TOPS-20
Commands Reference Manual

Electronic Distribution

July 1990

This manual describes all operating system commands available to the nonprivileged user of TOPS-20. For easy reference, the command descriptions are arranged alphabetically.

This manual supersedes the manual of the same name and order number, AA-FP65B-TM.

Change bars in the margins indicate material that has been added or changed since the previous release of this manual.

OPERATING SYSTEM: TOPS-20 (KL Model B) Version 7.0

SOFTWARE: TOPS-20 EXEC Version 7.0

digital equipment corporation
maynard, massachusetts

First Printing, September 1985
Revised, June, 1988
| Software Update Tape 2, July 1990

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.

The software described in this document is furnished under a license and may only be used or copied in accordance with the terms of such license.

No responsibility is assumed for the use or reliability of software on equipment that is not supplied by Digital Equipment Corporation or its affiliated companies.

Copyright C 1985, 1988, 1990 Digital Equipment Corporation.

All Rights Reserved.
Printed in U.S.A.

The following are trademarks of Digital Equipment Corporation:

CI	DECtape	LA50	SITGO-10
DDCMP	DECUS	LN01	TOPS-10
DEC	DECwriter	LN03	TOPS-20
DECmail	DELNI	MASSBUS	TOPS-20AN
DECnet	DELUA	PDP	UNIBUS
DECnet-VAX	HSC	PDP-11/24	UETP
DECserver	HSC-50	PrintServer	VAX
DECserver 100	KA10	PrintServer 40	VAX/VMS
DECserver 200	KI	Q-bus	VT50
DECsystem-10	KL10	ReGIS	
DECSYSTEM-20	KS10	RSX	d i g i t a l

CONTENTS

FIGURES

PREFACE

The TOPS-20 Commands Reference Manual is an alphabetically-arranged description of all operating system commands (EXEC commands) available to the nonprivileged timesharing user of TOPS-20.

In addition, there are two appendixes for quick reference - a list of commands grouped by function, and an alphabetical summary of commands showing what variety of argument each uses and whether it calls a program or otherwise affects memory.

To use the TOPS-20 Commands Reference Manual properly, you should first read and understand the TOPS-20 User's Guide. The occasional summary of information you will find here cannot substitute for the more complete presentations offered in this manual.

In addition, you may need to reference the following manuals for more information related to operating system commands:

TOPS-20 Operator's Command Language Reference Manual

TOPS-20 Monitor Calls Reference Manual

TOPS-20 System Manager's Guide

TOPS-20 Operator's Guide

TOPS-20 User Utilities Guide

TOPS-20 Edit Reference Manual

EDT-20 Reference Manual

TOPS-20 LINK Reference Manual

TOPS-10/20 Batch Reference Manual

EXEC Message Manual

Conventions Used in This Manual

Underlined text	indicates what the user types in command examples.
^letter	means press the keys labeled CTRL and the specified letter simultaneously, for example ^C.
Ellipsis ...	means that items in a command line can be optionally repeated.
Carriage return	is implied in command examples.

CHAPTER 1

INTRODUCTION

COMMANDS AND ARGUMENTS

A complete TOPS-20 command consists of the command name and usually one or more arguments. In the most general sense, arguments are any combinations of letters, numerals, punctuation marks and other characters that you type after the command name itself to complete the meaning of the command. These arguments can be file specifications, switches, subcommands, and values for switches and subcommands, as well as words and numbers (the arguments to the SET and TERMINAL commands, for example). The following pages contain general information about each variety of argument.

File Specifications

Information and programs for TOPS-20 are usually stored in uniquely labeled files. Therefore, file specifications or "filespecs" are the most common variety of argument to a command. A complete file specification is of the form:

```
dev:<dir>name.typ.gen;att;...;att
```

where:

dev: is a device (usually a file structure)

<dir> is a directory name (enclosed in angle < > or square [ ] brackets)

name is a filename

.typ is a file type

.gen is a generation number

;att is a file attribute

You need file attributes in only a few situations and can usually let dev:, <dir>, and .gen take default values (that is, values defined by the state of your job - see the Special Features section below), so you can give most file specifications in the shortened form, name.typ, without being unclear. In a few cases, an entire file specification is assumed if you do not supply one when you give the command (for the CREATE and EDIT commands, for example, and for LOAD-class commands - COMPILE, LOAD, EXECUTE, and DEBUG).

Whenever you omit the dev: field of the filespec, the system assumes you mean your connected structure (DSK:). This is the public structure (usually named PS:), which all users must log in to, unless you connect to a directory on another structure by using the CONNECT command. Give the INFORMATION STRUCTURE command if you are unsure of the name for your connected structure.

Whenever you omit the <dir> field of the filespec, the system assumes you mean your connected directory. Unless you have given a CONNECT command, this is your "log-in" directory, the directory on the public structure that you must log in to and which usually has a name composed of your surname, or surname and initials, enclosed in angle or square brackets. You can change your connected directory by giving a CONNECT command. Use the DIRECTORY command to see the name of your connected directory.

