

Software Tools Subsystem
Version 6 to Version 7 Conversion Guide

Daniel H. Forsyth, Jr.
T. Allen Akin

School of Information and Computer Science
Georgia Institute of Technology
Atlanta, Georgia 30332

March, 1980

TABLE OF CONTENTS

Introduction	1
Extensions to Pathname Syntax	1
Pathnames Referring to Disk Files	1
Pathnames Referring to Other Devices	2
Status of Version 6 Commands	3
Obsolete Commands	3
Superseded Commands	4
Modified Commands	5
Enhanced Commands	6
Unchanged Commands	8
New Commands	9
Status of Version 6 Subroutines	11
Obsolete Routines	11
Superseded Routines	11
Modified Routines	13
Enhanced Routines	14
Unchanged Routines	15
New Routines	15
Converting 'Rf' Programs to 'Rp'	19

Introduction

The release of Version 7 of the Software Tools Subsystem marks the end of a period of intensive development. A number of changes and enhancements have been made to the Subsystem since the August, 1979 release of Version 6; they should provide considerably improved performance and increased functionality.

The price that must be paid for this improved system is the effort required to resolve a number of incompatibilities. All Subsystem object code must be relinked (in some cases recompiled), and all shell programs must be checked for changes caused by additions to existing commands. This Conversion Guide represents an attempt to outline the incompatibilities and the means for upgrading from Version 6 to Version 7.

There are four major areas of incompatibility: Subsystem pathname syntax has been changed to allow references to disks by packname and to access devices other than disk; a number of commands have been deleted, altered, or enhanced; several library routines have been changed; and a new Ratfor preprocessor accepting a slightly different language is now the standard tool for Subsystem programming.

As a learning aid for conversion, starting with Version 7 the Reference Manual and the User's Guide will be marked with revision bars and deletion asterisks. Careful perusal of both documents is recommended.

Extensions to Pathname Syntax

After the Version 6 release, it became necessary to extend Subsystem pathnames to handle disk selection in a more flexible way. In addition, the ability to specify devices other than disk was desirable. Unfortunately, this could not be done in a clean, upward-compatible manner. The following subsections outline the differences and extensions that most affect the Version 6 user. All users are referred to the User's Guide to the Primos File System for additional information; type

```
guide -p fs | os >/dev/lps/f
```

Pathnames Referring to Disk Files

Perhaps the most inconvenient feature of Version 6 pathnames was the lack of the ability to specify a logical disk by its packname. One would frequently have to scan a "status disks" listing looking for the appropriate pack, then convert the octal logical disk number to decimal for the Subsystem pathname handler.

The old-style full pathname looked something like

```
<disk_number>/ufd/dir/.../dir/file
```

The new-style pathname assumes that there is a "root" directory above all started-up disks, and that this directory is named "/". There are then four alternatives for a full pathname:

```
/<disk_number>/ufd/dir/.../dir/file  
/<packname>/ufd/dir/.../dir/file  
/*ufd/dir/.../dir/file  
//ufd/dir/.../dir/file
```

You may now identify a file on a particular logical disk by specifying the packname or logical disk number of that disk as the first node of the pathname. Note that new-style disk numbers are in octal, not decimal; thus, the disk numbers in a "status disks" listing may be used without conversion. Furthermore, you may identify a file on the disk to which you are currently attached by using a star ("*") in place of a packname.

If you require the old top-level directory search procedure, you must specify an empty packname; this causes the MFDs of all logical disks to be searched for the first directory in the pathname. To illustrate, the Version 6 pathname

```
/system
```

must be specified in Version 7 as

```
//system
```

This is the primary area of incompatibility between Version 6 and Version 7 pathnames, and the one which will probably give you the most difficulty; not only must you convert pathnames hidden in any existing programs and command files, but you must also retrain yourself to use the new syntax.

As with Version 6, backslashes at the beginning of a pathname may be used to "back up" the directory tree before beginning the search for a file. However, at Version 7 the "dot file" kluge used to implement this feature is no longer needed. If you have any dot files (files named "." containing the full pathname of the directory in which they appear), you should delete them.

