

Software Tools Subsystem
Version 7.1 to Version 8 Conversion Guide

Terrell L. Countryman
Jeanette T. Myers
Peter N. Wan

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

April, 1982

TABLE OF CONTENTS

Introduction	1
Global Changes	1
Terminal Type Handling	1
Templates	1
Speed	1
Memory Segments	2
Process Ids to Three Digits	2
Cldata Template	2
Exception Handling	2
DBG Support	3
New Shell Control Variables	3
Deleted Macro Definition	3
Change in Value of EOS	3
New Reference Manual Sections	3
New Subsystem Libraries	4
New Subsystem Templates	5
Status of Version 7.1 Commands	5
Obsolete Commands	5
Superseded Commands	5
Modified Commands	6
Enhanced Commands	8
Unchanged Commands	13
New Commands	14
Status of Version 7.1 Subroutines	17
Obsolete Routines	17
Superseded Routines	18
Modified Routines	18
Enhanced Routines	19
Unchanged Routines	21
New Routines	22

Introduction

Version 8 of the Software Tools Subsystem differs from Version 7.1 in a number of ways, most of which will not impact the average user. Most changes are extensions or internal performance improvements, and affect one or two commands rather than the entire Subsystem. This conversion guide is divided into three sections: **Global Changes** discusses the alterations that affect large portions of the user interface; **Status of V7.1 Commands** and **Status of V7.1 Subroutines** describe additions, deletions, and modifications made to individual commands and subroutines.

Global Changes

Terminal Type Handling

Various programs and library routines now support the tailoring of output for specific terminals. This entailed changes in the library, the macro definitions file, the Subsystem common blocks, 'swt', 'se', 'term', and 'term_type'. You will be affected by these changes if you have added terminal types to 'se' or if you have used the preliminary version of the Virtual Terminal Handler (VTH) library on the Version 7.1 release. Since a prompt for terminal type may now occur upon Subsystem entry, you may have to add terminal types to the "=ttypes=" file or educate your users about terminal types.

If you want to extend the terminal type knowledge of the Subsystem, you must add new terminal types and information concerning them to the file "=ttypes=", add a new initialization file in the directory "=vth=" for each new terminal type, and modify and recompile 'se' with new code to handle the new terminal.

Templates

Templates are no longer mapped to a single case. Unless you have some single-case terminals or regularly program in upper-case only, this change is unlikely to affect you.

If a template must contain imbedded equals signs, use two consecutive equals signs to pass one through the template expander; in an earlier version of the template processor, the "at sign" was used to "escape" the equals sign.

Speed

Version 8 of the Subsystem uses considerably less CPU time for I/O than Version 7.1. Unfortunately, if you are moving up to

Version 8 Conversion Guide

Revision 18 of Primos, you may not notice much difference; Revision 18 can be that much slower than Revision 17.

Memory Segments

Prime has now used all available private memory segments. At Version 8 of the Subsystem, it has become necessary to increase the NUSEG parameter in the Primos configuration file to at least 42 (octal), to provide private segments for the Subsystem that do not conflict with standard Prime programs. This implies that programs using secret knowledge of the Subsystem's common blocks must be relinked. It also implies that you cannot bring up Version 8 without rebooting your system.

Listed below are the segments required for the Subsystem:

	<u>Version 8</u>	<u>Version 7.1</u>
SWT Shell and template area	2030	2030
SWT Screen Editor	2031	2031
SWT Library	2035	2035
SWT Common	4040	4036
SWT Stack	4041	4035

Process Ids to Three Digits

To accomodate the increase in the number of processes in Primos 18, process ids will be three digits instead of two. The "=termlist=" file has changed slightly in format to accomodate the increased id lengths and AMLC numbers.

Cldata Template

The location of "cldata", a Primos command interpreter data structure referenced by the Subsystem shell, is now a template. The Version 8 release has "cldata" defined to be segment 6002, word 6, which applies to you if you are running Primos 18.3 or above. If you are running Primos 18.2 or lower, there is a commented template in the "Configuration Options" section of "=template=" that you need to use.

