You can also enter a complete command line as long as it is enclosed by quotes ("command line").

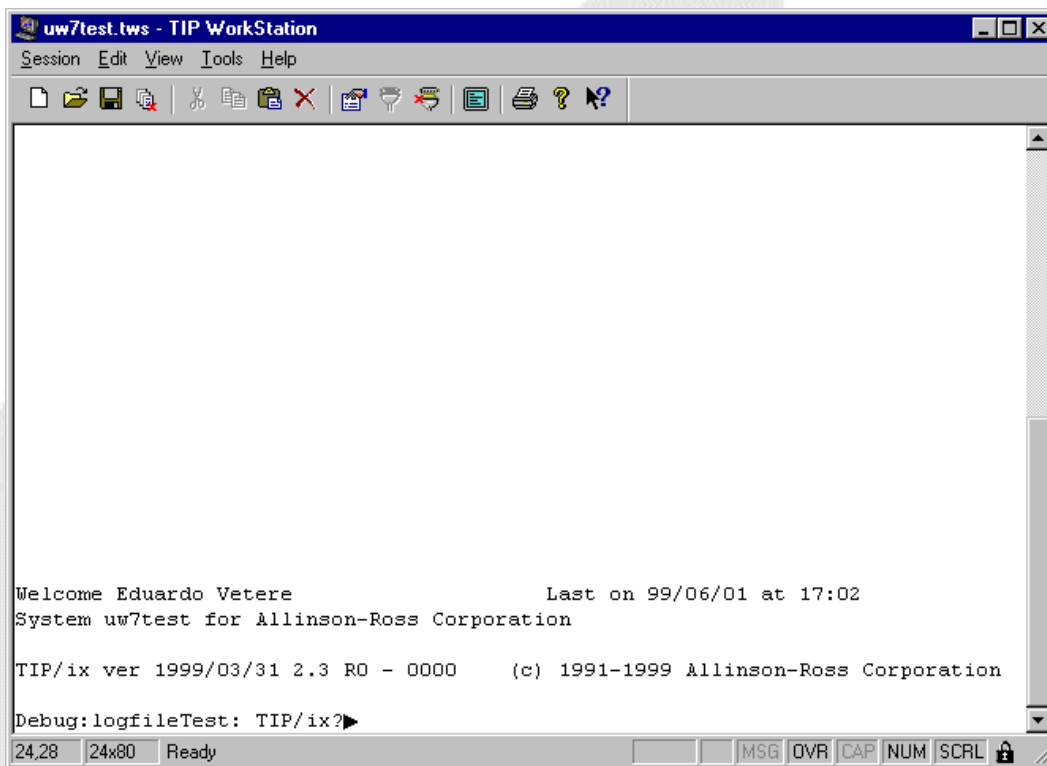
After the initial transaction terminates the TIP/ix session continues (contrast with **-m** option above).

#### **Example:**

```
tipix -laTD logfile
```

It is possible to combine most options as shown by this example. In this example a TIP/ix session will operate in test mode with maximum logging requested for both the TIP/ix shell and transaction programs. The transaction log will be written to the file *logfile* in the users home directory.

A display similar to the following is displayed after entering the command in this example. The prompt indicates the modes that logging is on ("Debug"), that the transaction log will be written to the file *logfile*, and that the user is running in test mode ("Test").



To exit the TIP/ix shell program enter the command **fin**.



## tipix - Command Line Syntax

The TIP/ix command line provided to transactions has this form:

```
[.]trid[,options] [parameter1]...[parameter8]
```

Where:

**[.]** If specified, start the transaction as a background program. This is only valid for programs that do not solicit input from a terminal.

**trid** The transaction id is the name of the program (transaction) to run.

### options

The options are separated from the trid with a comma. Options are from one to eight characters which are defined by the particular transaction program.

### parameters

You can specify up to eight parameters. At least one space must separate the first parameter from the *trid* or *options*. Commas may be used as placeholders for omitted parameters.

For example, run the "payroll" transaction in the background with the "x" option. The second parameter is omitted.

```
.payroll,x mar,,1999
```

## tipix - Shell Commands

The following are some special TIP/ix transactions (commands) which may operate differently than the UNIX commands with the same name:

**cd** Change directory. If no arguments are supplied, it displays the current working directory.

**clear** Clear the terminal screen and reissue the TIP/ix prompt.

**fin** Terminate the TIP/ix shell.

**free** Remove any files in the Active File Table *and* clear the MCS (screen format) cache.

### setenv

Set environment variable. This command sets environment variables with syntax similar to that of the C shell command of the same spelling. Note that environment variable changes from within TIP/ix do not remain in effect when the TIP/ix shell terminates.

Example: setenv A "hello world"

### unsetenv

Remove a variable from the TIP/ix environment. Note that environment variable changes from within TIP/ix do not remain in effect when the TIP/ix shell terminates.

Example: unsetenv A

## tipix - Command Recall

The TIP/ix command shell (**tipix**) keeps a command history of the last ten (**10**) unique (and valid) commands for each TIP/ix session. When a session terminates the command history for the session is discarded.

The command history can be accessed using the following commands of the TIP/ix shell.

| Command | Description                                                                                                                     |
|---------|---------------------------------------------------------------------------------------------------------------------------------|
| ;C      | Clears the command history for the TIP/ix session.                                                                              |
| ;R      | Recalls (displays) the entire command history (maximum of 10 commands).                                                         |
| ;Rn     | Recalls (displays) the nth command from the command history. The valid range for n is 1 to 10. 1 refers to most recent command. |
| ;Rp     | Recalls (displays) the most recent command with a prefix of p, where p can be from 1 to 7 characters.                           |
| ;X      | Executes the most recent command.                                                                                               |
| ;Xn     | Executes the nth command from the command history. The valid range for n is 1 to 10. 1 refers to most recent command.           |
| ;Xp     | Executes the most recent command with a prefix of p, where p can be from 1 to 7 characters.                                     |

---

## tippack - export TIP/ix files

The **tippack** utility is a Unix program that enables you to export TIP/ix files. (You import them later on another system with [tipupack](#).)

You can export:

- TIP\$SYS system records (containing definitions of files, programs, users, terminals, groups, etc),
- files,

- and object files.
- to media including:
  - disks,
  - diskettes,
  - 9-track tape,
  - QIC,
  - and DAT.

Package files are in **tar** format. TIP/ix system files are located under \$TIPROOT/tipfiles.

**Syntax:**

```
tippack [-hnv] -f packlist [-o "taroptions"] package
tippack [-hntv] [-o "taroptions"] package files
```

**Where:**

- h** Output help message about usage.
- n** No prompting. Run tippack without prompt even if a record or file is not found.
- t** List the table of contents of the package file to STDOUT. Not valid with the parameter *packlist*.
- v** Generate a list of the records and files that are being exported to the package. The list is printed on STDOUT. The format is the same as with option -t.
- f** Obtain packing list from a file. If used, the parameter *packlist* must be specified.

If not specified, **tippack** reads the packing list from standard input.

**packlist**

Specifies a packing list (text file). Used with option -f. Not valid with option -t or with parameter *files*. This file contains:

Entries for records and files to export;

Optional definitions of the location where records and files will be imported to.

**-o "taroptions"**

Pass *taroptions* to the underlying **tar** command to generate the package. If -o is *not* specified, "cf" is passed to the **tar** command.

If the options contain an embedded space, use quotes.

**package**

Specifies the file that **tippack** will generate for export.

Package file can be either disk file or special file. If a special file (/dev/rmt/c0s1, for example) is specified, the output will be written to a device such as diskette or tape.

**files** Specifies a list of files to be exported. It is optional and not valid with the option -f. This parameter has the same syntax as that in a packlist. Entries are separated by colons.

## Packing List

The packing list is a text file that specifies what to export, and optionally, where to put it (where to import it to).

In general, items from UNIX, such as filenames and options, are case-sensitive, and items from TIP/ix are case-insensitive.

The entries for the packing list are defined in the following sections.

## TIP/ix System File Records: SET

You can use the system maintenance transactions ([smfile](#), [smprog](#), etc) to group related TIP\$SYS system records (containing definitions of files, programs, users, terminals, groups, etc) into sets. The purpose of these sets is to make exporting these system records more convenient.

Specify which sets (of TIP\$SYS system records) to export.

```
SET name [*] [... name-n [*]]
```

Export and specify a new set name for later import.

```
SET name [TO to-name]
```

### Where:

#### name

Specify which sets to export. Separate set names with spaces. Wildcards are supported (unless TO is specified). For example, foo\*.

Set names are case-insensitive. For details about sets, see [smfile](#).

**TO** Keyword. Can start on a new line.

#### to-name

Specify a new set name for import. If used, only one set can be specified by the name parameter. It is not valid if wild cards are used for name-n already. New set name is case-insensitive.

**Examples:**

Export sets AP and PAY.

```
set ap pay
```

Export set OLDSMP. When it is imported to the destination, set OLDSMP will have the new name SMP.

```
set oldsmpt to smpt
```

Export all sets with prefix PMA or DDS.

```
set pma* dds*
```

## TIP/ix System File Records: RECORD Statement

Specify which TIP\$SYS system *records* (containing definitions of files, programs, users, terminals, groups, etc) to export.

```
RECORD type1 key1[*] [... key1n[*]]
 [type2 key2[*] [... key2n[*]]] ...
```

Export, and specify a new record name for later import.

```
RECORD type1 key1 [TO to-key]
```

**Where:**

**type** Specify the type of record to export. The types are one or more of the following keywords:

**file** Specifies data file record.

**group** Specifies group set record.

**LOCAP** Specifies LOCAP record.

**mcs** Specifies MCS screen format.

**msg** Specifies a canned message.

**print** Specifies printer record.

**prog** Specifies program record.

**queue** Specifies queue record.

**sec** Specifies security record.

**user** Specifies user record.

**term** Specifies terminal record.

Multiple types can be specified for the same key.

For example:

```
RECORD prog yearend file yearend
```

**key[\*]** Specify which records to export. Most keys have only one part (no slashes). However, the keys for security record and MCS screen format have several parts.

**security record key***group/recordtype/item*

Record type can be F, L, P, or Q (or in lowercase).

See [smsec](#) for more information about group, record type, and item.**MCS key***[group/]name***msg key***language/product/IDnumber*

Parts in a key are separated by a comma or a slash "/".

Multiple keys can be specified and are separated by space characters. Wild cards are supported. For example, ipa\*.

Key fields are case-insensitive. Parameters *types* and *keys* must be used in pairs and are separated by space characters. Any type can only be specified once per clause.**TO** Keyword. Can start a new line.

to-key Specify a new name to be used for the key when it is imported. If used, only one key can be specified by the key parameter. Wild cards not allowed. The new name is case-insensitive. The type of key are assumed as the same as that defined in the type-key pair.

If you need to specify a keyword (such as *prog*) as a key, specify only one type-key pair at a time.