Pathnames Referring to Other Devices

As a special case, full pathnames beginning with "/dev" are interpreted as names of additional file system "devices." For example, "/dev/null" refers to a "null" device, which sinks all data written to it and always returns end-of-file when read. The standard ports may be named by pathnames of the form "/dev/stdin" or "/dev/stdout". Furthermore, the user's terminal now has the name "/dev/tty".

The most useful of file system devices at the moment is the line printer spooler device. This device has the name `"/dev/lps"`, possibly followed by a number of spooler options separated by slashes, blanks, or dashes. For example, `"/dev/lps/f"` refers to a print file with Fortran forms control. Use of `"/dev/lps"` rather than one of the spooling programs `'sp'` or `'pr'` is highly recommended, since it typically eliminates one entire copy of the data to be spooled, thus improving run time and disk space utilization. In fact, `'sp'` and `'pr'` have been re-implemented as shell programs that use `"/dev/lps"`; this has caused some minor changes in their command-line syntax. See the Reference Manual entries for `'sp'`, `'pr'`, and `'open'` for further details.

Status of Version 6 Commands

This section summarizes the user-visible changes that have been made to Subsystem commands for Version 7. It is divided into several subsections: obsolete commands, superseded commands, modified commands, enhanced commands, and unchanged commands. The final subsection is a summary of commands that are new for the Version 7 release.

Obsolete Commands

The commands in this subsection were part of the Version 6 Subsystem, but are not included in the Version 7 release. Most of them were used only by certain shell programs and have outlived their usefulness. In the other cases, the commands were relics of past Subsystems, and either were no longer useful, or no longer worked.

`alldoc`

Its use was restricted to `"system"` and `"=doc=/print/man"` and `"=doc=/print/guide"` now perform its function.

`edit`

It was a proper subset of `'ed'` and it didn't work.

`fixdot`

Dot files are gone. (three cheers!)

`gloss`

Didn't seem to be useful.

`hscript`

It was used only by `'help'` to strip out the description part of a reference manual entry.

lnews

System 'news' serves its purpose just as well.

login

Anyone running an accounting package prohibits 'login' while logged in.

lpublish

System 'publish' serves its purpose just as well.

lsubscribe

System 'subscribe' serves its purpose just as well.

mkci

This command hasn't been useful for about three years.

tabs

No one ever implemented tab expansion.

whoami

'Login_name' does exactly the same thing, and is faster.

Superseded Commands

The commands in this section are not part of the Version 7 Subsystem; their functionality has been subsumed by other commands. Each entry describes the command and options you can use to get the same results.

catlist

Use "cat -n".

ccnt

Use "tc -c".

compare

Use "diff -vc".

du

Use "lf -w".

dup

Use "term -noecho" or "term -echo".

lcnt

Use "tc -l".

monitor

Use "mon".

psp

Use 'lps' or Primos 'prop' and 'spool'.

total
Use "stats -tq".

wcnt
Use "tc -w".

Modified Commands

The commands listed in this section have been modified for the Version 7 release and are no longer completely compatible with their Version 6 counterparts. Each entry gives a brief description of the changes, but before using any of these commands, please check the corresponding Reference Manual entry to be sure of the command's exact behavior.

as6800
The 6800 cross-assembler now accepts a "-l" option, causing an assembly listing to be produced on standard output.

chat
The syntax for protection attributes has been changed to "{t|w|r|a}[/]{t|w|r|a}"].

day
The argument syntax has been slightly changed.

find
Output lines are not labelled with the input file name unless the "-v" option is specified.

hd
'Hd' now accepts packnames or logical disk numbers.

help
It runs much faster and can now access the subroutines section of the Reference Manual and search for an arbitrary pattern in the manual index. Argument syntax is slightly different.

lam
'Lam' can laminate more than two files and can insert arbitrary delimiters between lines. It no longer reuses the last line of the shorter of two input files.

lf
The output format of the file type and protection attributes have been changed.