When you omit the .gen field of the filespec the system usually assumes you mean the highest generation (largest generation number) of the file. (A few commands, for example, DELETE, RENAME, and DIRECTORY, act on all generations of a file unless you specify a particular generation.) When you create and edit text files, compile and debug programs, or do anything else to produce another generation of a file, the system automatically works with the highest existing generation and labels the changed file with the next higher generation number. Therefore when you omit the generation number in a filespec given as argument to a TOPS-20 command, you are assured of using the most recent version of the file. Although you can override this default action by specifying particular generations of input and output files, it is simplest and most straightforward to allow the defaults to prevail.

Specification of file attributes is optional. You can assign attributes in order to have a file automatically marked for deletion when you log out; to associate a file with a valid account; and so forth. Appendix C lists the available file attributes.

Pressing the ESC key instead of typing a filespec field will usually cause any default for the remaining fields to be printed on your terminal.

There are two characters (called "wildcard" characters) that you can include in any field of a filespec to include all files matching the rest of the filespec. An asterisk (*) fills in for zero or more

characters of a field, while a percent sign (%) fills in for a single character only. (However, only the complete field "DSK*:" is allowed as wildcard for the device field, and only the complete field "*" is allowed as wildcard for the generation field.) Therefore you could give the command DIRECTORY *.CBL to find out what source files written in COBOL are in your connected directory, or the command DELETE *.Q* to remove the EDIT program's backup files from your directory.

Switches

Switches are arguments used with LOAD-class (COMPILE, LOAD, EXECUTE, and DEBUG) and EDIT-class (CREATE and EDIT) commands, as well as with Queue-class commands - that is, those affecting entries in processing queues (CANCEL, DISMOUNT, MODIFY, MOUNT, PLOT, PRINT, PUNCH, and SUBMIT).

Switches can also be used with the following program-control commands: DDT, GET, MERGE, R, and RUN. The REWIND command also accepts a switch.

Switches allow you to quickly give many options chosen from a large list, and let you specify to which files they apply when you give more than one filespec in a single command.

Give switches on the same line as the command, typing a slash (/) before each switch. If your command requires more than one line, simply keep typing without giving a carriage return. The system will begin a new line automatically and will read your command as if you had typed it on a single line. Or you can end your command line at any point with a hyphen (-) and carriage return, and continue the command on the next line; the hyphen will not be considered part of your input.

Keep in mind the way each class of command considers switches. EDIT-class commands operate on only one file at a time, and the switches must be given before the input filespec - this is the simplest case.

Queue-class and LOAD-class commands treat switches according to their position in the command line. If you give them before any filespecs, they act as default switches for all filespecs in the command (they will be in effect unless you override them with later switches applying to individual files only). If you give them after the first filespec, they apply only to the preceding file. In addition, there are a few switches of a different sort for the PLOT, PRINT, and PUNCH commands - these apply to all files no matter where they appear in the command line. These are called job switches (because they affect the entire printing job) and are presented in a separate list in those command descriptions.

A switch is a default if the system assumes it in the absence of

others. For example, for LOAD-class commands, /FORTRAN is the default for all switches that specify which compiler to use (like /MACRO, /COBOL, and /PASCAL). The /NOCOMPILE switch is the default for /COMPILE, /NOCOMPILE, and /RELOCATABLE. Most defaults for these commands apply to only a pair of switches; however, the /BINARY, /NOSEARCH, and /NOOPTIMIZE switches are assumed, for example, unless you specify /NOBINARY, /SEARCH, and /OPTIMIZE.

Default switches for the other classes of command operate similarly. Some are in effect unless you specify their opposite; others are in effect with a default argument unless you specify another argument; still others must be specified to be in effect, but are supplied with a default argument. The list of switches presented with each command description distinguishes these cases. When you give two or more switches of the same kind (for example, /BINARY and /NOBINARY), the last switch given usually prevails.

Subcommands

Subcommands are a variety of arguments used chiefly with DIRECTORY-class (DIRECTORY, FDIRECTORY, TDIRECTORY, and VDIRECTORY) commands and with the BUILD command. If you want to give subcommands to any of the DIRECTORY-class commands, type a comma at the end of the command line just before pressing the carriage return. After the system prompts you with a double at sign (@@) you can give subcommands, one on each line. If you give no subcommands to the DIRECTORY command, the ALPHABETICAL and HEADING subcommands will be in effect, giving an alphabetical listing of files. For a few subcommands, default arguments will be in effect if you give the subcommands without supplying any. You can give subcommands in any order, requesting or declining special categories of information and specifying the format of its presentation. If you give mutually exclusive subcommands (ALPHABETICAL and CHRONOLOGICAL, for example) the last-given subcommand prevails. Note that the FDIRECTORY, TDIRECTORY, and VDIRECTORY commands are equivalent to the DIRECTORY command with certain subcommands automatically included, and can be further modified with other subcommands from DIRECTORY's list. To cancel a DIRECTORY-class command while giving subcommands, type a CTRL/C.

The BUILD command operates differently from DIRECTORY-class commands in putting you automatically into subcommand mode without your typing a final comma on the command line, and in offering a special subcommand to cancel the command while you are typing subcommands.

See the individual command descriptions for more detailed information about subcommands.

Other Arguments

Some command arguments are not introduced by special characters such as slashes or double at signs, but still are particular words, or "keywords," having a special meaning to the system. Give these on the same line as the command itself, leaving at least one space before and between arguments. Certain of these (for example, the LOGICAL-NAMES argument to the INFORMATION command) are actually composed of more than one word, joined by hyphens so that the system will not interpret them as more than one argument. If the complete command will not all fit onto one line, simply keep typing: the system automatically begins a new line when necessary, but interprets the typing as if it had all appeared on one line. Or you can end a command line at any point with a hyphen (-) and a carriage return, and continue the command on the next line; the hyphen will not be considered part of your input.