For example:

```
RECORD prog prog
```

**Examples:**

Export program records for ACTOPA, OPAABC, and ACTOPB.

```
record prog actopa opaabc actopb
```

Export security record TIP\$SYS/F/DOCREC and DEV/F/PMREC+.

```
record sec ip$sys,f/docrec dev,f/pmrec
```

Export file records with prefix SMP or MR, and user records JOHN and DAVE, together with their security records.

```
record file sec smp* mr* user sec JOHN DAVE
```

Export printer record OLDHPJET. When imported, OLDHPJET is renamed to record HPJET.

```
record print oldhpjet to hpjet
```

## Files: FILE Statement

Specify which *files* to export. (This exports the actual files – not just the file definition records in TIP\$SYS.)

```
FILE type1 key1[*] [... key1n[*]]
 [type2 key2[*] [... key2n[*]]]
 [path3[*] path4[*] ...]
 [SET name5[*] name6[*] ...]
```

Export one record, and specify a new type and key for later import.

```
FILE type1 key1 TO to-type to-key
FILE path1 TO to-type to-key
```

Export one record, and specify a new path for later import.

```
FILE type-1 key-1 TO to-path
FILE path1 TO to-path
```

### Where:

- type** Specify the record type. Only three types are valid: *file*, *prog*, or *sec*. Only one type can be specified for each type-keys clause. The rules described in RECORD for system records also apply.
- keyn** Specify record names. If the key is for security record, the type can only be either F or P. All other rules described in RECORD for system records apply. For each type-keys clause, **tippack** will obtain information about the UNIX paths of the files from TIP/ix system records. Type-keys pairs are separated by space characters.
- path** Specify UNIX paths for the files to export. If specified, TIP/ix system files will not be accessed for information about the path.

For executables, **tippack** will search the file according to the path specified or using the environment variable PATH. Wild cards can be used for pattern matching. Can be mixed with the type-keys clauses. If the path is specified starting with \$TIPROOT, **tippack** will recognize it and [tipupack](#) will export the file with the path relative to \$TIPROOT.

**tippack** also recognizes other environment variables used in the path. However, it will substitute the environment variables with their values.

**Note:** Mixing parameter path-n with type-keys pairs may cause confusion. If this is the case, put all UNIX paths after the keyword *file*, before any type-keys pair or use a separate line for UNIX paths.

**SET** Keyword to specify sets.

**name-n**

Specify set names. Every file defined in the sets specified will be exported. If an item in a set cannot provide a UNIX path for files, the item is ignored. Set names are separated by space characters. Wild cards are supported. For example, foo\*. Set names are case insensitive. **tippack** will obtain information about UNIX paths of the files to export.

**TO** Keyword. Can start a new line.

**to-type**

Specify new type/key used for the correspondent key or path when it is imported. Only three types are valid: *file*, *prog*, or *sec*. Only one type-key pair can be specified. Not valid if the parameter to-path has been specified. Other rules described in RECORD for system records apply.

**to-key**

Specify new type/key when it is imported. Only one type-key can be specified. Not valid if the parameter to-path has been specified. It is also not valid if wild cards are used for key-n or path-n already or if sets are specified. The new key is case insensitive. Information about the UNIX path of the file will be obtained by [tipupack](#) from TIP/ix system records when imported.

**Note:** Since the UNIX path is obtained from the host system only when the file is imported, the UNIX path is not well defined in the package file. If not sure, the user should not specify the to-type-key clause. Instead, when importing the user can modify the packing list on the site where he can know more about the UNIX path defined in records.

**to-path**

Specify new path to be used when it is imported. Only one to-path can be specified. Not valid if the to-type/to-key has been specified. It is also not valid if wild cards are used for key-n or path-n already or if sets are specified.

Since **tippack** will search the executables using environment variable PATH, it may pick up the wrong file if user's path is not set up right. If in doubt, absolute UNIX paths should be used to specify executables for export instead of type-key pair, relative UNIX path, or bare file name (that is, use /u/prod/actpy instead of actpy).

If files are imported using paths that defined by system records, [tipupack](#) searches the system file that is in the destination system. This approach



may cause error since the paths defined in the system file may not match that in the package.

#### Examples:

Export executables defined by TIP/ix system records IMPRAA and IMPRAB for programs. Also exports executable payrec by a UNIX path and files defined by TIP/ix system records ALPFLE and SAMIMS for files.

```
file prog impraa imprab /u/prod/bin/payrec
file alpfle samims
```

Export executable defined by TIP/ix system record IMPRAA. When imported, the executable with IMPRAA will be placed as a UNIX file /u/prod/bin/imprbb.

```
file prog impraa to /u/prod/bin/imprbb
```

## Files: OBJECT Statement

This is a simplified version of the FILE statement. If no TIP/ix system record is involved, then the keyword file can be ignored. This is useful if the files are in the current directory and have names the same as the keywords, for example prog and user.

```
path-1 [path-2 ...][to to-path]
```

## Object Files

Export object modules. This is a special case of exporting files.

```
OBJECT lib1 object1[*] [... object1n[*]]
 [lib2 object2[*] [... object2n[*]]]
```

Export one object record, and specify a new object for later import.

```
OBJECT lib1 object1 TO to-lib
```

#### Where:

**libn** Specify the UNIX paths of the archive file where the object files are in.

**objectn** Specify object file names. Multiple object files can be specified for each lib-n and are separated by space characters. Object files also can be specified by wild card.

**TO** Keyword. Can start on a new line.

**to-lib** Specify new archive file when it is imported. Only one lib-objects pair can be used.

Export object files tipcw.o, gosstr.o, and packbuf.o from archive /u/tipix/lib/libbat.a.

```
object /u/tipix/lib/libbat.a tipcw.o
 gosstr.o packbuf.o
```

Export object files regst.o from archive /u/dev/lib/lib1.a. In addition, object file regst.o will be updated to the archive /u/prod/lib/libonline.a when imported.

```
object /u/dev/lib/lib1.a regst.o to
 /u/prod/lib/libonline.a
```

## Comment

Anything after a pound sign is interpreted as comment. If a pound sign is part of the file/record name, it must be escaped by the sequence '\#'.

```
comment text
```

## Echo

Output text to a terminal.

```
ECHO message text
```

## Example Packing List

```
packing list for demo
#
set opa smp mrf* # export sets
set oar to oar1
Export set. Use new set name on import.
record prog sec mad*
export prog and sec records with prefix mad
record prog mrfad smp10g file hdrfile docrec samims
export records
record prog smp11g to smp11ga
use new record name on import.
export files
file prog oar25t mrfmd20
/u/dev/bin/smpalg /u/dev/bin/pma25t
use new location and record on import
file prog oar26r to /u/dave/mybin/oar26r
file set agt pkey*
export all files in set AGT and sets with
prefix PKEY
export object files
object /u/tipix/lib/libbat.a tipcw.o gosstr.o packbuf.o
```

## tippager - Page Browsing Transaction

The **tippager** transaction enables users to browse through an application's paging file. Typically, a user transaction saves data in the paging file then invokes **tippager** to view the data.

For details about creating a paging file, see TIPPAGE in the *TIP/ix Programming Reference*.

### Syntax:

From a user transaction:

```

 MOVE "TIPPAGER" TO PIB-TRID
 CALL "TIPXCTL"

```

If a paging file has been created by another transaction, you can invoke tippager from the TIP/ix command line:

```
tippager
```

There are no parameters.

### Error Conditions:

If **tippager** cannot open the paging file it displays an error message.

## Commands

The user interface is provided through function keys:

| Key | Function      | Description                                                                                                                                                |
|-----|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|
| F1  | HOME          | Display the first page of the paging file.                                                                                                                 |
| F2  | NEXT          | Display the next page. If you are viewing the last page, F1 takes you to the first page of the file.                                                       |
| F3  | PREVIOUS      | Display the previous page. If you are viewing the first page, pressing F2 takes you to the last page of the file.                                          |
| F4  | HOME<br>INDEX | Since index pages are not currently supported, F4 behaves exactly like F3.                                                                                 |
| F5  | STATUS        | Display the status of the paging file (including the total number of pages stored, the location of the current page, and maximum number of pages allowed). |
| F6  | EXIT          | Terminate the pager process (without releasing the pages).                                                                                                 |

|    |        |                                                                                                                     |
|----|--------|---------------------------------------------------------------------------------------------------------------------|
| F7 | DELETE | Delete all pages from the file. (Exit the paging transaction and release all the information from the paging file.) |
| F8 | HELP   | List help on the available commands.                                                                                |
| F9 | PRINT  | Print page(s) of the paging file. <code>tippager</code> displays the following dialog box:                          |

Error! Not a valid filename.

By default, the *current page* is printed on the *default system printer*.

To specify which page(s) to print, put an X in one of the PAGE RANGE fields, then press XMIT.

---

## tippcstm - Thread Manager

The thread manager is used to dynamically alter the attributes affecting execution of **reusable** transaction programs. COBOL transaction programs may be defined as reusable in an effort to improve system performance by reducing the overhead associated with preparing a program for execution. Refer to “Serial Reusable Execution Mode” of [smprog](#) for a discussion of desirable attributes for reusable transaction programs.

There is a certain amount of overhead involved in starting programs when a request is made to run a program from TIP/ix. This involves setting up a UNIX process to be used in the execution of program. If a program is designated as reusable then when the program terminates the UNIX process is not destroyed. Instead it is kept available to service the next request to run the program and therefore saves the overhead associated with preparing the UNIX process for execution.

The term *thread* refers to the logical execution of a transaction. The active execution of a thread requires the use of a UNIX *process*. A single UNIX process may be used repeatedly to execute transaction threads serially. If a transaction program is defined as *reusable* (via [smprog](#)) then the transaction runs as a child process of the thread manager. The thread manager manages the pool of processes (*servers*) used to execute the *reusable* program according to the attributes set in the program definition (see [smprog](#)).

The **tippcstm** utility allows a TIP/ix system administrator to manage the execution of reusable programs and even dynamically alter some of the reusable execution attributes that were set in the program definition.

### Syntax:

```
tippcstm [options]
```

**Where:**

**b[oot]** Restart the thread manager. This is the default if no option is specified at the command line.

**s[hutdown] pcs**

Shut down the entire thread management system.

WARNING: This action may stall TIP/ix.

**s[hutdown] all**

Shut down all idle server processes.

Will not shut down any server that is currently executing a program.

**s[hutdown] server name**

Shut down all idle server processes used to execute the program definition name (see smprog).

Will not shut down a server if it is currently executing the program name.

**alter name min=n**

Change the minimum number of threads to n for the server for the program definition name.

Keep at least this many processes available to service requests to execute the program definition name. For recommendations, see smprog.

**alter name max=n**

Change the maximum number of threads to n for the server for the program definition name.

If the maximum number of threads are active for a program then further service requests will wait until one of the processes becomes available). A process becomes available when an IMS program calls "RETURN" or a TIP/ix program calls "TIPRTN". For recommendations, see smprog.

**alter name use=n**

Change the maximum number of times that a process can be used to service a request to run the program name before it is reloaded. A reload may be required to refresh the data segment.

**alter name idle=n**

Change the maximum number of seconds that a process servicing requests for the program name can be idle to n.

If the process remains idle for this length of time then it will be shutdown unless the number of server processes for

the program name is less than or equal to the minimum number specified for name.