Exception Handling

The new version of the shell now allows you to intercept exceptional conditions, such as pointer faults, arithmetic value errors, interrupts, etc. Quits (via control-p or "break") abort the current program and return to command level in the shell, rather than leaving the you stranded in Primos, as was the case in previous versions of the shell. If you have a shell variable named "_quit_action" (the value is not important), then when the quit occurs, the shell will prompt you as to whether to abort the

Version 8 Conversion Guide

current program, continue, or call Primos. If Primos is called, the current program may be continued by typing START, or the Subsystem may be re-entered by typing REN.

DBG Support

There is now some support for invoking the symbolic debugger from the Subsystem. Please see the Reference Manual entry on 'dbg' for more information.

New Shell Control Variables

The variable "_kill_resp" can be used to set the character string that will be printed on your screen whenever you type a "kill" (the default is the string "\\"); "_prt_form" can be used to specify your usual printing form (the default is to use the default installation printer form); "_prt_dest" can be used to specify your favorite line printer (the default is to use the default installation printer); and, "_quit_action" can be used to take advantage of the new quit handling capabilities of the shell (by default, this variable is not declared and you are automatically returned to the command level of the shell, with no choice of error handling for quits).

Note that when a variable is set using either 'declare' or 'set', the value does not take effect until you exit the Subsystem and re-enter.

Deleted Macro Definition

One macro definition has been removed from the standard macros file. ESCCHAR is no longer defined; use ESCAPE instead.

Change in Value of EOS

After Version 8 of the Subsystem, the value of the end-of-string character (EOS) will be changed. The current value of EOS, as defined in the Subsystem definition file, is -2. It will be changed to the value 0 to maintain consistency with the way the C library handles the end-of-string. If you have any programs which depend on the value of EOS being of a certain magnitude (i.e., being negative), you should recode them to avoid depending on that assumption. This change will require the recompilation of all local Subsystem programs.

New Reference Manual Sections

Two new sections have been created to contain low level commands (Section 5) and low level subroutines (Section 6). You should not invoke these commands and routines under normal circumstances; they are usually support routines for user-callable

Version 8 Conversion Guide

Subsystem commands and routines.

The following commands were moved from sections 1 and 3 to section 5:

bmerge	bnames	bs
bsl	bugfm	bugn
guess	mkcl	

The following routines were moved from sections 2 and 4 to section 6:

at\$	bponu\$	c\$end
c\$incr	c\$init	call\$\$
chunk\$	cof\$	cpfil\$
cpseg\$	dgetl\$	dmark\$
dmpcm\$	dmpfd\$	dopen\$
dputl\$	dread\$	dsdbiu
dseek\$	dwrit\$	findf\$
finfo\$	first\$	flush\$
gkdir\$	gcifu\$	getfd\$
gfnam\$	icomn\$	iofl\$
ioinit	ldseg\$	ldtmp\$
lopen\$	lutemp	mkdir\$
mkfd\$	mkpa\$	mktr\$
reonu\$	rmfil\$	rmseg\$
rtn\$\$	sprot\$	st\$lu
t\$clup	t\$entr	t\$exit
t\$time	t\$trac	tcook\$
tgetl\$	tmark\$	tputl\$
tread\$	tscan\$	tseek\$
ttyp\$f	ttyp\$l	ttyp\$q
ttyp\$r	ttyp\$v	twrit\$
upkfn\$	vt\$alc	vt\$c1r
vt\$db	vt\$db1	vt\$db2
vt\$db3	vt\$def	vt\$del
vt\$dsw	vt\$serr	vt\$get
vt\$gsq	vt\$idf	vt\$ier
vt\$ndf	vt\$out	vt\$pos
vt\$put	vt\$rdf	zmem\$

New Subsystem Libraries

Vthlib has been totally rewritten, and is now supported as part of the Subsystem. It is much faster than the earlier version, and offers more features. In general, the rewritten routines are not compatible with those of the earlier release of VTH. Programs which used the earlier version of VTH will have to have the VTH calls recoded to use the new routine names (and new calling formats).