A few command arguments must be accompanied by special symbols to be interpreted correctly. Enclose directory names in angle brackets when using them with the CONNECT or names (for DEFINE), device names (used with ASSIGN), and structures and tape sets (used with MOUNT or DISMOUNT) all require a colon at the end. (But note that when supplying what look like structure and tape set names to the related CANCEL command, you are supplying jobnames, and must not include the colon.) If you punctuate a command argument incorrectly, the system will usually print a message reminding you of this.

SPECIAL FEATURES

Defaults

The concept of "defaults," or command arguments assumed when you do not specify a choice, is important for understanding TOPS-20. To speed processing of commands and to help inexperienced users, the system uses defaults when necessary for completing commands that you give. By taking advantage of this defaulting action, you can make the system work faster and more efficiently for you. This manual displays prominently the available choices of command arguments and the established defaults for these.

There are different kinds of default. When you give file specifications as arguments to, say, the PRINT command, the system assumes that you are referring to the highest (most recent) generation of these files in your connected structure and connected directory. By specifying a different structure, directory, or generation you can override this default, but only if you already have the right (established by user membership in a group, perhaps, or by a prior ACCESS command) to do so.

When you give the INFORMATION BATCH-REQUESTS command without further arguments, you are presented with a listing of all requests in the batch input queue. The /USER switch allows you to limit this display to the jobs of the user named. If you give the switch without supplying a user name, your own user name is used as default. (But the /USER switch to the related SUBMIT command has meaning for privileged users only, who can use it to run batch jobs under other user names; for non-privileged users this switch effectively defaults to your own user name.) Only a few command arguments behave like the /USER switch. The /TIME switch to the SUBMIT command is worth noting: by not giving the switch, you set a time limit of 5 minutes; by giving the switch without specifying a time limit, you are setting a 1-hour limit; and you can set any other time limit by supplying it as argument to the switch.

Note that none of the three switches to INFORMATION BATCH-REQUESTS and INFORMATION OUTPUT-REQUESTS (/ALL, /FAST, and /USER) is used as a default: each calls for a listing that differs in some respects from that yielded by the unmodified command. However, the INFORMATION LOGICAL-NAMES command, which allows keywords ALL, JOB, and SYSTEM, has JOB as default for these. The list of arguments associated with each command makes these choices clear.

To discover what default argument (if any) is established for a switch, subcommand, or other argument, press the ESC key instead of giving the argument: the default will be printed on your terminal.

"Recognition" input is another feature of the TOPS-20 operating system that speeds up your input, by finishing the typing of a command or command argument for you when you have given only part of it and then prompting your next response.

As soon as you have typed enough of a command or argument to distinguish it from others, press the ESC key: the system will finish the word, if possible, and prompt your next input with guidewords enclosed in parentheses. (Note that this manual does not show guidewords except in the Format section of each command description.) By pressing the ESC key without beginning an argument you make the system print the default argument, if there is one. If the system cannot help you, either because you have not typed enough characters to make your intentions clear or because there is no default, your terminal makes a warning noise - either a ringing bell or high-pitched beep.

Because the ESC key does not produce a printed character, you may be unable to remember exactly where you pressed it when later examining the output from a hard-copy terminal. If you want to avoid this possibility, use the TERMINAL NO RAISE command to make your terminal produce lowercase input, to distinguish it from the uppercase printing of the system.

For TOPS-20 Version 7, recognition has been expanded so that presssing

the ESC key also shows as much of a command keyword or file specification as possible. For file specifications, recognition is in effect only for the file name, extension, and version.

You cannot use recognition for device names or directory names (including file structures), jobnames (for example, with the MODIFY or CANCEL command) or logical names.

Abbreviation

By abbreviating commands and command arguments, you can further increase the speed with which you give instructions to TOPS-20. The smallest unique abbreviation for a command or argument will stand for the entire word; if there is a default choice for further arguments, the system will assume you want this too. (You can determine sufficient abbreviations by using the ESC key: any correctly recognized abbreviation will stand for the word.) For example, the abbreviated command FD stands for the FDIRECTORY command; I L stands for INFORMATION LOGICAL-NAMES JOB.

There are a few cases where non-unique abbreviations stand for a frequently used command. For example, DIS is the abbreviation for DISABLE even though two other commands begin with the letters DIS - DISCARD and DISMOUNT.

Special Abbreviation	Command
C	CONTINUE
D	DEPOSIT
DIS	DISABLE
E	EXAMINE
EX	EXECUTE
INFORMATION F	INFORMATION FILE-STATUS
LOG (When not logged in)	LOGIN
LOG (When logged in)	LOGOUT

You cannot use abbreviation (or recognition) in a few cases. The names of devices (including file structures), and jobnames (given, for example, as arguments to the MODIFY or CANCEL command) must be typed in full; and of course passwords cannot be abbreviated or recognized.

Question Marks

Whenever you type a question mark (?) instead of (or even in the middle of) a TOPS-20 command or command argument, the system responds with instructions or a list of possible completions. By using question marks and recognition when you are unsure of the proper command or argument, you can have the system help you during your terminal session.