---

## tipsam - System Activity Monitor

The **tipsam** (**TIP/ix System Activity Monitor**) transaction allows the administrator to take an eagle's eye view of the dynamic TIP/ix system. **tipsam** presents information on the processes, files and terminals on a system.

The information is presented from the most active to the least active, and is updated regularly at a configurable rate.

### Syntax:

```
tipsam [Refresh-Rate]
```

### Where:

#### Refresh-Rate

Specify how often to update the screen (from 1 to 99 seconds).

The default is to refresh every 3 seconds.

**tipsam** starts up with the following screen layout containing three sets of the seven most active items on the system. The items are grouped as transactions, files and terminals.

```

uw7test.tws - TIP WorkStation
Session Edit View Tools Help
TIP/ix System Activity Monitor 99/06/02 15:04
Page 1 of 1
1 2 3 4 5 6 7
Active Transaction Programs Total Trans: 7
Prog: SMPROG
Used: 1
Msgs: 2
Resp: 0.00

Active Data Files Total Files: 23
File: TIP$SYS TIP$MCS TIP$TRM TIP$MSG TIP$SEC TIP$QUE INVEN
I/Os: 91 18 14 10 9 8 1
Updt: 10 0 0 0 0 0 0

Active Terminals Total Terms: 3 Total Msgs: 82
User: JDOE EDUARDOV BJONES
Term: 20636 20630 20633
Msgs: 43 24 15

Memory -M: Total: 148K Free: 121K Most Used: 27K
 -L: Total: 600K Free: 428K

MSGWAIT: Quit F1: Menu F2/F3: Browse Mode Next/Prev
23.75 24x80 Ready 0608 MSG DVR CAP NUM SCRL

```

The information on the above screen is updated at regular intervals. Thus the list of items presented and their order may change dynamically. The screen refresh rate and other options can be selected through a main menu. This main menu can be invoked either by pressing F1 or the programmable Menu Key (**default** is usually set to Ctrl \).

## Main Menu

This menu contains the following list of options.



To select an option, either type the first character of the option (underlined), or use the cursor keys or the mouse.

## Transactions/Files/Terminals

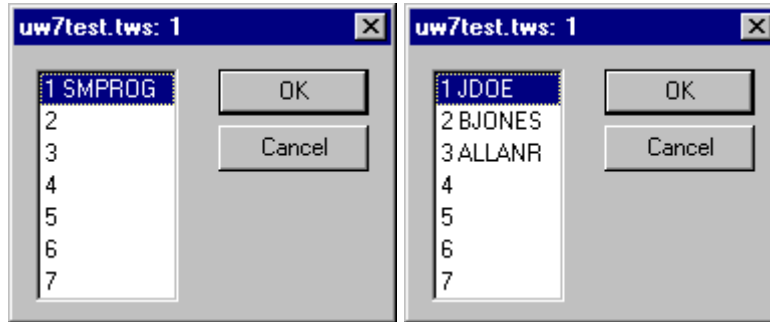
**Transactions**

**Files**

### Terminals

A secondary list of up to seven items is presented as shown below

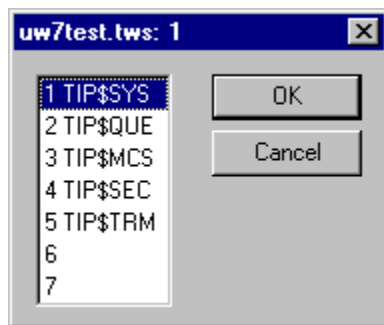
#### Transactions/Terminals Window:



Transaction Window

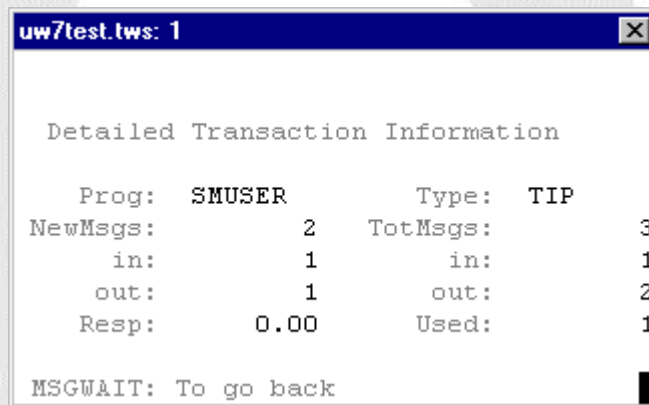
Terminal Window

#### Files Window:



Select one of these items to get a detailed report on that item overlaid on the main screen. The layout depends on the type of the item selected. The screen formats used are as follows.

#### Detailed Transaction Format



#### Where:

**Prog** The name of the transaction or program.



**NewMsgs**

The total number of messages (in and out) generated by the transaction during the last refresh cycle.

**Resp** The average response time of the transaction to process a single input message.

**Type** The type of emulation being used for the transaction, it could be a TIP, IMS or TIP-1100.

**TotMsgs**

The total number of messages (in and out) generated by the transaction since the system start up time.

**Used** The number of times the program (in Prog field) has been used.

**Detailed File Format**

```

uw7test.tws: 1

 Detailed File Information

File: TIP$MSG Type: MIRAM
I/Os: 10 Status: Open
Updates: 0 Server: tipfcs
Pathname:
/home1/tipix23/tipfiles/tipmsg

MSGWAIT: To go back

```

**Where:**

**File** The logical name of the file.

**I/Os** The total I/Os done on the file.

**Updates**

The total writes to the file.

**Pathname**

The full path of the file maximum of 64 characters.

**Type** The type of the file i.e. DYN, DAM, ISAM, LIB etc.

**Status**

Shows whether the file is Open, Close or Off etc.

**Server**

The name of the server currently associated with the file.

### Detailed Terminal Format

```

uw7test.tws: 1
Detailed Terminal Information

User: BJONES Trid: SMFILE
inMsgs: 1 Terminal: 22902
outMsgs: 2
MSGWAIT: To go back

```

#### Where:

**User** The login name of the user.

**InMsgs**  
The total number of input messages.

**OutMsgs**  
The total number of output messages.

**Trid** The name of the program currently running on that terminal.

**Terminal**  
The terminal Id.

### Setup Screen

```

uw7test.tws: 1
Setup Information

Refresh Rate (sec.) : 3

Sort Transactions By:
Msgs: Y Used: Response:

Sort Files By:
IOs: Y Updates:

Sort Terminals By:
Msgs: Y

```

The setup screen enables you to change **tipsam**'s defaults:

#### Where:

**Refresh rate**  
A valid refresh rate is from 1 to 99 sec.

**Sort Transactions**  
Transactions can be prioritized by one of the following:

## Messages I/Os

---

### tipsubp

Tipsubp is not supported in TIP/ix, see the additional considerations of [genmain](#) for a description on how tipsub functions can be compiled in TIP/ix.

---

### tipupack - import TIP/ix files

The tipupack utility reads package files created by the [tippack](#) utility.

The tipupack utility can:

- Import selected items from a package file.
- Import files from media including disks, diskettes, 9-track tape, QIC, and DAT.
- Generate log files.
- Overwrite the locations of files to be imported to.
- Report the contents of package files.

The data is imported into system files in \$TIPROOT/tipfiles (tipsys.dat, tipsec.dat).

#### Syntax:

```
tipupack [-hntvx] [-b backup] [-f packlist]
 [-o tar options]
 [-x packlist] package [files]
```

#### Where:

##### -b backup

Save original items to a file specified by the parameter backup. The file has the same format as package files and can be processed for restoration. tipupack uses command 'tar -cf' to create the backup package.

**-f** Obtain packing list from a file. If used, the parameter *packlist* must be specified. Normally **tipupack** uses the packing list from the package file so this option is not necessary.

**-h** Output help message about usage.

**-n** Run tipupack without prompt even a record or file is not found.

- t** View contents of the package file. Other parameters and options will be ignored if specified.
- v** Print items when importing from package file. This can be redirected to a disk file and thus serves as a log file for the user.

**-x filename**

Extract the packing list from the package file and save it to the file specified. Useful when importing all items but want to modify a few of them.

- o** Passes options to the underlying command tar for generating the package. The default options used for tar command are "xf". If option -o is used the parameter that follows it will be used as options for tar command. If the parameter has embedded space, it should be enveloped by quotes.

**packlist**

Specifies a text file. It is optional and not valid with the parameter *files*. If specified, the parameters and entries in this file will be used to import selected files to the specified locations. This file contains:

Entries for records and files to import;

Optional definitions of the location where records and files are imported to.

**package**

Specifies the file that **tipupack** will process for import. Package file can be either disk file or special file. If a special file (*/dev/rmt/c0s1*, for example) is specified, the input will be from a device such as diskette or tape.

- files** Specifies a list of files to be imported. It is optional and not valid with the parameter *pack-list*. This parameter has the same syntax as that in a pack-list. Entries are separated by semicolon.

See [tippack](#) for details of the packlist.

**Additional Considerations:**

If any *terminal records* are imported with **tipupack**, you should run:

```
smterm re
```

to reset the terminal definitions. This is because tipupack only updates terminal records in the tipsys file and not the tiptrm file. By executing [smterm](#) re, you recreate the terminal records in the tiptrm file for every terminal record in the tipsys file.

## tlib - Librarian Services

TLIB is a utility transaction program that provides online librarian facilities. TLIB manipulates library elements, TIP/ix edit buffers, terminal auxiliary devices (cassette, diskette, printer), and DOS files on Personal Computers (computer).

To manipulate DOS files, TLIB must be executed at a Personal Computer that is logged on to TIP/ix and is using tipfe.

TLIB is able to manipulate source (S) library elements.

For certain commands TLIB recognizes three pseudo library element types:

- D** Directory. Implies the library header information including module name, module type, comments, date and time stamp (similar to a LIBS table of contents listing).
- F** Fast directory. Implies only the module name and module type.
- E** Edit buffer. Indicates that the specification applies to a TIP/ix edit buffer.

TLIB may be used interactively or may be given a single command on the command line. If a single command is given on the command line TLIB will attempt only that command and terminate. When used interactively, TLIB prompts the user for each command.

### Syntax 1 - Interactive:

Start TLIB in interactive mode. TLIB prompts the terminal operator for each command until an ["end"](#) or ["quit"](#) command is given.

```
TLIB[/options]
```

### Syntax 2 -Batch:

Execute a single TLIB command.

```
TLIB[/options] cmd parameters
```

## TLIB Commands

If TLIB detects that it has been called with a transaction name other than "TLIB", it assumes that the transaction code is the command and does not treat the first parameter as a command (the implied command is inserted and the parameters are internally shifted right one position).