New Subsystem Templates

The templates "`=newcmdnc0=`" and "`=newsystem=`" specify where newly compiled Subsystem files that belong in "`cmdnc0`" and "`system`" are placed during a recompilation of the Subsystem. The "`=ttypes=`" file contains a list of terminals supported by your Subsystem and their characteristics. The "`=cldata=`" template, mentioned above, indicates where the Primos "`cldata`" data structure is located. The "`=sysname=`" template is used to indicate the Primenet node name, if the system is a network system.

Obsolete templates have been removed from the Version 8 release template file.

Status of Version 7.1 Commands

This section summarizes the user-visible changes that have been made to Subsystem commands for Version 8. 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 8 release.

Obsolete Commands

The commands in this subsection were part of the Version 7.1 Subsystem, but are not included in the Version 8 release. Most of them were used only by certain shell programs and have outlived their usefulness. In other cases, the commands were relics of past Subsystems, and either were no longer useful, or no longer worked.

No commands are obsoleted at Version 8.

Superseded Commands

The commands in this subsection are not part of the Version 8 Subsystem; their functionality has been subsumed by other commands. Each entry describes the command and options you can use to get the same results.

ar

Has been completely rewritten. The old '`ar`' is now '`old_ar`'. All existing archives should be processed with '`old_ar`', and then reprocessed with '`ar`', since support of '`old_ar`' will be dropped in a future release.

dumpls

Use the new command '`dump`' instead. The command line would be "`dump ls`".

dumpsv

Use the new command 'dump' instead. The command line would be "dump sv".

Modified Commands

The commands listed in this subsection have been modified for the Version 8 release and are no longer completely compatible with their Version 7.1 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.

dprint

The argument syntax has been changed slightly. The length option is now "-l <length>", rather than "-<length>".

'Dprint' can now handle the generation of multiple copies.

mon

Accepts four commands while running; three are used to reformat the screen and the fourth command redraws the screen.

Now uses VTH to do output, so it will work on terminals besides Beehives (any terminal supported by VTH).

pg

The control-c response causes 'pg' to ignore any remaining file names that were command line arguments and exit to the command interpreter.

Default prompt now shows the file being displayed as well as the page number.

The "-s" option has been added; it allows you to specify the screen size as an argument. The old syntax of "-<screen size>" was ambiguous if the <screen size> was 1, 2, or 3 (it was too close to the Subsystem convention of referencing the standard input/output ports with a "-<port number>").

rttime

By default, the output of the command being measured is diverted to /dev/tty; you can specify diversion to /dev/null if no output is desired.

se

Your terminal type can now be obtained by a call to the Subsystem. (This usually eliminates the need to know terminal type mnemonics, or at least the need to retype them as long as the Subsystem knows your terminal type --- if the Subsystem does not know your terminal type, it will prompt you for it when you invoke 'se').

Long input lines are now scrolled horizontally, allowing the cursor to remain visible at all times.

New options include "oh[<baud>]", to tell 'se' your baud rate; "olm[<column>]", to set the left margin of text to be displayed in the window (permitting handling of very wide files); and, "os[s | f | f77]", to set several programming language related parameters at once ("oss" for PMA, "osf" for FTN, "osf77" for F77).

Documented options include "ok", to indicate whether or not the current edit buffer has been saved; and, "om" to display a message (sent via 'to').

New commands include "e!", "w!" and "q!" which can be used to enter, write or quit without having 'se' tell you if you are about to destroy the contents of your edit buffer or the contents of an existing file. These replace the old forms "ea", "wa", and "qa", respectively.

New terminal types supported by 'se' are the hz1510, ts1, tvi, and z19.