log
Log files now reside in the user's profile directory, rather than in his login directory.

mkdir

'Mkdir' has a slightly different argument syntax.

se

'Se' now has support for the IBM 3101 ASCII terminal, and has an "m" option for reading messages sent by the 'to' command. The "insert newline" (control-n) key has been redefined to control-underscore to allow for several more control characters.

slice

'Slice' now accepts arbitrary patterns for starting and ending delimiters, and allows a choice as to whether the delimiting lines are included in the output.

sp

'Sp' now requires an argument consisting of a single slash to separate its list of file names from its list of spool options. You may want to use the pathname "/dev/lps" instead of 'sp' anyway.

Enhanced Commands

Commands in this section have been functionally enhanced for Version 7, but remain compatible with their Version 6 counterparts.

cat

'Cat' can now accept lists of file names and can print a heading line between files.

chown

'Chown' can change the owner passwords in an entire subtree of the file system.

cmp

New alternatives for relational operators have been added.

cp

'Cp' can now copy file system subtrees, including segment directories. 'Cp' now always copies date of modification and protection attributes.

del

'Del' can delete file system subtrees, including segment directories.

dnum

'Dnum' handles cartridge module disk numbers and does considerable error checking.

dprint

'Dprint' can print acceptable overstruck graphics for many Greek letters and can handle subscripting and

superscripting from the text formatter.

error

'Error' can now pass back a user-selectable error status to the shell.

fc

'Fc' has options for controlling debugging mode and optimization.

fmt

'Fmt' now has nestable functions, diversions, number registers and several new commands. It can produce most Greek letters and subscripting and superscripting on a Diablo using 'dprint'.

history

'History' now uses 'fmt' to format new history entries, and 'phist' to print the history file.

iota

'Iota' can now generate a sequence of numbers in an arbitrary range, in an arbitrary output format.

ld

'Ld' can link PL/I Subset G programs.

mail

'Mail' accepts multiple addressees and checks the validity of each one.

mklib

'Mklib' now accepts arbitrary pathnames, rather than being restricted to names in the current directory.

pr

'Pr' is now a shell program, and can also accept spool options.

print

'Print' has an option to produce output suited either for a terminal or a line printer.

shtrace

There are more shell trace options.

speling

'Speling' now accepts file name arguments.

stacc

'Stacc' now has an 'ext_term' declaration to declare externally-defined terminal symbols, and accepts actions after a production.

Version 7 Conversion Guide

subscribe

'Subscribe' now balks at duplicate subscriptions.

term

'Term' now handles changing "escape" and "retype" control characters. It can also set and display the disposition of the QUIT key.

to

'To' now checks for a legal user name before sending the message.

xref

'Xref' can now cross-reference extremely large programs without killing itself.

Unchanged Commands

This section lists the commands that have no user-visible changes made for Version 7.

Version 7 Conversion Guide

ar	arg	args
as11	as8080	banner
basys	bye	case
cd	change	clock
cn	common	como
copy	crypt	ctime
cto	declare	declared
detab	drop	dumpls
dumpsv	e	echo
ed	ek	else
entab	eval	exit
fdmp	field	file
files	forget	fos
goto	hp	if
include	index	installation
intel	join	kill
kwic	length	lex
lib	line	lk
locate	login_name	macro
mkusr	mot	mt
news	nargs	opt6800
opt8080	os	out
pause	pg	ph
profile	publish	pwd
pword	quote	rcl
rf	rmusr	rnd
rot	save	scroll
sema	set	sh
size	sort	source
sspl	ssr	stop
substr	symbols	systat
tail	take	tee
term_type	tlit	time
ts	uniq	unoct
unrot	us	usage
vars	when	whereis
who	whois	x

New Commands

This section list commands that are new for Version 7.

alarm

Makes a terminal into a very expensive alarm clock.

argsto

Prints arguments up to a user-specifiable delimiting argument.

batch

Interface to the Primos batch subsystem.