TLIB recognizes the following commands:

| TLIB Command | Description |
|--------------|-------------|
|--------------|-------------|

|               |                                                                                          |
|---------------|------------------------------------------------------------------------------------------|
| F1            | In interactive mode, recall last command for cannibalization and possible re-submission. |
| COPY*         | Copy an element or edit buffer to an element, auxiliary device or DOS file.              |
| DELETE/ERASE* | Delete a library element.                                                                |
| DIR*          | Detailed directory of a library.                                                         |
| END           | End TLIB program.                                                                        |
| FDIR*         | Fast directory of a library.                                                             |
| HELP          | Display help information on terminal.                                                    |
| LIST*         | List (on the terminal) an element or edit buffer.                                        |
| PRINT*        | Print a listing of an element or edit buffer.                                            |
| QUIT          | End TLIB program and logoff.                                                             |
| SETOF         | Set TLIB option OFF.                                                                     |
| SETON         | Set TLIB option ON.                                                                      |

BACK, JOB, PUNCH and RECOVER are **not** supported.

In the above table, commands that are suffixed with an asterisk are also implemented as TIP/ix transactions (with the transaction name equal to the TLIB command name). The definition of separate transaction "clone" names permits the system administrator to restrict access to certain commands to users with appropriate security levels and also permits the use of commands (like [COPY](#)) as stand alone transactions.

## TLIB Options

TLIB recognizes the following options (whether supplied on the command line or referenced by a [SETON](#) or [SETOF](#) command in interactive mode). The initial state of these options is "not in effect". Command line options are processed by TLIB from left to right.

| Option | Description                        |
|--------|------------------------------------|
| A      | ASSEMBLER mode - use columns 1-72. |
| C      | COBOL mode - use columns 1-72.     |
| H      | Not supported.                     |
| I      | Not supported.                     |
| K      | Not supported.                     |
| L      | Print with line numbers.           |
| M      | Do not print heading lines.        |

- N Do not print title page.
- O COPY: Omit overwrite prompt.  
DELETE/ERASE: Ignore edit buffer changed flag.
- Q Do not display any messages (Quiet mode).
- R RPG mode  
use columns 1-74;  
columns 1-5 set to spaces.
- S Scratch input edit buffer.
- T Not supported.
- X Not supported.
- Y Not supported.
- Z Removes sequence numbering and update stamps  
when copying elements to a personal computer if the  
language has been set for the element.
- 1...8 Not supported.

#### Additional Considerations:

Some options are mutually exclusive (for example, the language specification options like "R" and "C").

## TLIB Input and Output Specifications

The following tables illustrate the various formats that the parameters to TLIB can take. P1 refers to the first parameter after the TLIB command. P2 is the next parameter and so on.

### TLIB Input Specifications

| Parameters       | Description                                                                                                                                                                                                   |
|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| P1, P2, P3       |                                                                                                                                                                                                               |
| omitted          | If parameters 1 through 3 are omitted, the terminal is used as an input device.<br><br>Input is solicited line by line until MSG WAIT is pressed to signal end of input.                                      |
| File/elt [,type] | Input OS/3 Library element.<br><br>Default is source type "S". Other choices are Macro, Proc, Load module, Object module, Directory, Fast directory, Name of Proc, Internal Symbol Dictionary of Load module. |
| group/name,E     | TIP/ix edit buffer specification. Pseudo type code "E" indicates a TIP/ix Edit Buffer.                                                                                                                        |

d:ffffff.eee    DOS file specification: (8.3 format)  
                   d:     DOS drive identifier  
                   ffffff DOS file name  
                   .eee    DOS file extension

### TLIB Output Specifications

| Parameters                 | Description                                                                                                                                                                                                                                                              |
|----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| P4, P5, P6,<br>P7          |                                                                                                                                                                                                                                                                          |
| omitted                    | If no output specifications are provided, TLIB output is directed to AUX0 (full screen display via TIPPRINT).                                                                                                                                                            |
| File [,element]<br>[,type] | OS/3 library element specification<br>Default element name is input element name (P2)<br>Default type is input type (P3)                                                                                                                                                 |
| Group<br>[,name], E        | TIP/ix edit buffer specification. Edit buffer output length defaults to 85 characters; use parameter seven to override this length.<br><br>Group: the edit buffer group<br>name: the edit buffer name (required if group is given)<br>type (must be "E" for Edit buffer) |
| d:ffffff.eee               | DOS file specification: (8.3 format)<br>d:     DOS drive identifier<br>ffffff DOS file name<br>.eee    DOS file extension                                                                                                                                                |
| AUXn                       | Auxiliary device specification<br>"n" is auxiliary device number (1 through F)                                                                                                                                                                                           |
| printername                | A printer name acceptable to TIPPRINT.<br><br>See description of TIPPRINT in the documentation of the TIP/ix File Control System - FCS.                                                                                                                                  |



## COPY - COPY Data

The TLIB Copy command is able to copy data to and from a wide range of locations. Not only can standard library elements be copied, TIP/ix edit buffers, auxiliary devices, DOS files and the spool file can be used.

### TLIB Inputs

The COPY command recognizes the following input specifications:

- library element
- library directory
- DOS file on a computer
- TIP/ix edit buffer
- the terminal (entered at keyboard).

### TLIB Outputs

The COPY command recognizes the following output destinations:

- library element
- DOS file on a computer
- terminal auxiliary device (AUX0 and AUX1)
- TIP/ix edit buffer
- TIPPRINT printer destination

### Syntax:

The combinations of input and output specifications make the syntax of the COPY command rather cumbersome to illustrate. However, a general rule is: the first 3 parameters pertain to the input specification, parameters 4 through 7 pertain to the output specification.

```
Copy P1 [,P2] [,P3] ,P4 [,P5] [,P6] [,P7]
```

### COPY Input Spec

Specify the input specification for COPY (parameters P1, P2 and P3) from the following table:

| Parameters       | Description                                                                          |
|------------------|--------------------------------------------------------------------------------------|
| P1, P2, P3       |                                                                                      |
| omitted          | If parameters 1 through 3 are omitted, the terminal is used as an input device.      |
|                  | Input is solicited line by line until MSG WAIT is pressed to signal end of input.    |
| File/elt [,type] | Input library element. Default is source type "S".                                   |
| group/name,E     | TIP/ix edit buffer specification. Pseudo type code "E" indicates TIP/ix Edit Buffer. |
| d:ffffff.eee     | DOS file specification: (8.3 format)                                                 |

d: DOS drive identifier  
 ffffffff DOS file name  
 .eee DOS file extension

### COPY Output Spec

Specify the output specification for COPY (parameters P4, P5, P6 and P7) from the following table:

| Parameters                    | Description                                                                                                                                                                                                                                                                                                                                             |
|-------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| P4, P5, P6, P7                |                                                                                                                                                                                                                                                                                                                                                         |
| omitted                       | If no output specifications are provided, TLIB output is directed to AUX0 (full screen display via TIPPRINT).                                                                                                                                                                                                                                           |
| File<br>[,element]<br>[,type] | Library element specification<br>Default element name is input element name (P2).<br>Default type is input type (P3)                                                                                                                                                                                                                                    |
| Group<br>[,name], E           | TIP/ix edit buffer specification<br>Edit buffer output length defaults to 85 characters; use parameter seven to override this length.<br>Group: the edit buffer group<br>name: the edit buffer name (required if group is given)<br>Pseudo type code must be "E" for Edit buffer.                                                                       |
| d:fffffff.eee                 | DOS file specification: (8.3 format)<br>d: DOS drive identifier<br>fffffff DOS file name<br>.eee DOS file extension<br>By default, the DOS file is copied into the current working directory (as defined by the UTS control page). To change the target directory, press Ctrl-F c (or Alt-3) then specify the target directory in the UTS control page. |
| AUXn                          | Auxiliary device specification.<br>"n" is auxiliary device number (0 and 1)                                                                                                                                                                                                                                                                             |
| printername                   | A printer name acceptable to TIPPRINT.                                                                                                                                                                                                                                                                                                                  |

See description of TIPPRINT in the TIP/ix Programming Reference.

### Example of COPY commands

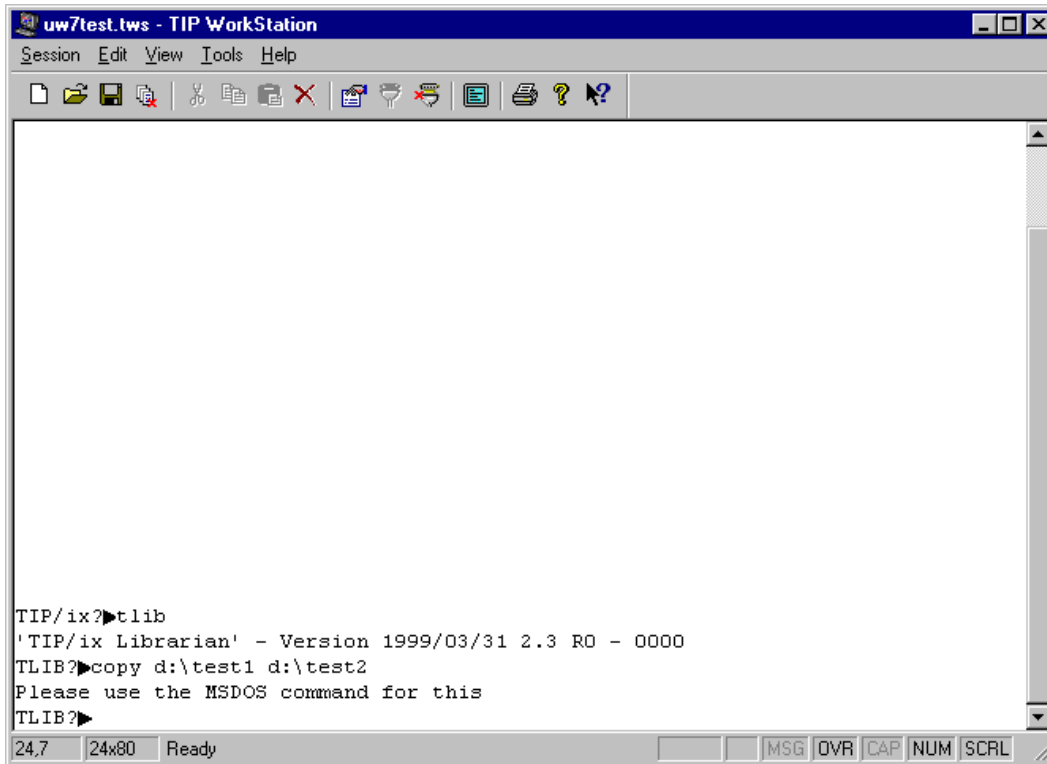
- COPY JCS/TIP30 , , SYSGEN**  
Copies the element TIP30 (default type "S") from library JCS to library SYSGEN as element name TIP30.
- COPY JCS/TIP30 , , SYSGEN/OLDTIP**  
Copies the element TIP30 (default type "S") from library JCS to library SYSGEN as element name OLDTIP.
- COPY PG11 , , E, AUX1**  
Copies the edit buffer PG11 to the terminal AUX1 device (presumably a printer).
- COPY JCS/TIP30 , , B: /TEST/PRN**  
Copies library element TIP30 from library JCS to DOS file B:TEST.PRN
- COPY TEST , , E, RDR, jobxyz , cards**  
Copies edit buffer TEST to the OS/3 reader as LBL name "JOBXYZCARDS"
- COPY SRC/PAY040 , S, C: PAY040 . COB**  
Copy source element PAY040 from library SRC to a computer file named C:PAY040.COB

### Additional Considerations

The following options affect the COPY command:

- A** ASSEMBLER mode - use columns 1-72.
- C** COBOL mode - use columns 1-72.
- O** If output library element already exists, overwrite it without issuing a confirmation prompt.
- Q** Do not display any messages (Quiet).
- R** RPG mode - use columns 1-74; columns 1-5 set to spaces.
- S** Scratch input edit buffer.
- Z** Removes sequence numbering and update stamps when copying elements to a personal computer if the language has been set for the element.