'Se' now takes advantage of terminal hardware line insert/delete functions (if they are available for the given terminal) to speed up processing over slower transmission lines (i.e., dialups). The "-h" command line option and the "oh" 'se' command set/query the baud rate you are running at; 'se' decides how many nulls to put out, and whether to use the hardware line insert/delete functions or not, based on a combination of the baud rate and terminal type. If you want to add terminal types to 'se' for locally available terminals, the file "`=src=/spc/se.u/how_to_add_terminal_types`" explains how to do it (it also includes information on where to add the code necessary to handle the hardware line insert/delete functions). We at Georgia Tech would be interested to know about terminals that you add to both 'se' and the VTH package, so that we can include them in future releases of the Subsystem.

sh

The shell now handles breaks and control-p. If you declare the variable "_quit_action", you receive a prompt after a break and are allowed to continue the program, terminate the program and return to the Subsystem, or terminate the program and drop out to Primos (for debugging). If "_quit_action" is not declared, interrupted programs simply return control to the shell.

term

Support for the Subsystem terminal type management routines has been added.

New "-newline" and "-eof" options to specify newline and end-of-file characters; new "-vth" and "-se" options to specify whether or not the terminal type is supported by the Virtual Terminal Handler (VTH) and 'se'.

Removed "-tab" and "-enb".

term_type

Now uses the Subsystem terminal type management facilities. Options have been added to query the values of particular terminal attributes, as well.

x

'X' now calls the Primos command interpreter directly, via the Primos routine CP\$. (This reduces execution time and the amount of garbage displayed on your terminal.)

'X' can execute Primos commands interactively, regaining control when you generate an end-of-file.

Enhanced Commands

Commands in this subsection have been functionally enhanced for the Version 8 release, but remain compatible with their Version 7.1 counterparts.

change

Accepts a string as an argument that is to be searched for a pattern.

declare

Will not modify a shell variable that has already been declared at the current level.

declared

No longer decides whether a shell variable is "declared" if an illegal lexical level offset is supplied.

define

Finally does what the documentation says it will do.

diff

A "-b" option has been added to allow direct binary comparison of files. (Note that files that compare "equal" under the usual text comparison may not be equal under the binary comparison, because of blank compression.)

e

Takes 'se' options as arguments and uses new terminal type handling.

f77c

New "-w" option to generate floating round instructions which improves the accuracy of single precision floating point calculations.

fc

New "-w" option to generate floating round instructions which improves the accuracy of single precision floating point calculations.

file

No longer returns "-1" when it could not perform a test on a file; it returns a zero, which is in accordance to the documentation. An error message is written to ERRROUT for this case.

fmt

New ".eo" and ".oo" commands to specify different page offsets for even- and odd-numbered pages.

Documented the ".dv" (divert stream) command which, when used in conjunction with ".so", can produce an automatic table of contents.

guess

Requires an argument that is the command name to be used and optionally accepts a "tolerance" level.

guide

Version 8 Conversion Guide now available.

hd

New "-n" and "-u" options to display "normalized" (440 words/record) or "unnormalized" (1024 words per record) record counts, respectively.

help

Documented the "-u" option to print usage for command(s).

'Help' now uses 'page' for paging so you can use all the responses acceptable to 'page'.

ld

The templates "`=cm_loc=`" and "`=tp_loc=`" are checked to allow overriding the default segment numbers for Subsystem common blocks and template storage areas. This is useful if you are modifying the Subsystem and must run a production copy and a development copy side-by-side.

lf

New "`-q`" option to print nonowner password.

mail

Mail is saved in the file defined by the template "`=mailfile=`". The Subsystem default is "`=varsdir=/.mail`".

Checks for valid addressee name(s) before reading the letter to be sent.

mkclist

New "`-s`" argument to create the system defined command list ("`=ubin=`" is ignored).

moot

An "`index`" command has been added to summarize the entries that have been made in the current conference.

mt

'`Mt`' has been heavily modified to fix all known bugs.

New "`-v`" option to cause '`mt`' to print the number of blocks read or written.

plgc

New "`-w`" option to generate floating round instructions which improves the accuracy of single precision floating point calculations.

print

New "`-i`" option to indent listing, "`-j`" option to cause '`print`' to put a FORMFEED character at the end of a page instead of generating the number a blank lines required to get to the top of the next page, and a "`-l`" option to indicate the number of lines per page.

publish

A warning is now issued if you interrupt a '`publish`' (interrupting a '`publish`' has possible harmful side effects).

retract

A "`-q`" option has been added to allow retraction of a news article without printing a retraction notice.

rp

Declarations are now handled as a separate data stream.