These features, together with the HELP command, which provides information on various system programs, are valuable supplements to the written documentation for TOPS-20.

For TOPS-20 Version 7, new functionality has been added to make the question mark feature more useful. More possible completions and choices are now listed. In addition to possible command names listed in response to a question mark, the system also displays possible filenames from the system (SYS:) directory. For example:

```
@L ? Command, one of the following
   LOAD    LOGOUT
    or system program name
   LOGIN
   LOGOUT
```

Where the LOGIN and LOGOUT that appear after "or system program name" are files from the system directory that start with L.

The functionality of the question mark feature has been expanded for file specifications also. Question mark lists possible file names, extenstions (including nulls), and file version numbers. For example:

```
@DIRECTORY E? FILE NAME
   EXTRA
   EXTUSR
   EMACS
@DIRECTORY EMACS.? FILE NAME
   INIT
   VARS
@DIRECTORY EMACS.INIT.? FILE NAME
  1
  2
```

DATE-TIME ARGUMENTS

You can specify date and time arguments to many of the TOPS-20 commands. The following sections describe the formats for these arguments.

Date

The examples below show the various formats that are acceptable for the date argument:

```
Jun 30 1981          30 Jun 1981
May 1, 82            1/May/1982
January 000005 75    0005-January-000075
F/13-83              5/17/83
```

If the month and day are both numeric, the first number of the two, if less than 13, is considered to be the month. Otherwise, the second number is considered to be the month. For example:

```
2/15/83              is February 15th
15/2/83              is February 15th
```

You can abbreviate the month to as few characters as possible without causing it to be confused with another month. Thus,

```
O
Jun
Jul
Ja
```

are acceptable abbreviations for October, June, July, and January.

Many commands allow you to give the day of the week or "today" for the date.

If you specify the time along with the date, you must separate the two arguments by at least one space and/or no more than one tab.

For the time argument you can specify:

- time according to a 24-hour clock:

  ```
  /AFTER:17:00:00
  ```

- AM and PM:

  ```
  @SET ALERT 5:00PM
  ```

- the following time zones:

Arguments	Zone	Hour from Greenwich
GST, GDT, GMT	Greenwich	0
AST, ADT	Atlantic	4
EST, EDT	Eastern	5
CST, CDT	Central	6
MST, MDT	Mountain	7
PST, PDT	Pacific	8

YST, YDT	Yukon	9
HST, HDT	Hawaii/Alaska	10
BST, BDT	Bering	11
DAYLIGHT	Daylight time for your zone	
STANDARD	Standard time for your zone	

Examples

```
6:00PM-EDT     is 5:00PM EST
6:00PM-PST     is 9:00PM EST
6:00PM-GMT     is 1:00PM EST
```

Note that a hyphen (-) is required before the zone.

The basic time format is:

```
hh:mm:ss
```

where:

```
hh      is hours, must be less than 24, and is optional
mm      is minutes, must be less than 60, and is required
ss      is seconds, must be less than 60, and is optional
```

The colon between hours and minutes is optional.

Examples (based on a 24-hour clock):

```
3       is 00:03:00AM
125     is 1:25AM
14:30   is 2:30PM
25:33   is 00:25:33AM
```

Relative Date-Time Arguments

Many commands accept relative dates and times. You can specify that an event is to occur at a certain amount of time from the current time, from today, or from a certain day of the week. Likewise, you can specify relative times in the past.

Examples

```
@SET ALERT +30      sets an alert for 30 minutes from now

@DIRECTORY,         produces a listing of files that
@@BEFORE TODAY      were created before today's date
```

EXEC MESSAGES

The TOPS-20 Command Processor, or EXEC, displays messages in response to errors or other conditions that arise when using the TOPS-20 Command Language. There are three kinds of EXEC messages: error, warning, and information.

Error messages begin with a question mark (?). An error message indicates that a failure has occurred and execution of the command or program has stopped. If, for example, you type the wrong password in a LOGIN or CONNECT command, you receive the message **?Incorrect password**, and the command is not executed.

Warning messages begin with percent sign (%). A warning message also indicates that a failure has occurred, but execution of the command or program usually continues. For example, suppose you type the command DIRECTORY *.PAS, *.REL to list all the files with the extensions .PAS and .REL, and, your connected directory does not contain any .REL files. The EXEC lists all the files with the .PAS extensions and then prints the warning message, **%No such file type *.REL**.

The third type of message you can get from the EXEC (and sometimes from the system) is an information message. Information messages are enclosed in square brackets [ ] and inform you about the status of the system or the result of an EXEC command. For example, the message **[DECSYSTEM-20 continued]** indicates that a temporary pause in service has ended. The message **[n pages freed]** indicates that your EXPUNGE command freed n disk pages. Usually information messages require no response from you.

When you need more information than is provided by an EXEC message, see the EXEC Message Manual. This manual contains detailed descriptions of all EXEC messages, plus actions to take for correcting errors.

NEW AND CHANGED FEATURES

The TOPS-20 EXEC has been enhanced considerably for TOPS-20 Version 7. The following is a list of new commands, and new arguments, subcommands and switches to previously existing commands. Changes affecting the command are shown as bulleted items.