TLIB does not supports copying from an DOS file to an DOS file, for obvious reasons. If you attempt an DOS to DOS copy you will receive the following message:



```

uw7test.tws - TIP WorkStation
Session Edit View Tools Help
TIP/ix?>tlib
'TIP/ix Librarian' - Version 1999/03/31 2.3 R0 - 0000
TLIB?>copy d:\test1 d:\test2
Please use the MSDOS command for this
TLIB?>
24.7 24x80 Ready MSG OVR CAP NUM SCRL

```

### Error Conditions:

The input file, element or edit buffer may not be found or the output file may not be available for use.

## DELETE - Delete Library Element

This command is used to delete an element from an library or to delete a TIP/ix edit buffer (also see the TLIB RECOVER command).

### Syntax:

```

DELEte P1 [,P2] [,P3]
ERASE P1 [,P2] [,P3]

```

Specify the item to be deleted from the following table:

| Parameters       | Description                                                                         |
|------------------|-------------------------------------------------------------------------------------|
| P1, P2, P3       |                                                                                     |
| File/elt [,type] | Input library element. Default is source type "S".                                  |
| group/name,E     | TIP/ix edit buffer specification Pseudo type code "E" indicates TIP/ix Edit Buffer. |

### Example:

```

DELETE JCS/MYJOB

```

Deletes element "MYJOB" from library "JCS".

### Additional Considerations

ERASE may be used as a synonym for DELEte.

The following options affect the DELETE command:

**Q** Do not display any messages (Quiet)

### Error Conditions:

The specified element may not exist or the file cannot be accessed.

## DIR - Display Library Directory

This command displays a directory of a library (or some subset of a library) at the terminal. A line, containing the module's name, type, comment and date and time last changed, will be displayed for each module selected.

### Syntax:

```
Dir file [,prefix] [,,printer]
```

### Where:

- file** The selected library name as defined in the TIP/ix catalogue.
- prefix** An element name prefix to be used to select some subset of the elements in the library. If the prefix parameter is omitted, the default is assumed to be "\*" - all elements.
- printer** The destination printer (default is AUX0 - full screen display). Other possibilities are, for example, PRNTR, or AUX1.

### Example:

Display a directory of all elements with names that begin with "TH\$", in the library file catalogued with the name "TIP".

```
DIR TIP,*TH$
```

DIR command output:

```

Uw7test - TIP WorkStation
Session Edit View Tools Help
[Icons]
Continue ▶Yes ▶No
Listing: TIP/DIRECTORY 1999/03/31 10:43
DISPLAYERR,S by tipixusr 2000/03/13 14:36
DPS-DEF-STATUS,S by tipixusr 2000/03/13 14:36
DPS-OPTIONS,S by tipixusr 2000/03/13 14:36
DPSSTATUSCOB,S by tipixusr 2000/03/13 14:36
DYNAMIC-CONTROL,S by tipixusr 2000/03/13 14:36
GC-INCDA,S by tipixusr 2000/03/13 14:36
GC-INREC,S by tipixusr 2000/03/13 14:36
GETFIELD,S by tipixusr 2000/03/13 14:36
GETLINE,S by tipixusr 2000/03/13 14:36
GETSCREEN,S by tipixusr 2000/03/13 14:36
IMA,S by tipixusr 2000/03/13 14:36
IMA74,S by tipixusr 2000/03/13 14:36
INFO-BUFFER,S by tipixusr 2000/03/13 14:36
OMA,S by tipixusr 2000/03/13 14:36
OMA74,S by tipixusr 2000/03/13 14:36
PACKET,S by tipixusr 2000/03/13 14:36
PAGE-STATUS,S by tipixusr 2000/03/13 14:36
PIB,S by tipixusr 2000/03/13 14:36
PIB74,S by tipixusr 2000/03/13 14:36
PSCRAT-INFO,S by tipixusr 2000/03/13 14:36
PUTSCREEN,S by tipixusr 2000/03/13 14:36
RCB,S by tipixusr 2000/03/13 14:36
RCBOPT,S by tipixusr 2000/03/13 14:36
RETURNERR,S by tipixusr 2000/03/13 14:36
REXMIT-BUFF,S by tipixusr 2000/03/13 14:36
SENDERROR,S by tipixusr 2000/03/13 14:36
STATUS-CELL,S by tipixusr 2000/03/13 14:36
STATUS-WORD,S by tipixusr 2000/03/13 14:36
1,15 30x80 P533 MSG DVR CAP NUM SCRL

```

## END - End TLIB Interaction

The End command terminates the TLIB program in a normal fashion.

### Syntax:

**End**

No parameters required.

## FDIR - Display Abbreviated Library Directory

This command displays a "fast" directory of a library (or some subset of a library) at the terminal.

Up to six element names are listed on each output line; the module name and type is displayed for each module selected.

### Syntax:

**Fdir file [,prefix] [,,printer]**

**Where:**

- file** The selected library name as defined in the TIP/ix catalogue.
- prefix** An element name prefix to be used to select some subset of the elements in the library. Default is list all elements. Elements are listed without regard to the type of the element.
- printer** The destination printer (default is AUX0 - full screen display). Other possibilities are, for example, PRNTR, or AUX1.

**Example:**

Display a directory of all elements with names that begin with "TH\$", in the library file catalogued with the logical file name "TIP".

```
FDIR TIP,*TH$
```

## HELP - Help for TLIB Commands

The Help command invokes the TIP/ix HELP system to display syntax help for the TLIB program and its clone transactions ([COPY](#), [PRINT](#) etc).

**Syntax:**

```
Help
```

No parameters required.

## LIST - List Input at Terminal

The LIST command displays data on the terminal.

**Syntax:**

```
List P1 [,P2] [,P3] [,printer]
```

Specify the item to be listed from the following table:

| Parameters       | Description                                                                                                                                              |
|------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|
| P1, P2, P3       |                                                                                                                                                          |
| omitted          | If parameters 1 through 3 are omitted, the terminal is used as an input device. Input is solicited line by line until is pressed to signal end of input. |
| File/elt [,type] | Input library element Default is source type "S".                                                                                                        |

|              |                                                                                                                        |
|--------------|------------------------------------------------------------------------------------------------------------------------|
| group/name,E | TIP/ix edit buffer specification Pseudo type code "E" indicates TIP/ix Edit Buffer.                                    |
| d:ffffff.eee | DOS file specification:                                                                                                |
| printer      | The destination printer (default is AUX0 - full screen display). Other possibilities are, for example, PRNTR, or AUX1. |

**Example:**

List the directory of the file catalogued with logical file name "JCS".

```
LIST JCS, ,D
```

**Additional considerations:**

The following options affect the LIST command:

|          |                                                         |
|----------|---------------------------------------------------------|
| <b>A</b> | ASSEMBLER mode - use columns 1-72                       |
| <b>C</b> | COBOL mode - use columns 1-72 columns 1-6 set to spaces |
| <b>Q</b> | Do not display any messages (Quiet)                     |
| <b>R</b> | RPG mode - use columns 1-74 columns 1-5 set to spaces   |
| <b>S</b> | Scratch input edit buffer                               |

**Error Conditions:**

The named element may not exist or the file cannot be accessed or the type may be incorrect.

**PRINT - Print Input**

This command creates a printed display of the input specification at the site printer, an auxiliary print device or any print destination recognized by the TIPPRINT interface (see TIPPRINT in the **TIP/ix Programming Reference**).

Unless inhibited by the appropriate option, output sent to the site printer is preceded by a separator (header) page to facilitate identification of the printout.

**Syntax:**

```
Print p1 [,p2] [,p3] [,printer] [,hdr]
 [,case] [,plen] [,copies]
```

**Where:**

Specify the item to be printed from the following table:

| Parameters | Description |
|------------|-------------|
|------------|-------------|



P1, P2, P3

omitted If parameters 1 through 3 are omitted, the terminal is used as an input device. Input is solicited line by line until is pressed to signal end of input.

File/elt [,type] Input library element Default is source type "S".

group/name,E TIP/ix edit buffer specification Pseudo type code "E" indicates TIP/ix Edit Buffer.

d:ffffff.eee DOS file specification:

**printer** The destination printer (default is the site printer PRNTR). Other possibilities are, for example, AUX0 or AUX1.

**hdr** YES/NO choice of a header (separator) page. Default is "N" if the destination is an AUX printer, otherwise, default is "Y".

**case** Choice of upper case translation. Default is translate to upper case ("U") if printer is the site printer, otherwise, default is no translation ("L").

**plen** The logical length of the page to be printed. The default is 66 lines per page.

**copies**

The number of copies to be generated; default is 1 copy.

**Example:**

Print source element named "ACTPAY" from the library with catalogued logical file name "JCS" on the terminal's auxiliary printer. No separator page is to be printed and all alphabetic characters are to be translated to upper case.

```
PR jcs/actpay, ,aux1,n,U
```

**Additional considerations:**

The declared format of a library element, an edit buffer or an DOS file (e.g., COBOL or Assembler or RPG etc) will cause the PRINT command to produce a printout that is more than a simple list of the lines.

The following options affect the PRINT command:

- A** ASSEMBLER mode - use columns 1-72
- C** COBOL mode - use columns 1-72 columns 1-6 set to spaces
- L** Print WITH line numbers
- M** Do not print heading lines (Minus)
- N** Do not print title page

- Q** Do not display any messages (Quiet)
- R** RPG mode - use columns 1-74 columns 1-5 set to spaces
- S** Scratch input edit buffer

### Error Conditions:

The specified element or edit buffer was not found or the file could not be accessed or the type is invalid.

## QUIT - End TLIB and LOGOFF

The QUIT command causes the TLIB program to discontinue prompting the user for more commands and terminates the TLIB program normally.

If the TLIB program was executing at stack level one (TLIB was NOT called by another program) the user will be logged off TIP/ix.

### Syntax:

`Quit`

No parameters are required.

## SETOF - Set TLIB Option Off

The TLIB command SETOF is used when TLIB is being executed interactively. In the interactive mode, TLIB prompts the terminal user for successive commands. If the desired command requires a particular option to be OFF, the only way the option can be turned OFF is by first using the SETOF command.

Of course, when TLIB is first invoked, all options are initially OFF. Once TLIB begins prompting for commands, an option that was ON may be set OFF using this command.

### Syntax:

`SETOF opt1 [,opt2] [,opt3] [, . . .]`

- opt** Each parameter represents up to 8 option characters that are to be set in the OFF state. See for a table of valid option characters.

There are seven parameters available after the command (SETOF). TLIB allows a parameter to SETOF to consist of one or more option characters.