You can now put statements in your Ratfor program that will not be touched by 'rp' and you can indicate if those statements are to be routed to the "declaration", "data", or "code" stream.

A "-g" option has been added to invoke an algorithm that tries to eliminate chains of GOTO statements. (When this was applied to the preprocessor itself, a 10% speedup resulted.)

New "-x" option and a accompanying translation table can be used for user definable character code translation.

A new "-y" option is available that causes 'rp' to not place "call init" and "call swt" statements in the Fortran code.

Several internal speed-up improvements have been made.

String tables now allow multiple slashes, causing marginal index entries to be duplicated. The maximum string table size has been increased.

The standard Ratfor macro definitions file now includes "SET_OF_GRAPHICS" and "SET_OF_SPECIAL_CHAR" for use in "when" clauses in Ratfor.

sema

Now handles named and negative (Primos system) semaphores.

stacc

A "null token" construct (epsilon) has been added to cause a match without scanning any input.

A "quick select" construct has been added to permit fast selection between a number of alternatives beginning with distinct terminal symbols.

General processing speed has been improved by eliminating the use of temporary files.

'Stacc' can now generate code in the C programming language, as well as Ratfor. The Reference Manual entry has been corrected so that it no longer indicates that SSPL, Pascal, and PLP are supported (they are not).

tail

Now correctly accepts a filename as an argument, even if it is the only argument (before, 'tail' would try to convert the lone file name as the number of lines

Version 8 Conversion Guide

parameter, get an error in the conversion, and end up reading the default number of lines from standard input).

tlit

Running time has been improved drastically.

For compatibility with 'take', 'drop', and other command line utility functions, 'tlit' can also accept strings as arguments to be transliterated.

to

The header line format has been changed, to provide more information.

translang

Added the "nor" operator, which was inadvertently left out of the lexical analyzer.

who

Changed to call the new Primos GMETR\$ to access the system data bases.

An "r" flag is appended onto the pid if the user is a remote user.

Unchanged Commands

This subsection lists the commands that have no user-visible changes made for Version 8.

alarm	arg	args
argsto	asll	as6800
as8080	banner	basys
batch	block	bs
bug	bye	cal
case	cat	cd
chat	chown	clear
clock	cmp	cn
col	common	como
copy	copyout	cp
crypt	ctime	cto
date	day	declared
del	detab	dmach
dnum	drop	echo
ek	else	entab
error	esac	eval
exit	f77cl	fcl
fdmp	fi	field
files	find	fixp
fmt	focld	forget
fos	fsize	goto
history	hp	if
imi	include	index
installation	intel	iota
join	kill	kwic
lam	ld	length
lex	lib	line
link	lk	locate
log	login_name	lps
macro	memo	mkclist
mkdir	mklib	mktree
mkusr	mot	mv
nargs	news	opt6800
opt8080	os	out
p4c	p4cl	passwd
pause	pc	pcl
ph	phist	phone
plgcl	pmac	pmacl
pr	print	profile
publish	pwd	pword
quote	rcl	rf
rfl	rmusr	rnd
rot	rsa	save
scroll	sep	set
sh	show	shtrace
size	slice	sort
source	sp	speling
sspl	ssr	st_profile
stats	stop	subscribe
substr	symbols	systat
take	tc	tee

template	then	time
ts	uniq	unoct
unrot	us	usage
vars	when	whereis
whois	xref	

New Commands

This subsection lists commands that are new for Version 8.

basename

Select various portions of a pathname.

bmerge

Merge object code files into one file for building a library.

bnames

Print entry point names in object files.

bs1

Identical to 'bs' except that it reduces search time, with the possible result of having a less intelligent guess.

bugfm

Format a bug report created with the 'bug' command.

bugn

Process the highest bug number.

cc

Compiles a C program with the Subsystem C compiler.

ccl

Compiles and loads a C program.

cdmlc

Compiles a Prime DBMS Cobol Data Manipulation Language program.

cdmlcl

Compiles and loads a DBMS Cobol Data Manipulation Language program.