Changes to Commands and New Arguments, Subcommands, and Switches

| ARCHIVE
|
| NORETAIN
| VISIBLE
| BUILD

```
|            ABSOLUTE-INTERNET-SOCKETS
|            [NOT]EXPIRATION-OF-PASSWORD
|            EXPIRE
|            INTERNET-ACCESS
|            INTERNET-WIZARD
|            NOT SECURE
|            SECURE
|
|       INFORMATION DIRECTORY
|
|             o  Displays date and time of last interactive login
|
|             o  Displays date and time of last non-interactive login
|
|             o  Displays date and time password expires
|
|             o  Displays number of interactive login failures since last
|                login
|
|             o  Displays number of non-interactive login failures  since
|                last login
|
|             o  Indicates if directory is SECURE
|
|
|       INFORMATION INTERNET STATUS
|
|       INFORMATION SYSTEM-STATUS
|
|             o  Displays number of days for password to expire
|
|             o  Indicates whether the password dictionary is enabled
|
|       MODIFY
|
|            /REMOTE-PRINTER:
|
|       SET
|
|            DIRECTORY NO SECURE
|            DIRECTORY SECURE
|            FILE [NO] PERMANENT
|            FILE [NO] TEMPORARY
|            FILE [NO] SAVE-BY-BACKUP-SYSTEM
|            FILE [NO] SECURE
|            FILE [NO] UNDELETABLE
|            PASSWORD
|
|       SYSTAT
|
|            CONNECT-TIME
```

- Displays remote user name in the origin field

CHAPTER 2

COMMAND DESCRIPTION

2.1 ACCESS

Obtains ownership rights to a directory and the group rights of its user-group list.

Format

@ACCESS (TO DIRECTORY) **dev:<directory>**
Password:**password**

where:

dev:<directory>	is the directory that you want to access. **Default** dev: - your connected structure **Default** <directory> - the directory (on the specified structure) of the same name as your connected directory.
password	is the password of the directory (not requested for your log-in directory or a directory of the same name as your log-in directory on a domestic structure).

Characteristics

Capabilities

Your capabilities (such as WHEEL, OPERATOR, CONFIDENTIAL) are associated with your log-in user name only. If you give the ACCESS command for a directory whose owner has for example WHEEL capabilities, you do not gain these

capabilities.

Restrictions

One Directory Per Structure

You can access only one directory at a time on each mounted structure. Each ACCESS command ends any previous ACCESS command for that structure (including the implicit access obtained by the LOGIN command). If you access another directory on the public structure you give up your own group rights on the public structure. These are restored when you give an ACCESS command for your log-in directory.

Not For Files-only Directories

Because a files-only directory does not have an owner or user group rights, you cannot give an ACCESS command for it. Use CONNECT instead.

Related Commands

CONNECT	for making a directory your connected directory
END-ACCESS	for surrendering rights to an accessed directory
MOUNT STRUCTURE	for making a structure available for access, and ensuring the continued availability of an accessed structure

Examples

1. Access another user's directory.

```
@ACCESS <HOLLAND>
Password:___
```

2. Access another user's directory so you can copy a file from it to your connected directory.

```
@ACCESS <HOLLAND>
Password:___
@COPY <HOLLAND>MAX.MEM HOLMAX.MEM
 <HOLLAND>MAX.MEM.1 => HOLMAX.MEM.1 [OK]
@END-ACCESS <HOLLAND>
```

3. Access the login directory of a user whose group rights you want to borrow.

```
@COPY <MANUALS>CHKCRF.MAC CHKCRF.MAC
?Directory access privileges required - "<MANUALS>CHKCRF.MAC"
@ACCESS <HOLLAND>
Password:___
@COPY <MANUALS>CHKCRF.MAC CHKCRF.MAC
 <MANUALS>CHKCRF.MAC.4 => CHKCRF.MAC.1 [OK]
@END-ACCESS <HOLLAND>
```

4. Access the directory of a user on another structure. Then examine the directory and copy a file from it.

```
@MOUNT STRUCTURE SNARK:
Structure SNARK: Mounted
@ACCESS SNARK:<HOLLAND>
Password:___
@DIRECTORY SNARK:<HOLLAND>

   SNARK:<HOLLAND>
 ACCT.MEM.1
 ACTGEN..1
 COMP.FOR.1
 COMPUT.CBL.1
   .REL.1
 DIFFER.FOR.1
   .QOR.1
 MAIL.TXT.2
 OVERVIEW.MEM.1

 Total of 9 files
@COPY SNARK:<HOLLAND>COMP.FOR
 SNARK:<HOLLAND>COMP.FOR.1 => COMP.FOR.1 [OK]
@END-ACCESS SNARK:<HOLLAND>
@DISMOUNT STRUCTURE SNARK:
Structure SNARK: Dismounted
```

2.2 ADVISE

Links your terminal with another user's terminal so that you can give commands to that user's job. The advisee can still give commands to his job.

Format

@**ADVISE** (USER) **argument**

where:

argument is either a user name or terminal line number.

Characteristics

Input to Other Job

For as long as the ADVISE command is in effect, the commands you give affect the advisee's job instead of your own.

Ending Advice

To end an advising link that you have formed between terminals, you must type CTRL/E. This CTRL/E is not echoed on either terminal.