Version 7 Conversion Guide

clear
Clears the terminal screen.

col
Produces multi-column output.

copyout
Copies command output directly to a file in the spool queue.

diff
Compares files.

guide
Accesses the Subsystem User's Guides.

imi
Produces down-line load stream for an IMI Prom Programmer.

lps
Lists spool queues and cancels print files.

memo
Maintains files of personal reminders.

mktree
Converts a Subsystem path name into a Primos tree name.

mon
Displays system usage statistics.

moot
Initial version of a teleconferencing system.

passwd
Sets non-owner passwords.

phist
Prints the Subsystem history file in a readable format.

pllc
Compiles PL/I Subset G programs.

pllcl
Compiles and links PL/I Subset G programs.

rfl
Preprocesses, compiles, and links programs written in the "new" Ratfor.

rp
Preprocesses programs written in the "new" Ratfor.

sep
Supports separate compilation of parts of large programs.

st_profile
Analyzes Ratfor statement count profiles.

stats
Performs simple statistical analyses.

tc
Counts pages, lines, words, and characters in text files.

Status of Version 6 Subroutines

This section summarizes the user-visible changes to the Subsystem library routines. It is divided into several subsections: obsolete routines, superseded routines, modified routines, enhanced routines, and unchanged routines. The final subsection is a summary of routines that are new for the Version 7 release.

Obsolete Routines

The routines listed here were only used by other library routines. Since their services are no longer required, they have been deleted.

din\$	dinc\$	dout\$
doutc\$	fbuf\$	gbuf\$
getnod	ibuf\$	inperr
spool	tinc\$	toutc\$

Superseded Routines

The following routines have been subsumed by other more powerful routines. Each entry names the Version 7 routine that performs the same function.

atof
Use 'ctod' or 'decode'.

atoi
Use 'ctoi', 'gctoi', or 'decode'.

atol
Use 'ctol', 'gctol', or 'decode'.

Version 7 Conversion Guide

ator
Use 'ctor' or 'decode'.

ctof
Use 'ctod' or 'decode'.

find
Use 'findf\$'.

flush
Use 'flush\$'.

ftoc
Use 'dtoc' or 'encode'.

gatoi
Use 'gctoi' or 'decode'.

gatal
Use 'gctol' or 'decode'.

getflt
Use 'input'.

getint
Use 'input'.

getlnt
Use 'input'.

getrea
Use 'input'.

getstr
Use 'input'.

putflt
Use 'print'.

putint
Use 'print'.

putlit
Will be obsolete in Version 8. Use 'putlin' instead.

putlnt
Use 'print'.

putoct
Use 'print'.

putpak
Use 'print'.

putrea
Use 'print'.

string
Use 'ptoc'.

Modified Routines

The routines listed in this section have been modified so that they are no longer compatible with their Version 6 counterparts. Although each entry briefly describes the changes that have been made, you should examine the corresponding Reference Manual entries to determine the exact behavior of the routines.

chkarg
'Chkarg's behavior when no key-letter arguments are found has been changed.

dgetl\$
'Dgetl\$' has been completely rewritten. See the Reference Manual entry for complete details.

dputl\$
'Dputl\$' has been completely rewritten. See the Reference Manual entry for complete details.

error
A call to 'error' now terminates all currently executing shell files by passing an error status back to the shell. It also accepts either an EOS-terminated unpacked string, or a period-terminated packed string.

gctoi
The argument order has been changed from "gctoi (str, base, i)" to "gctoi (str, i, base)". Optional base indicators in the input are separated from the number by the letter 'r' instead of being enclosed by parentheses (e.g., '16rff' instead of '(16)ff').

gctol
The argument order has been changed from "gctol (str, base, i)" to "gctol (str, i, base)". Optional base indicators in the input are separated from the number by the letter 'r' instead of being enclosed by parentheses (e.g., '16rff' instead of '(16)ff').

getto
'Getto' now takes two additional arguments: it returns a password associated with the last node of the path and returns a flag indicating whether it changed the attach point.