The following two commands are identical:

```
SETOF L,N,Q
SETOF NLQ
```

**Example:**

```

Uw7test - TIP WorkStation
Session Edit View Tools Help
TIP/ix?▶tlib
'TIP/ix Librarian' - Version 1999/03/15 2.2 R0 - 0143
TLIB?▶setof N
ON=() OFF=(A C L M N O Q R S)
TLIB?▶
30,7 30x80 MSG OVR CAP NUM SCRL

```

In this example, several Print commands are to be issued. Before the print commands, options L (print line numbers) and N (no header separator pages) are set on. Before printing module PAY010, however, a header page is desired, so option N is set off.

## SETON - Set TLIB Option On

The TLIB command SETON is used when TLIB is being executed interactively. In the interactive mode, TLIB prompts the terminal user for successive commands. If the desired command can benefit from a particular option, the only way the option can be turned ON is by first using the SETON command.

Of course, when TLIB is first invoked, one or more options may be set on via the command line. Once TLIB begins prompting for commands, the command line option field is no longer accessible - hence the need for an explicit command to manipulate options.

**Syntax:**

```
SETON opt1 [,opt2] [,opt3] [, ...]
```

**Where:**

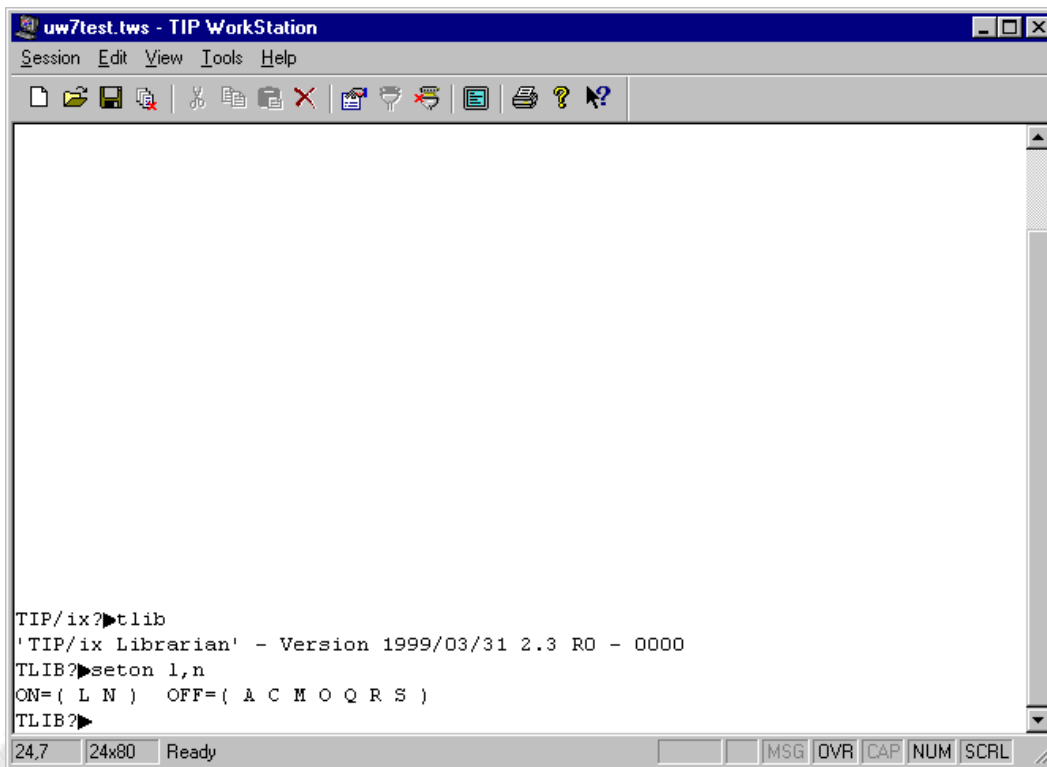
**opt** Each parameter represents up to 8 option characters that are to be set in the ON state. See TLIB Options for valid options.

There are seven parameters available after the command (SETON). TLIB allows a parameter to SETON to consist of one or more option characters.

The following two commands are identical:

**SETON L,N,Q**

**SETON NLQ**

**Example:**


```

uw7test.tws - TIP WorkStation
Session Edit View Tools Help
TIP/ix?▶tlib
'TIP/ix Librarian' - Version 1999/03/31 2.3 RO - 0000
TLIB?▶seton l,n
ON=(L N) OFF=(A C M O Q R S)
TLIB?▶
24,7 24x80 Ready
MSG OVR CAP NUM SCRL

```

In this example, several Print commands are to be issued. Before the print commands, options L (print line numbers) and N (no header separator pages) are set on.

---

## users - List user ids

The **users** utility displays a list of valid TIP/ix user ids and the name associated with each user id (as entered via the [smuser](#) utility in the field *Full name*). If no name exists in the user id record then the *Comments* field (as entered via [smuser](#)) will be displayed.

This information may be useful to users who wish to use the MAIL utility to send a message to an individual whose user id is not known or immediately obvious.

**Syntax:**

`users [user id]`

**Where:**

user id Specifies the user ids to list. Prefix notation may be used.

For example: users \*P

The default is to list all users.

**Example of users program output:**

```

Uw7test - TIP WorkStation
Session Edit View Tools Help
TF$USROA TIP/ix User List Effective 99/03/31 SCOTTC P641 15:54
=====
==User== =====Identification===== ==User== =====Identification=====
BJONES BJones
DEFAULT Default user record
FRED
GBOLAHAN GBOLAHAN
JDOE Jane Doe
MICHAELH Michael Hor
QACAT1 QA Test Account 1
QACAT2 <<maximum of 28 characters>>
ROOT TIP/ix Super User
SCOTTC Scott Craven
TIPIXUSR TIP/ix admin user
TIPQUEUE TIP/ix Queue Server

<--EOF--> F1:Refresh F2:Forward F3-Backward Msg-Wait:Exit
24.77 30x80 Ready P641 MSG OVR CAP NUM SCRL

```

**Additional Considerations:**

The listing is sorted into ascending order by user id.

## whoson - Who is on

The **whoson** program displays information about current users of the TIP/ix system. The information shown for each user includes the user name, terminal name, currently active program, input and output message counts, and average response time.

### Syntax 1 - online (from TIP/ix command line):

```
whoson[,opt] [-F|-T|-U] [-f|-l] [-]
 [*|!]value] [,dest]
```

### Syntax 2 - batch (from UNIX command line):

```
whoson [-F|-T|-U] [-f|-l] [-s] [-]
 [*|!]value]
```

### Where:

- opt** Command line option to restrict the display to specific subsets. This option is only available when whoson is run from a TIP/ix session (online).
  - dbms** Display users who are using the database interface.
  - ntp** Display terminals that are connected to or from another LOCAP by TIP distributed transaction processing.
- f** Display the last TIP function CALLED (by the application) at each TIP/ix session instead of the response time. This option can be useful in determining what an application program is doing. The *-f* must precede the *-* when both options are specified.
- l** Display the logon time (time **tipix** was started) for each TIP/ix session instead of the response time. This option can be useful in determining what an application program is doing. The *-l* must precede the *-* when both options are specified.
- i** report total I/Os
- v** report average I/O per transaction
- I** Display TCP/IP address instead of user name
- r** Display average response time. (default)
- F** Use the *value* parameter to match *files* that users are accessing.
- T** Use the *value* parameter to match *terminal names* only.

- b Use to display the time of the last input message
- t Use to display the TCP/IP port being used
- P Use to display the OS2200 PID value.
- h Use to display some short whoson usage information.
- s Only display toplevel information (related to execution stack level 0 and 1).
- Optional parameter to indicate that **whoson** is *not* to display the version information (this is often used when **whoson** output is directed to a file and the version information is not wanted.)

### Value

You can limit the output of the **whoson** program by specifying a user id, terminal, file, or program. The default is "\*" (display all TIP/ix sessions).

If the meaning of *value* is not restricted (by specifying -U, -T or -F), **whoson** attempts to match the value four ways (user id, terminal, file, and program).

- \* Match anything that starts with specified value.
- ! Match anything that does not start with specified value.

#### **user id**

Display all TIP/ix sessions operated by this user id (or user id prefix)

#### **terminal**

Display all TIP/ix sessions operating at this terminal (terminal name prefix).

**file** Display all TIP/ix sessions where this file has been assigned.

#### **program**

Display all TIP/ix sessions where this program is running at the highest stack level of the TIP/ix session.

**dest** **Whoson** uses TIPPRINT to AUX0 when run under TIP/ix. This means that when the optional DEST parameter is specified that it attempts to open it using TIPPRINT (if unsuccessful **whoson** writes the output to the file instead of the terminal. If the file does not exist, **whoson** attempts to create the file).

-i -v -r may be followed by a numeric value that is then used to limit which user sessions get displayed.

For example:

**whoson -r1.5** will only show session with response greater than 1 ½ seconds

**UNIX Example:**

```
$ whoson
whoson will use ROLL
```

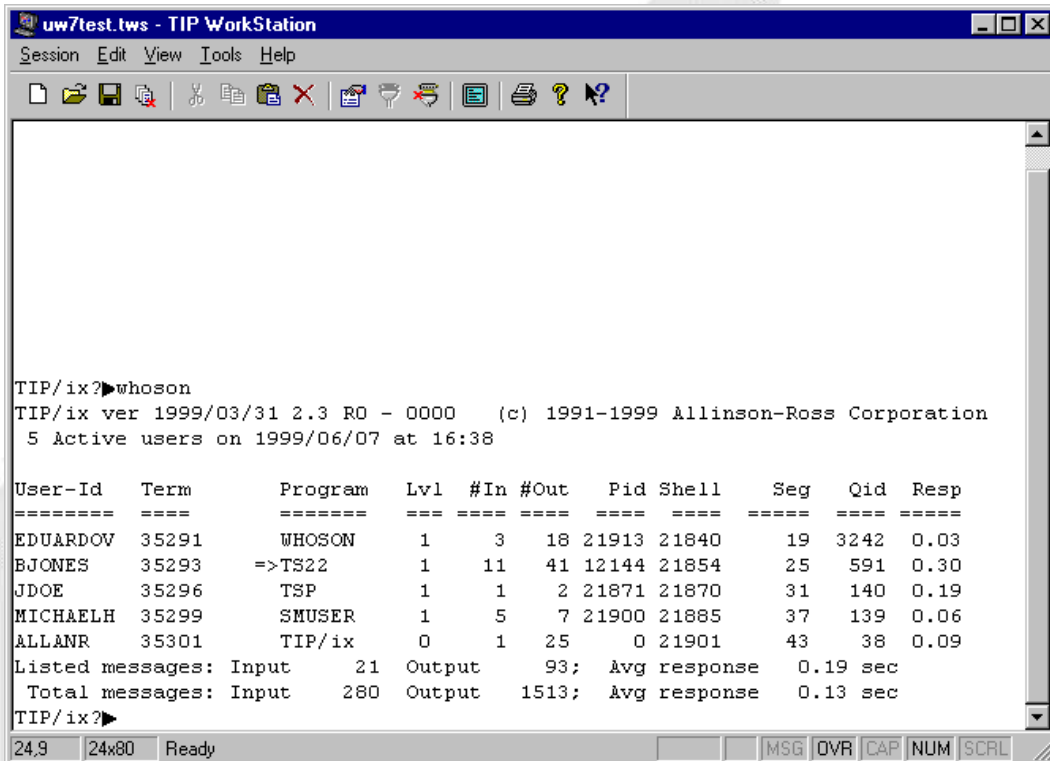
**TIP/ix Example:**

```
►WHOSON
 uses TIPPRINT to AUX0

►WHOSON * AUX1
 uses TIPPRINT to AUX1

►WHOSON * /tmp/foo
 sends output to a file
```