cobc

Compiles a Cobol program.

cobcl

Compiles and loads a Cobol program.

csubc

Compiles a Primos DBMS Cobol subschema.

dbg
Interface to Primos source level debugger.

ddlc
Compiles a Prime DBMS schema.

des
An implementation of the National Bureau of Standards Data Encryption System.

dump
Debugging aid which dumps the shell's various internal data bases in a semi-readable format. This command supersedes the Version 7.1 commands 'dumpls' and 'dumpsv'.

fdmlc
Compiles a program written in the Prime DBMS Fortran Data Manipulation Language.

fdmlcl
Compiles and loads a Prime DBMS Fortran Data Manipulation Language program.

fsubc
Compiles a Primos DBMS Fortran subschema.

last
Allows you to look at the last few lines of a file.

Can also count the number of lines in a file very quickly.

mkcl
Make a command list in compressed binary format for use with the 'guess' command.

old_ar
Subsystem archiver from Version 7.1; included to allow you to retrieve your files and convert to the new archiver, 'ar'.

plpc
Compiles a PL/P program.

plpcl
Compiles and loads a PL/P program.

primos
Allows the use of the Primos command interpreter from the Subsystem. This command is somewhat different from the 'x' command, in that 'primos' causes a new level of the Primos command interpreter to be initiated.

radix

Convert numbers from one radix representation to another.

raid

Examine bug report submitted with 'bug' command (this is intended for the use of the Subsystem manager).

The bug reports can also be optionally be printed so that a hardcopy may be obtained.

rdcat

Relational database command which concatenates two identical relations.

rdextr

Relational database command which extracts relation data from a given relation.

rdjoin

Relational database command which joins two relations.

rdmake

Relational database command which constructs a relation from a data file.

rdprint

Relational database command to print a relation or a relation descriptor.

rdproj

Relational database command to project a relation.

rdsel

Relational database command to select tuples of a relation.

rdsort

Relational database command to sort a relation.

rduniq

Relational database command to remove duplicate tuples from a relation.

sol

Game of solitaire. A good demonstration of the new Virtual Terminal Handler (VTH) package.

spell

Faster than 'speling'.

Has a "verbose" output format to aid in locating misspelled words.

Is more intelligent about not reporting formatter commands as misspelled words.

tip

Check if terminal input is pending.

vpsd

Interface to invoke the Primos V-mode Symbolic Debugger on Subsystem programs.

Status of Version 7.1 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, unchanged routines and new routines.

Obsolete Routines

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

cmdf\$\$

Obsoleted because of a smarter shell.

rtr6800

The SSPL run-time support library for the M6800 microprocessor has been removed.

vt\$bc

Obsoleted by the new VTH routines.

vt\$cc

Obsoleted by the new VTH routines.

vt\$lld

Obsoleted by the new VTH routines.

vt\$l1

Obsoleted by the new VTH routines.

vt\$mv

Obsoleted by the new VTH routines.

vt\$pk

Obsoleted by the new VTH routines.

vt\$rc

Obsoleted by the new VTH routines.

vt\$upk

Obsoleted by the new VTH routines.

Version 8 Conversion Guide

vtceol
 Obsolete by the new VTH routines.

vtceos
 Obsolete by the new VTH routines.

vtenc
 Obsolete by the new VTH routines.

vtinl
 Obsolete by the new VTH routines.

vtins
 Obsolete by the new VTH routines.

vtmvdn
 Obsolete by the new VTH routines.

vtmvlf
 Obsolete by the new VTH routines.

vtmvr
 Obsolete by the new VTH routines.

vtmvup
 Obsolete by the new VTH routines.

vtpos
 Obsolete by the new VTH routines.

Superseded Routines

The following routines have been subsumed by other more powerful routines. Each entry names the Version 8 routine that performs the same function.

inloc\$
 Use 'decode'.

itoc0
 Use 'gitoc' or 'encode'.

itoc8
 Use 'gitoc' or 'encode'.

prot\$
 Use 'sprot\$'. The name was changed to avoid a conflict with the Primos routine of the same name.