Refused Advice

Ordinarily, you cannot advise a job unless its terminal is set to receive advice. However, if you have WHEEL or OPERATOR capabilities enabled, you can ADVISE any job, providing the user has not given the TERMINAL INHIBIT command.

Special ADVISE Commands

Once you are advising another job, you can give special commands to send comments or control characters, or to relink to a terminal that has broken links with yours. These commands are:

CTRL/E	End an ADVISE link
CTRL/~ +	Restart an ADVISE link after a BREAK command is typed at the advised terminal
CTRL/~ (	Start a comment (or use the REMARK command)

CTRL/~)	End a comment (or use the REMARK command)
CTRL/~ ?	Display the list of ADVISE control characters
CTRL/~<CHAR>	Send CTRL/CHAR

Note that the tilde character (~) can have different meanings with various terminal models. Consult your terminal's manual for the character equivalent to 36 octal in ASCII code.

Special Cases

Advisee Has More Than One Job

If more than one job is logged in under the user name you specify, the system gives you a list of that user's terminal numbers and associated programs to choose from, then prints TTY: to prompt your response. Type your choice of terminal number after the prompt.

Advising a Pseudoterminal (PTY:)

If you try to advise a PTY: the system informs you of this and asks you to confirm with a carriage return.

Restrictions

Compatible Terminals

Unless the terminals involved in an advising link have compatible characteristics (such as terminal width, ability to handle tabs and lowercase letters), some information can be lost or overprinted. To avoid this problem, the user of the faster or more capable terminal should adjust his terminal's characteristics, if possible before the ADVISE command is given.

Detached Jobs

You cannot advise detached jobs.

Warning

Talking Between a VT100 and a VT52

If links between VT100 and VT52 terminals are established

using an ADVISE (or TALK) command, the VT52 may function improperly during or after the linked interval (for example, by requiring frequent CTRL/Q commands to print multiple lines of output). Turning the terminal off and then on again (after the linked interval) will correct this problem.

Effect on Terminal

The ADVISE command leaves your terminal at the advisee's terminal's command level, controlling his job.

Related Commands

RECEIVE ADVICE	for allowing other users to advise you
REFUSE ADVICE	for preventing other users from advising you
REMARK	for sending comments only
TALK	for linking terminals so that your commands affect only your own job
TERMINAL INHIBIT	for refusing all types of terminal communication including advice, links, system messages, user messages, and notices of new mail.

Examples

1. Advise a user, then immediately type CTRL/E to end advice.

```
@ADVISE D.CROWLEY
 Escape character is <CTRL>E, type <CTRL>^? for help
 D.CROWLEY JOB 51 EXEC

LINK FROM LATTA, TTY 226
 [Advising]

^E  !Not displayed on terminal
 [Advice terminated]
```

2. Advise a user's job and access a directory for him.

```
@ADVISE BONSAVAGE
 Escape character is <CTRL>E, type <CTRL>^? for help
 BONSAVAGE JOB 48 EXEC

LINK FROM LATTA, TTY 226
```

```
 [Advising]
!I'LL ACCESS THE DIRECTORY FOR YOU, THEN YOU CAN USE IT.
@ACCESS <SARTINI>
Password:
@!OKAY, NOW YOU CAN USE IT.
@!Thank you.
^E
 [Advice terminated]
```

3. Advise another user, demonstrating how to use the FILCOM program.

```
@ADVISE D.CROWLEY
 Escape character is <CTRL>E, type <CTRL>^? for help
 D.CROWLEY JOB 51 EXEC

LINK FROM LATTA, TTY 226
 [Advising]
!HERE'S HOW TO COMPARE FILES USING THE FILCOM PROGRAM.
@FILCOM

*=VERCBL.TXT, BAKVER.TXT/A

No differences encountered

*^C
@!SEE?  THE SWITCH AT THE END (/A) MEANS TO COMPARE THEM IN ASCII
@!MODE. DON'T FORGET THE CTRL/C WHEN YOU'RE DONE.
@!THANKS.
^E
 [Advice terminated]
```

4. Advise a user who is logged in at more than one terminal. Choose one of them.

```
@ADVISE LATTA
 TTY25, EXEC
 TTY41, EXEC
 TTY27, EXEC
TTY: 27
 Escape character is <CTRL>E, type <CTRL>^? for help
 LATTA Job 22 EXEC

LINK FROM D.CROWLEY, TTY 225
 [Advising]
     .
     .
     .
^E
 [Advice terminated]
```

2.3 APPEND

Adds the contents of one or more source files to the end of a new or existing destination file on disk, leaving the original source files unchanged.

Format

@APPEND (SOURCE FILE) **source filespec(s)** (TO) **destination filespec, @@subcommand**

where:

source filespec(s)	is a single file specification, or a series of them separated by commas.
destination filespec	is the specification of the destination file on disk; this can be a new file.
@@subcommand	means that after a final comma you can type an optional keyword, modifying the mode or format of information transfer.

APPEND Subcommands
(when used with the paper tape reader - PTR:)

ASCII	specifies that the files being appended are written in ASCII mode, with 36-bit words each consisting of five 7-bit bytes and a parity bit; the parity bit means that the eighth hole of the paper tape is never punched.
BINARY	specifies that the files being appended are composed of 36-bit words, each consisting of six 6-bit bytes, with the seventh hole of the paper tape set always to 0 and the eighth hole set always to 1; causes a checksum calculation.
BYTE n	specifies that the byte size of the destination file is to be n (any decimal number). If you do not give the BYTE subcommand, the destination file will have the same byte size as the source file.