gitoc

'Gitoc' now takes four arguments instead of five. If the "base" argument is negative, the number to be converted is assumed to be unsigned; otherwise, it assumed to be signed two's complement.

gltoc

'Gltoc' now takes four arguments instead of five. If the "base" argument is negative, the number to be converted is assumed to be unsigned; otherwise, it assumed to be signed two's complement.

input

'Input' has been completely rewritten. See the Reference Manual entry for complete details.

mkdir\$

'Mkdir\$'s calling sequence has been changed. See the Reference Manual entry for complete details.

tgetl\$

'Tgetl\$' has been completely rewritten. See the Reference Manual entry for complete details.

tputl\$

'Tputl\$' has been completely rewritten. See the Reference Manual entry for complete details.

Enhanced Routines

The routines listed in this section have additional functionality in the Version 7 release, but remain compatible with their Version 6 counterparts.

date

A new key has been added to 'date' to retrieve the current date in the format "Monday, January 31, 1980".

getarg

It is now possible to retrieve the command name as argument 0.

open

Files can be opened to the terminal, line printer spooler, and other devices.

print

A number of new format codes and options have been added.

remark

'Remark' now also accepts an unpacked string for an argument.

Version 7 Conversion Guide

remove

'Remove' is now capable of removing segment directories.

Unchanged Routines

No user-visible changes have been made to the routines listed in this section.

call\$\$	cant	chkinp
close	create	ctoi
ctol	ctor	delete
dsdbiu	dsdump	dsfree
dsget	dsinit	enter
equal	exec	execn
fcopy	follow	getch
getkwd	getlin	index
init	itoc	itoc0
itoc8	length	lookup
ltoc	mapdn	mapstr
mapsu	mapup	missin
mktabl	mktemp	putch
putdec	putlin	rewind
rmtabl	rmtemp	rtoc
scopy	st\$lu	substr
swt	t\$clup	t\$entr
t\$exit	t\$time	t\$trac
trunc	type	wind

New Routines

The routines listed in this section are new for the Version 7 release.

atoc

Convert address to character string.

c\$end

Run-time support routine for Ratfor statement count profile.

c\$incr

Run-time support routine for Ratfor statement count profile.

cof\$

Close all files opened since last call to 'iofl\$'.

cpfil\$

Copy an open file.

cpseg\$
Copy an open segment directory.

ctoa
Convert from character string to address.

ctoc
Copy character string to character string, paying attention to maximum length.

ctod
Convert from character to double precision floating point.

ctomn
Convert form non-printing character to ASCII mnemonic.

ctop
Convert from character string to packed string.

ctov
Convert from character string to PL/I varying character string.

decode
Convert from character string to various other formats.

delarg
Delete a command line argument.

dmark\$
Return the position of a disk file.

dopen\$
Open a disk file.

dread\$
Read words from a disk file.

dseek\$
Position a disk file.

dtoc
Convert double precision floating point to character string.

dwrit\$
Write words to a disk file.

encode
Convert from various data formats to a character string.

expand
Expand templates in a string.

filcpy
Copy a file or a segment directory.

filtst
Test a file for various conditions.

findf\$
Test file existence.

finfo\$
Get information on a file.

flush\$
Flush a file's Subsystem buffer.

gkdir\$
Get the path name of the current directory.

getvbn
Get the name of a file in a user's variables directory.

gfnarg
Parse file name arguments on the command line.

gklarg
Parse key-letter arguments on the command line.

gvlarg
Obtain the next argument not beginning with a hyphen.

icomn\$
Initialize the Subsystem common blocks.

inloc\$
Initialize a two word array with a long integer.

iofl\$
Mark all currently open files.

ioinit
Initialize the Subsystem i/o routines.

jdate
Compute a Julian date.

lopen\$
Open a line printer spooler file.

lutemp
Obtain value for a template.

mapfd
Obtain the Primos file unit number of a disk file.