Modified Routines

The routines listed in this subsection have been modified so that they are no longer compatible with their Version 7.1 counterparts. Although each entry briefly describes the changes

Version 8 Conversion Guide

that have been made, you should examine the corresponding Reference Manual entries to determine the exact behavior of the routines.

cof\$

Requires a "state" argument.

enter

'Enter' is now a function that returns a pointer to the dynamic storage area containing text of next symbol.

expand

If a template must contain uninterpreted "="s, do not precede it by a "@" but by another "=".

iofl\$

Requires a "state" argument.

sys\$\$

New argument to specify file unit from which the Primos command takes its input.

tscan\$

The 'path' argument is changed by this routine, but was not documented to say so. The documentation has been changed.

vt\$db

Has been rewritten for new VTH library.

vt\$del

Has been rewritten for new VTH library.

vt\$out

Has been rewritten for new VTH library.

vtclr

Has been rewritten for new VTH library.

vtinit

Has been rewritten for new VTH library.

vtputl

Has been rewritten for new VTH library.

vtterm

Has been rewritten for new VTH library.

vtupd

Has been rewritten for new VTH library.

Enhanced Routines

The routines listed in this subsection have additional functionality in the Version 8 release, but remain compatible with

their Version 7.1 counterparts.

call\$\$

Accepts an optional argument for the creation of an on-unit.

date

There are now system defines for the request keys (so that actual numbers for the type of request need not be supplied).

New values returned are minutes, seconds and milliseconds past midnight.

dopen\$

Now takes an argument to determine the number of retries on encountering a "file in use" situation.

getto

MFD passwords are now consistently assumed to be "XXXXXX". Mixed-case passwords have caused several problems; the real source of the difficulty is a change Prime made to TA\$ that renders it incompatible with earlier revisions of Primos.

lopen\$

Will put in the values for the user's shell variables "_prt_form" and "_prt_dest", if available, in the spooler entry.

open

Now takes a fourth argument to determine the number of retries on encountering a "file in use" situation.

Unchanged Routines

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

addset	amatch	atoc
c\$end	c\$incr	cant
catsub	chkarg	chkinp
close	cpfil\$	cpseg\$
create	ctoa	ctoc
ctod	ctoi	ctol
ctomn	ctop	ctor
ctov	decode	delarg
delete	dgetl\$	dmark\$
dodash	dputl\$	dread\$
dsdbiu	dsdump	dseek\$
dsfree	dsget	dsinit
dtoc	dwrit\$	edit
encode	enter	equal
error	esc	exec
execn	fcopy	filcpy
filset	filtst	findf\$
finfo\$	flush\$	follow
gcd	gcdir\$	gctoi
gctol	getarg	getccl
getch	getkwd	getlin
getto	getvdn	gfnarg
gitoc	gklarg	gltoc
gtemp	gvlarg	icomn\$
index	init	input
invmod	ioinit	isatty
itoc	jdate	ldseg\$
ldtmp\$	length	locate
lookup	lsallo	lscmpk
lscomp	lscopy	lscut
lsdel	lsdrop	lsdump
lsextr	lsfree	lsgetc
lsgetf	lsinit	lsins
lsjoin	lslen	lsmake
lspos	lsputc	lsputf
lssubs	lstake	ltoc
lutemp	makpat	maksub
mapdn	mapfd	mapstr
mapsu	mapup	markf
match	mkdir\$	mkfd\$
mkpa\$	mktabl	mktemp
mktr\$	mntoc	move\$
omatch	open	page
parsdt	parstm	patsiz
prime	print	ptoc
putch	putdec	putlin
putlit	pwrmod	readf
remark	remove	reonu\$
rewind	rmfil\$	rmseg\$
rmtabl	rmtemp	rtn\$\$
rtoc	scopy	sctabl

sdrop	seekf	set_copy
set_create	set_delete	set_element
set_equal	set_init	set_insert
set_intersect	set_remove	set_subset
set_subtract	set_union	seterr
st\$lu	stake	stclos
strbsr	strcmp	strim
strlsr	substr	swt
t\$clup	t\$entr	t\$exit
t\$time	t\$trac	tgetl\$
tmark\$	tputl\$	tquit\$
tread\$	trunc	tseek\$
twrit\$	type	upkfn\$
vfyusr	vtoc	wind
wkday	wrtf	zmem\$