IMAGE	specifies that the files being appended are composed of 36-bit words, each consisting of one 8-bit byte; the 28 most significant bits are lost on output.
IMAGE BINARY	same as BINARY, but lacking the checksum.

APPEND Subcommands
(when used with devices other than the paper tape reader)

ASCII	specifies that the files being appended are written in ASCII mode, with 36-bit words each consisting of five 7-bit bytes and a parity bit; the parity bit means that the least significant bit is set to 0 on input and is lost on output.
BINARY	calls for a direct transfer of data in 36-bit bytes.
BYTE n	specifies that the byte size of the destination file is to be n (any decimal number). If you do not give the BYTE subcommand, the destination file will have the same byte size as the source file.
IMAGE	same as BINARY.
IMAGE BINARY	same as BINARY.

Output

As each file is appended, the system prints its specification and the word [OK]. Also, if recognition is used on the destination file specification, the system prints its status (Old generation, New generation, New file, or Superseding, for disk files; or OK, if the files are appended to a non-disk device).

Characteristics

Files Appended in Order Specified

The APPEND command attaches source files to the destination file in the order you specify them; the contents of the last

specified source will appear at the end of the destination file when APPEND is finished.

Subcommands Optional

For most purposes you do not need to use subcommands when transferring information with the APPEND command. These subcommands, specifying the format of the appended files, are required only when using certain devices (for example, devices of the form MTn: (tape drives) using labeled tapes, or PTR: (paper tape reader)) or under particular conditions (for example, when transferring files over network facilities). If you are appending information from disk files or from your terminal and you do not use any subcommands, the data will be appended as written, whether in a standard format (usually ASCII or binary) or not.

Special Cases

Wildcard Characters

Wildcard characters (* and %) can be used in source file specifications only. The files are then appended in alphabetical order.

Appending Information from your Terminal

If you type TTY: in place of source file specifications, the system appends any characters you then type (after completing the command itself), until you give a CTRL/Z to return your terminal to TOPS-20 command level. CTRL/U, CTRL/R, CTRL/W, and the Delete key can be used to edit the current line of terminal input.

Restrictions

Source Files With Differing Formats

You can use the APPEND command to transfer data from a magnetic tape, terminal, card reader, paper tape reader, or other device to disk files, but if source files written in differing formats are specified within the same command, some data can be lost in the transfer.

Mixing Sequenced and Unsequenced Files

Source files created by the EDIT program should not contain sequence numbers when they are appended. Mixing files that contain sequence numbers with files that do not will cause

EDIT to function improperly if used on the resulting file.

Appending to Archived Files

You can append the contents of an archived file to another file, by specifying it as the first (or source) argument of an APPEND command. You can then edit the resulting file, because it does not gain archive status although part of its contents are the same as those of the archived file; the archived file remains unchanged. However, you cannot give the specification of an archived file as the second (or destination) argument of an APPEND command, as this would change the file's contents.

Related Commands

COPY for making copies of files

Examples

1. Use the APPEND command to join two files.

```
@APPEND FORT.FOR FIL.FOR
 FORT.FOR.8 [OK]
```

2. Append two files to the end of a third file.

```
@APPEND FORT.FOR, GORT.FOR GIL.FOR
 FORT.FOR.8 [OK]
 GORT.FOR.6 [OK]
```

3. Access a directory and append a file from it to a file in your connected directory.

```
@ACCESS <MANUALS>
Password:___
@APPEND <MANUALS>REL3A.MEM REL3A.MEM
 <MANUALS>REL3A.MEM.4 [OK]
@END-ACCESS <MANUALS>
```

4. Use a wildcard character (%) to append several files to the end of another file.

```
@APPEND %ORT.FOR HIL.FOR
 FORT.FOR.8 [OK]
 GORT.FOR.6 [OK]
 HORT.FOR.3 [OK]
 MORT.FOR.2 [OK]
```

5. Use a wildcard character with the APPEND command to create a new file.

```
@APPEND *.TXT BACKUP.TXT
 MAIL.TXT.1 [OK]
 NEWRUN.TXT.1 [OK]
 NX.TXT.1 [OK]
```

6. Append a message from your terminal to the beginning of the file created in Example 5. Use the symbolic generation number -1 to specify this action.

```
@APPEND TTY:,BACKUP.TXT BACKUP.TXT.-1
 TTY:

!THIS IS A BACKUP FILE FOR ALL TEXT FILES.
^Z

 BACKUP.TXT.1  [OK]
```

2.4 ARCHIVE

Asks that a permanent off-line copy of specified files be made on magnetic tape, and prevents the disk copy (if retained) from being modified.

Format

@ARCHIVE (FILES) **filespec,...,**
@@subcommand

where:

filespec — is the specification of a file of which you want a permanent copy.

@@subcommand — means that after a final comma you can type the following optional subcommands:

- RETAIN — which causes the disk copies of the files being archived to be retained in your directory, rather than deleted and expunged.
- NORETAIN — which sets the file invisible.
- VISIBLE — which leaves file visible after ARCHIVE command.