**whoson** produces the following display:



```
TIP/ix?►whoson
TIP/ix ver 1999/03/31 2.3 RO - 0000 (c) 1991-1999 Allinson-Ross Corporation
 5 Active users on 1999/06/07 at 16:38

User-Id Term Program Lvl #In #Out Pid Shell Seg Qid Resp
===== ===== ===== === === === ===== ===== ===== ===== =====
EDUARDOV 35291 WHOSON 1 3 18 21913 21840 19 3242 0.03
BJONES 35293 =>TS22 1 11 41 12144 21854 25 591 0.30
JDOE 35296 TSP 1 1 2 21871 21870 31 140 0.19
MICHAELH 35299 SMUSER 1 5 7 21900 21885 37 139 0.06
ALLANR 35301 TIP/ix 0 1 25 0 21901 43 38 0.09
Listed messages: Input 21 Output 93; Avg response 0.19 sec
Total messages: Input 280 Output 1513; Avg response 0.13 sec
TIP/ix?►
```

### Multiple programs at a single session

If multiple programs are active at a single session then a line will be displayed for each program (stack level). In the preceding example whoson reports that the user MARY is running the program [smfile](#) at



stack level 2 and the program [tcm](#) at stack level 1. This indicates that [tcm](#) called TIPSUB to invoke [smfile](#).

The session for MARY would be included in the display for the command “whoson smfile” but not the command “whoson tcm”.

### DTP connections (=>)

If a user is connected to another TIP (either TIP/30 or TIP/ix) system the characters => will precede the program name and the LOCAP name of the TIP system that the user is connected to is displayed in place of the program name. To connect to a TIP system the LOCAP name must be defined locally (via [smlocap](#)).

In the preceding example whoson reports that BJONES is connected to the TIP system with a LOCAP name of TS22 under the "Program" heading.

### IMS External Succession (\*)

In IMS, *external succession* is used to initiate programs after an input message arrives from a terminal. An asterisk (\*) preceding the program name indicates that the program will be initiated when an input message arrives.

In the preceding example, **whoson** reports that the program PARTINQ will be initiated upon the arrival of an input message.

### Reusable Programs (+)

If a program is running in serial reusable execution mode (see [smprog](#)) then a plus sign (+) will precede the program name.

However, the reusable attribute is most commonly used with IMS programs. IMS programs spend most of their time waiting (in *external succession*) for an input message. So you are more likely to see an "\*" than a "+" in a **whoson** report.

In the preceding example, **whoson** reports that the program ORDER is running in serial reusable execution mode.

### PIB-LOCK-INDICATOR (H,R,O)

The value of PIB-LOCK-INDICATOR at the time of the last call to TIP/ix is reported for each session in a column between “Program” and “Lvl”. Often this column will be blank.

|              |              |
|--------------|--------------|
| <b>H</b>     | Hold.        |
| <b>R</b>     | Released.    |
| <b>O</b>     | Rolled back. |
| <b>blank</b> | Committed.   |

In the preceding example, **whoson** reports that the last PIB-LOCK-INDICATOR value for the session for the user ALLANR was an "H" (PIB-HOLD).

### TQL Programs (@)

An "@" character in the first position of the program name indicates a TQL program.

The following is a description of the fields reported by **whoson**.

**user id**

TIP/ix user id (see [smuser](#))

**Term** Terminal name (PIB-TERM-NAME) (see [smterm](#)).

**Program**

TIP/ix program being executed. If more than one program is active then each program and its stack level will be reported on separate lines. (see **TIPSUB** in the TIP/ix Programming Reference).

An "@" character in the first position of the program name indicates a TQL program.

**Lvl** Program execution stack level. Each TIPSUB increments the stack level and each TIPRTN decrements the stack level.

**#In** Number of input messages for this TIP/ix session.

**#Out** Number of output messages for this TIP/ix session.

**Pid** Unix process identifier for the processing running the application program.

**Shell** Unix process identifier for the process running the TIP/ix shell.

**Seg** Shared memory identifier (debugging information).

**Qid** Message Queue identifier (debugging information).

**Resp** Average response time for the TIP/ix session.

**Func** Last TIP function called at the TIP/ix session. Only displayed when **-f** option is specified.

**Login** Login time for the session. This is the time that **tipix** was run. Only displayed when **-l** option is specified.

**Example of requesting whoson report for a single user id.**

```

TIP/ix?>whoson SCOTT
TIP/ix ver 1999/03/15 2.2 R0 - 0143 (c) 1991-1999 Allinson-Ross Corporation
 1 Active user on 1999/03/31 at 10:54

User-Id Term Program Lvl #In #Out Pid Shell Seg Qid Resp
----- ---- ----- --- --- --- --- --- --- ---
SCOTT _TCP641 WHOSON 1 5 16 1045 1039 31 40 0.18
Listed messages: Input 5 Output 16; Avg response 0.18 sec
Total messages: Input 105 Output 164; Avg response 0.71 sec
TIP/ix?>

```

### Example output using the -f option and a program name.

```

Uw7test - TIP WorkStation
Session Edit View Tools Help
TIP/ix?▶whoson -f whoson
TIP/ix ver 1999/03/15 2.2 R0 - 0143 (c) 1991-1999 Allinson-Ross Corporation
 1 Active user on 1999/03/31 at 10:55

User-Id Term Program Lvl #In #Out Pid Shell Seg Qid Func
===== ==== ===== == == == == == ===== == ==
SCOTTTC _TCP641 WHOSON 1 8 27 1049 1039 31 40 roll
Listed messages: Input 8 Output 27; Avg response 0.15 sec
Total messages: Input 108 Output 175; Avg response 0.69 sec
TIP/ix?▶
30.9 30x80 Ready P641 MSG QVR CAP NUM SCRL

```

#### Additional Considerations:

The **whoson** utility is a UNIX program that can execute outside the TIP/ix shell. If executed outside TIP/ix, **whoson** attempts to determine if TIP/ix is running; if TIP/ix is *not* running, **whoson** executes the UNIX "who" command instead.

The user response time (since logon) is calculated as:

#### user response time =

(Sum of (time of completed delivery of first output msg after an input time of the input msg)) / number of input msgs

## wmi - Who am I

The **wmi** program displays information about a TIP/ix user, the TIP/ix system, and the UNIX system. **wmi** should also be used to determine the latest version number of TIP/ix if it has been previously updated using the update pack.

The following information is displayed for the TIP/ix session invoking **wmi**:

- the user id of the TIP/ix session
- the active groups and any associated group security levels (see smuser and groups)
- the user's security level
- the active account for the user (PIB-ACCOUNT-NUMBER)
- the terminal id (PIB-TID) and the normalized terminal address (see smterm)
- the current UNIX working directory

The following general TIP/ix system attributes are displayed:

- Site name of the TIP/ix system from \$TIPROOT/tipboot.sys (PIB-SITE-NAME)
- TIP/ix version.
- TIP/ix LOCAP name as defined by the parameter LOCAP in \$TIPROOT/conf/tipix.conf . This is the value of PIB-LOCAP.

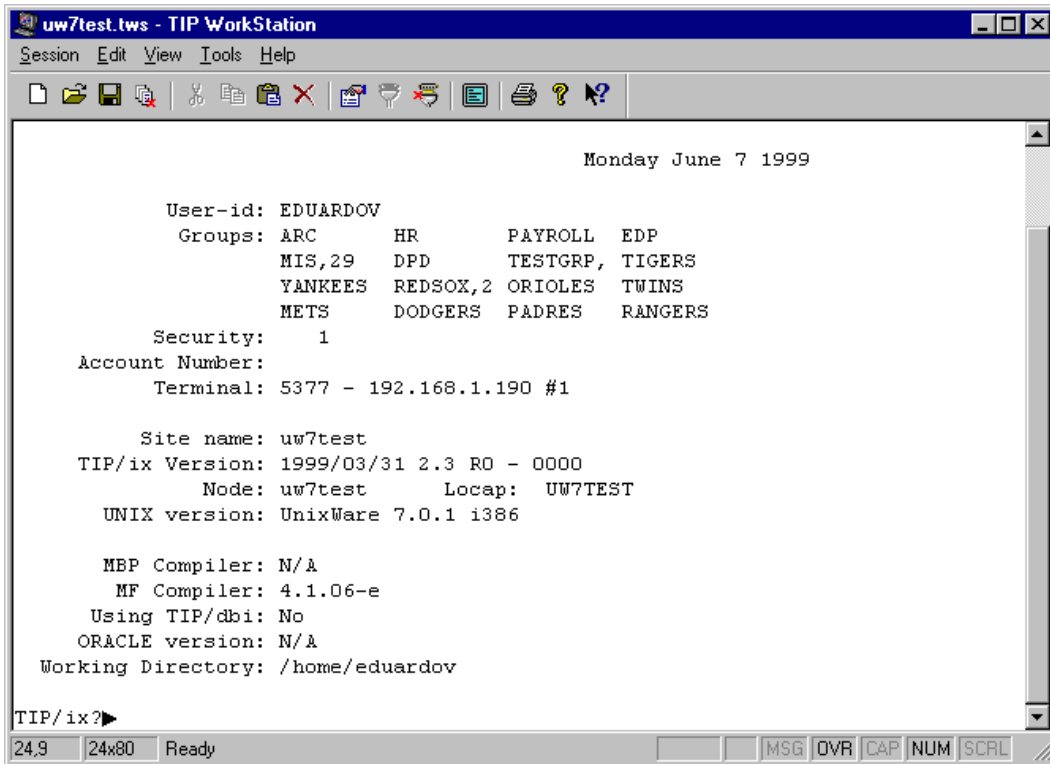
The following general UNIX system attributes are displayed:

- node name (uname -n)
- UNIX system name (uname -s)
- operating system release (uname -r)
- computer hardware name (uname -m)

**Syntax:**

**wmi**

**wmi** produces a display similar to the following:



```

uw7test.tws - TIP WorkStation
Session Edit View Tools Help
Monday June 7 1999

User-id: EDUARDOV
Groups: ARC HR PAYROLL EDP
 MIS,29 DPD TESTGRP, TIGERS
 YANKEES REDSOX,2 ORIOLES TWINS
 METS DODGERS PADRES RANGERS

Security: 1
Account Number:
Terminal: 5377 - 192.168.1.190 #1

Site name: uw7test
TIP/ix Version: 1999/03/31 2.3 RO - 0000
Node: uw7test Locap: UW7TEST
UNIX version: UnixWare 7.0.1 i386

MBP Compiler: N/A
MF Compiler: 4.1.06-e
Using TIP/dbi: No
ORACLE version: N/A
Working Directory: /home/eduardov

TIP/ix?
24.9 24x80 Ready MSG OVR CAP NUM SCRL

```

### Groups:

The active groups are displayed in order from top to bottom and from left to right. Therefore, in the preceding screen “ARC” is the 1st active group and “RANGERS” is the 16th active group.