New Routines

The routines listed in this section are new for the Version 8 release.

at\$

Subsystem interlude to Primos ATCH\$\$.

bponu\$

On-unit handler for "BAD_PASSWORD\$" condition.

c\$init

Initializes a Ratfor program in preparation for a statement count run.

chkstr

Check a string for printable characters.

dmpcm\$

Dump the contents of Subsystem common blocks in a printable format.

dmpfd\$

Dump information about a file descriptor.

file\$p

When called from a Pascal program, allows the program to use the I/O redirection and piping features of the Subsystem.

first\$

This routine sees if it has been called before; it is used by the Subsystem for initialization purposes.

gcifu\$

Get the file unit which is providing command input to the shell.

geta\$f

Allows Fortran programs access to the arguments from the Subsystem command line.

geta\$p

Allows Pascal programs to access the arguments from the Subsystem command line.

geta\$plg

Allows PL/I (subset G) programs to access the arguments from the Subsystem command line.

getfd\$

Look for an empty file descriptor.

getwrd

Retrieve the next word from a buffer.

gfnam\$

Get pathname of an open file.

gtattr

Returns a user's terminal attributes.

gttype

Returns the user's terminal type name.

init\$f

Allows the Fortran programmer to take advantage of Subsystem I/O (especially the standard input and output ports).

init\$p

Allows the Pascal programmer to take advantage of Subsystem I/O (especially the standard input and output ports).

init\$plg

Allows the PL/I (subset G) programmer to take advantage of Subsystem I/O (especially the standard input and output ports).

isadsk

Test to see if a file is a disk file.

sprot\$

Set the protection attributes for a file. This routine used to be named 'prot\$', but had to be renamed because of a name conflict with a Primos routine.

tcCook\$

Read a line from the terminal and handle operations of processing escape sequences, case and character set mapping, line kills, etc. ("cooking" the line).

ttyp\$f
Obtain the user's terminal type from the "=termlist=" file, if available.

ttyp\$l
List the available terminal types (as defined in the "=ttypes=" file).

ttyp\$q
Query for the terminal type from the user.

ttyp\$r
Return the user's terminal type from the Subsystem common area, if available.

ttyp\$v
Set the terminal's attributes in the Subsystem common areas.

vt\$alc
Allocate another VTH definition table for the keyboard macros.

vt\$cclr
Send the clear screen sequence.

vt\$db1
VTH debugging routine which prints mnemonics for the unprintable characters to be output.

vt\$db2
VTH debugging routine to dump the terminal input tables.

vt\$db3
VTH debugging routine to dump the macro definition table.

vt\$def
Allows the user to define a keyboard macro.

vt\$dsr
Perform a garbage collection on the VTH definition tables.

vt\$err
Print a VTH error message.

vt\$get
VTH input routine.

vt\$gsq
VTH input routine to receive a delimited sequence of characters.

vt\$idf
VTH input processor which invokes user-defined keyboard macros.

vt\$ier
Report an error in a VTH initialization file.

vt\$ndf
Remove a VTH macro definition.

vt\$pos
VTH positioning routine which moves the terminal cursor by means of absolute positioning sequences.

vt\$put
Copy a string into a VTH screen buffer.

vt\$rdf
Remove a VTH keyboard macro from the definition table.

vt\$enb
Enable input on a particular screen line.

vt\$getl
Retrieve a line from the VTH screen buffer.

vt\$info
Return information contained in the VTH common block.

vt\$move
Position the cursor to a given row and column.

vt\$msg
Display a message in the VTH status line.

vt\$opt
Set optional parameters for the VTH screen.

vt\$pad
Pad the rest of a field with blanks.

vt\$prt
Output formatted information to the screen buffers.

vt\$read
Read characters from the terminal into the screen buffers.

vt\$stop
Reset a terminal's attributes before terminating a program.