Version 7 Conversion Guide

markf
Obtain the current position of a file.

mkfd\$
Make an open Primos file unit into a Subsystem file.

mkpa\$
Convert a Primos tree name into a Subsystem path name.

mktr\$
Convert a Subsystem path name into a Primos tree name.

mntoc
Convert ASCII mnemonic to a character.

move\$
Move a block of words quickly.

parscl
Parse the command line.

parsdt
Parse a character string containing a date.

parstm
Parse a character string containing a time.

prot\$
Set protection attributes on a file.

ptoc
Convert a packed string into a character string.

readf
Read raw words from a file.

rmfil\$
Remove a file.

rmseg\$
Remove a segment directory.

sctabl
Return all entries in a symbol table one at a time.

sdrop
Drop characters from a character string.

seekf
Position a file.

seterr
Set the program error return code.

stake
Take characters from a character string.

strbsr
Search a string table using a binary search.

strcmp
Compare two strings.

strim
Drop trailing blanks from a string.

strlsr
Search a string table using a linear search.

tmark\$
Return the position of a terminal file.

tread\$
Read words from a terminal file.

tscan\$
Traverse a subtree of the file system.

tseek\$
Set the position of a terminal file.

twrit\$
Write words to a terminal file.

upkfn\$
Unpack a Primos file name.

vfyusr
Verify a Subsystem user name.

vtoc
Convert a PL/I varying string to an EOS-terminated string.

wkday
Calculate the day of the week of a given date.

writef
Write raw words to a file.

Converting 'Rf' Programs to 'Rp'

The language that 'rp', the new Ratfor preprocessor, handles is not exactly a superset of the language that 'rf' accepts. You will probably find that existing 'rf' programs must be slightly changed if they are to be preprocessed with 'rp'. Nevertheless, there are several reasons for making this conversion:

Version 7 Conversion Guide

- . 'Rp's lexical conventions are more consistent.
- . A program can run up to 10% faster when preprocessed with 'rp'.
- . 'Rp' provides a richer set of declaration and control structures.
- . 'Rp' produces much more readable Fortran.
- . 'Rp' runs slightly faster and provides better diagnostics.

One additional impetus for converting 'rf' programs is that although 'rf' still exists in the Version 7 Subsystem, it is now considered a "locally supported" program.

The major points to consider when converting 'rf' programs are:

- . Character constants are represented differently ('a'c instead of LETA, ';'c instead of SEMICOL, etc.) The old format character constants must be replaced throughout the program. This applies to printable characters only; the non-printing characters (such as TAB and NEWLINE) still have the same names.
- . The keywords **andif** and **orif** are no longer recognized. They must be replaced with the '&&' and '||' operators. When removing these old forms, be careful not to change the order of evaluation by removing sets of parentheses. 'Rp' also allows nesting of '&&' and '||'; but it **does not** allow them to be nested within the scope of the '&' and '|' operators. Thus,

```
(a == b || a == c) && (d == e || d == f)
```

is allowed, but

```
(a == b || a == c) & (d == e || d == f)
```

is not. Nor is the use of expressions containing '&&' and '||' allowed on the right-hand side of assignments.

- . The **case** statement, although currently accepted by 'rp', is not officially supported. All occurrences of **case** statements should be replaced with **select** statements. Keep in mind that a chain of **if ... else if ... else** statements can be replaced by a **select** statement with a significant increase in performance.
- . Ratfor and Fortran keywords are now really reserved. Their use as identifiers can cause syntax errors or misordered code.
- . Alphabetic case is significant in 'rp' identifiers.

Version 7 Conversion Guide

- . Compound statements may be included in the <initialize> and <reinitialize> clauses of **for** statement.
- . 'Rp' automatically includes the standard Subsystem definitions by default; the "include '/syscom/defi'" statement should be removed from existing programs. Note that all Subsystem include files now reside in the directory "=incl=" (nominally "//extra/incl"), not in "//syscom"; if you have **include** statements that reference any of these files, you must change them.
- . 'Rp' automatically includes a "call init" in every main program; existing calls to 'init' should be removed.
- . Subroutines and functions, especially when used as program structuring devices, can often be replaced by internal procedures for significant savings in code space and execution time.