If a number is displayed to the right of a group name then that represents the user’s security level within that group. If no number is displayed next to a group then the user’s general security level applies to access within that group. In the preceding example the security level within the group “ARC” is 9 (value displayed for the field “Security”) and the security level within the group “MIS” is 29.

### UNIX version:

This field displays the UNIX system name, the operating system release, and the computer hardware name. The value displayed matches the output of the UNIX command “uname -srm”.

---

## xfer - Data Transfer Facility

The **xfer** utility program may be used to transfer certain types of information between two TIP systems (TIP/ix and/or TIP/30) that are connected using distributed transaction processing (DTP).

Use **xfer** to transfer the following types of data between TIP systems:

**Library element**

Source module in a TIPFCS library file.

**Screen format**

Any TIP screen format.

To be able to use this facility, the two TIP systems must be properly connected using TIP distributed transaction processing (DTP), see separate description of this topic.

For transfers between two TIP/30 systems, see the TIP/30 documentation.

**Additional considerations:**

TIPFCS includes a file type of library which associates an 8 character TIPFCS name with a UNIX directory. Files in the directory can be accessed as library elements using TIPFCS. However, library element names are restricted (by the TIPFCS API) to 8 characters in length.

All command line parameters of **xfer** are converted to upper case so it is not possible to use **xfer** to copy a library element that contains any lower case letters in its file name.

## xfer a Library Element

The following syntax is used to transfer a TIPFCS Library element between TIP systems.

**Syntax:**

**XFER** *direction*, *lfile*, *lelt*, *ltype*, **LOCAP**, *rfile*, *relt*

**Where:**

**direction**

Whether data is to be sent or received in relation to the LOCAP where the **xfer** program is running:

Send    Send data from this LOCAP.

Receive    Receive data from other LOCAP.

**lfile**    The logical file name of the local library.

**lelt**    The local element name.

**ltype**    The local element type:

**S**    Source element.

**M or P**

Macro or proc. This type is only valid when a

TIP/30 system is involved in the file transfer. Macro or Proc is a valid library type on OS/3.

**LOCAP**

The LOCAP name of the remote LOCAP (the other LOCAP).

**rfile** The logical file name of the library on the remote LOCAP.

**reft** The remote element name (omitted if type is "L").

**Example:**

```
xfer S SRCLIB,TSTPROG,TEST,TSTLIB
```

Copy the source element named "TSTPROG" from the library "SRLIB" on the local TIP system to the library named "TSTLIB" on the remote TIP system identified by the LOCAP name of "TEST".

## xfer a TIP/30 Screen Format

The following syntax is used to transfer a TIP/30 Screen Format between LOCAPs.

**Syntax:**

```
xfer direction lgroup/lscreen/MCS LOCAP rgroup/rscreen
```

**Where:**

**direction**

Whether the screen format is to be sent or received (in relation to the LOCAP where the xfer program is running).

**Send**

Send data from this LOCAP.

**Receive**

Receive data from other LOCAP.

**lgroup**

The local group name associated with the screen format.

**lscreen**

The local screen format name.

**MCS** Required (literally) to identify that this is an xfer of a TIP/30 screen format.

**LOCAP**

LOCAP name of the remote LOCAP (the other LOCAP).

**rgroup**

The remote group name associated with the screen format.



**rscreen**

The remote screen format name.

**Example:**

```
xfer S TIP$$/TF$FSE0A,MCS,TEST,TIP$$/FSEFMT
```

Copy the screen format named "TF\$FSE0A" in the group "TIP\$\$" on the executing LOCAP to the remote LOCAP named "TEST", using the screen format name FSEFMT in the group TIP\$\$.

**Additional considerations:**

You can change the group or name of a screen format during the transfer to the remote TIP system.

**Example:**

```
xfer SEND TIP$$/TF$TSP0A,MCS TEST TIP$$/JUNK
```

## TIP/ix System Modules

The following table lists the executable modules (binaries) supplied with TIP/ix and indicates where the module may be used. The transaction names may differ from the module names.

### UNIX

The module may be executed as a UNIX program.

### TIP/ix

The module may be executed as a TIP/ix transaction program. Some modules can be used both ways.

### Internal

the module is only for TIP/ix internal usage and should never be executed on its own.

| Module       | UNIX | TIP/ix | Internal |
|--------------|------|--------|----------|
| aft          | ✓    | ✓      |          |
| apb          | ✓    | ✓      |          |
| artie        | ✓    |        |          |
| <b>B</b>     |      |        |          |
| boottip      | ✓    |        |          |
| <b>C</b>     |      |        |          |
| calendar     |      | ✓      |          |
| connect      |      | ✓      |          |
| <b>D</b>     |      |        |          |
| damsrv       |      |        | ✓        |
| dbischema    | ✓    |        |          |
| dbisubschema | ✓    |        |          |
| dbixsvr      | ✓    |        |          |
| dcheck       | ✓    |        |          |
| ddu          |      | ✓      |          |
| defkey       |      | ✓      |          |
| dfu          |      | ✓      |          |
| die          | ✓    | ✓      |          |
| disrv        |      |        | ✓        |
| dtprfa       |      |        | ✓        |
| <b>E</b>     |      |        |          |
| ebcasc       | ✓    |        |          |

|          |   |   |   |
|----------|---|---|---|
| emptyq   |   |   | ✓ |
| ej       | ✓ | ✓ |   |
| <b>F</b> |   |   |   |
| fclose   | ✓ | ✓ |   |
| fcsdbug  | ✓ |   |   |
| fin      |   | ✓ |   |
| fixperms | ✓ |   |   |
| fopen    | ✓ | ✓ |   |
| free     |   | ✓ |   |
| freset   | ✓ | ✓ |   |
| fse      |   | ✓ |   |
| fxp      | ✓ |   |   |
| <b>G</b> |   |   |   |
| genmain  | ✓ |   |   |
| gtr0     |   | ✓ |   |
| gtr1     |   | ✓ |   |
| gtr2     |   | ✓ |   |
| gtnw1    |   | ✓ |   |
| gtsi0    |   | ✓ |   |
| gtsi1    |   | ✓ |   |
| <b>I</b> |   |   |   |
| ims11    |   | ✓ |   |
| imsmnu   |   |   | ✓ |
| isonline | ✓ |   |   |
| isreorg  | ✓ |   |   |
| <b>J</b> |   |   |   |
| jrnswap  | ✓ |   |   |
| <b>L</b> |   |   |   |
| lclose   | ✓ | ✓ |   |
| lopen    | ✓ | ✓ |   |
| lreset   | ✓ | ✓ |   |
| <b>M</b> |   |   |   |
| mbpfcs   |   |   | ✓ |
| mbpisrv  |   |   | ✓ |

|            |   |   |   |
|------------|---|---|---|
| mdamsrv    |   |   | ✓ |
| menu       |   | ✓ |   |
| menuar     |   | ✓ |   |
| menudef    |   | ✓ |   |
| msg        | ✓ | ✓ |   |
| msgar      | ✓ |   |   |
| <b>N</b>   |   |   |   |
| notimpl    |   | ✓ |   |
| <b>P</b>   |   |   |   |
| pingtip    | ✓ |   |   |
| precob     | ✓ |   |   |
| purge      | ✓ | ✓ |   |
| <b>R</b>   |   |   |   |
| readjrn    | ✓ |   |   |
| rfaxlt     | ✓ |   |   |
| rollfwd    | ✓ |   |   |
| <b>S</b>   |   |   |   |
| samsrv     |   |   | ✓ |
| schema     | ✓ |   |   |
| scratch    |   | ✓ |   |
| seqreorg   | ✓ |   |   |
| smfile     |   | ✓ |   |
| smgrp      |   | ✓ |   |
| smlcap     |   | ✓ |   |
| smprint    |   | ✓ |   |
| smprog     |   | ✓ |   |
| smque      |   | ✓ |   |
| smsec      |   | ✓ |   |
| smtterm    |   | ✓ |   |
| smuser     |   | ✓ |   |
| sorter     |   | ✓ |   |
| status     | ✓ | ✓ |   |
| stopsystem | ✓ |   |   |
| stoptip    | ✓ |   |   |

**T**

|            |   |   |   |
|------------|---|---|---|
| tcm        |   | ✓ |   |
| tfd        | ✓ |   |   |
| tipback    |   |   | ✓ |
| tipbatch   |   |   | ✓ |
| tipbatsv   |   |   | ✓ |
| tipcd      |   |   | ✓ |
| tipclear   |   |   | ✓ |
| tipctl     | ✓ |   |   |
| tipdtp     |   |   | ✓ |
| tipdump    | ✓ |   |   |
| tipenv     |   |   | ✓ |
| tipfcs     |   |   | ✓ |
| tipfe      | ✓ |   |   |
| tipflg     |   | ✓ |   |
| tipforkr   |   |   | ✓ |
| tipgate    | ✓ |   |   |
| tiphelp    |   | ✓ |   |
| tipinstall | ✓ |   |   |
| tipix      | ✓ |   |   |
| tipixtcp   |   |   | ✓ |
| tipixtty   |   |   | ✓ |
| tiplib     |   | ✓ |   |
| tipmcsio   |   |   | ✓ |
| tipmon     |   |   | ✓ |
| tippack    | ✓ |   |   |
| tippager   |   | ✓ |   |
| tippcstm   |   |   | ✓ |
| tipque     |   |   | ✓ |
| tipsub     |   |   | ✓ |
| tipscript  |   |   | ✓ |
| tipstartup |   |   | ✓ |
| tipupack   | ✓ |   |   |
| tipxfer1   |   | ✓ |   |

|               |   |   |
|---------------|---|---|
| tipxfer2      |   | ✓ |
| tqladmin      | ✓ |   |
| tqlccp        |   | ✓ |
| tqlerr        |   | ✓ |
| tqlexec       |   | ✓ |
| tqlidump      |   | ✓ |
| tqlmon        | ✓ |   |
| tqlrun        | ✓ |   |
| tqlsdump      |   | ✓ |
| tqlupsel      |   | ✓ |
| tsi           | ✓ |   |
| tsp           | ✓ |   |
| tsprnt        | ✓ |   |
| ttuhlp        | ✓ |   |
| txmfm         | ✓ |   |
| <b>U</b>      |   |   |
| users         | ✓ |   |
| <b>W</b>      |   |   |
| whoson        | ✓ | ✓ |
| wmi           |   | ✓ |
| <b>X</b>      |   |   |
| Xt-m11 (MAIL) |   | ✓ |
| Xt-m19        |   | ✓ |
| xtnote (NOTE) | ✓ |   |
| xtuqxf        |   | ✓ |