Output

Notice of Archive Sent to Requestor

Whenever a file is taken off line as a result of your ARCHIVE command (for example, when you do not also give the RETAIN subcommand), the operator sends a mail message notifying you that the file has been archived.

Characteristics

Archived Files Unalterable

You cannot change the contents of files specified in an ARCHIVE command once the command is given, even if the files are not immediately copied to tape. This means that you cannot alter or add to them by using the EDIT or APPEND command, or overwrite them by using the COPY or RENAME command. In general, files for which you have requested

archival must not be given as the second filespec argument of these commands.

Archived Files Invisible

The files you specify in an ARCHIVE command ordinarily become invisible to most TOPS-20 commands as soon as the ARCHIVE command is given. However, if you include the RETAIN subcommand when giving the ARCHIVE command, the files remain visible. See Related Commands, below, for a list of commands you can use with invisible files.

Related Commands

CANCEL ARCHIVE	for canceling archival requests
DELETE, with the ARCHIVE subcommand	for deleting archived files
with the CONTENTS-ONLY subcommand	for deleting only the disk copy of files that also have a tape copy
DIRECTORY, with the ARCHIVE subcommand	for requesting information on archived files (visible and invisible) only
DIRECTORY, with the INVISIBLE subcommand	for requesting information on invisible files only
DISCARD	for giving up the tape copy of on-line files
INFORMATION ARCHIVE-STATUS	for determining if archival for the specified files (visible and invisible) has been accomplished
RETRIEVE	for restoring off-line files (visible and invisible) to on-line status
SET FILE INVISIBLE	for making visible files invisible
SET FILE VISIBLE	for making invisible files visible

Examples

1. Archive a file.

```
@ARCHIVE ARTEST.FIL
 ARTEST.FIL.1 [Requested]
```

2. Archive a file, but keep a copy on disk. Check the archive status of files.

```
@ARCHIVE ARCHEK.FIL,
@@RETAIN
@@
 ARCHEK.FIL.1 [Requested]
@INFORMATION ARCHIVE-STATUS
 ARCHEK.FIL.1 Archive requested,  Retain contents
 ARTEST.FIL.1 Archive requested
```

3. Attempt to use the EDIT editor to edit an archived file (first you must make it visible). Note that, afterwards, the unedited backup copy is the archived file, and that the edited file has no archive status.

```
@INFORMATION ARCHIVE-STATUS ARTEST.FIL
 ARTEST.FIL.1 Archive requested
@EDIT ARTEST.FIL

%No such filename, Creating New file
Input: ARTEST.FIL.2
00100   $
*EQ
@SET FILE VISIBLE ARTEST.FIL.1
 ARTEST.FIL.1 [OK]
@EDIT ARTEST.FIL.1
Edit: ARTEST.FIL.1
*P
00100  !TEST FILE FOR ARCHIVING
*I200
00200  !FIRST MODIFICATION
00300   $
*P^:*
00100  !TEST FILE FOR ARCHIVING
00200  !FIRST MODIFICATION
*EU

[ARTEST.FIL.2]
@INFORMATION ARCHIVE-STATUS ARTEST.*
 ARTEST.QIL.1 Archive requested
@TYPE ARTEST.QIL
!TEST FILE FOR ARCHIVING
@TYPE ARTEST.FIL
!TEST FILE FOR ARCHIVING
!FIRST MODIFICATION
```

```
@DIRECTORY ARTEST.*

   MISC:<LATTA>
 ARTEST.FIL.2
   .QIL.1
```

2.5 ASSIGN

Reserves a specific input-output device for your job.

Format

@ASSIGN (DEVICE) **dev:**

where:

dev: is the name of the device you want to assign. The colon after the device name is optional.

Restrictions

Assigning Magnetic Tape Drives

You can use the ASSIGN command to assign tape drives only if they are of the form MTAn:. Tape device names of the form MTn: are logical device names only, and are assigned automatically at the time of MOUNT TAPE commands.

Related Commands

DEASSIGN	for releasing a previously assigned device
MOUNT	for mounting a structure or magnetic tape set without assigning a specific drive
INFORMATION AVAILABLE DEVICES	for finding out which devices can be assigned or have been assigned to your job

Examples

1. Assign a tape drive to your job.

   @ASSIGN MTA0:

2. Find out which devices are available for timesharing use, then assign one to your job.

   @INFORMATION AVAILABLE DEVICES
   Devices available to this job:
    DSK, PS, SNARK, MISC, LANG, MTA1, LPT, LPT0, LPT1

```
 CDR, PCDR0, CDP FE1-15, PTY20-61, NUL, PLT, PLT0
 DCN, SRV
Devices assigned to/opened by this job: TTY41
@ASSIGN PCDR0:
```

2.6 ATTACH

Attaches a job to your terminal.

Format

@ATTACH (USER) **name** (JOB #) **number**
PASSWORD: **password**

where:

name — is the user name of the job's owner.

number — is the job number

Default the only job, or only detached job, or only job other than your current job, logged in under the user name you give.

password — is the associated password (not requested if you are currently logged in under the same user name as the job that you are attaching).

Characteristics

Current Job Detached

If you give the ATTACH command while logged in, your current job is detached. You can use the LOGOUT n command to log out this detached job.

Hint

Using ATTACH to Restore Phoned-in Jobs

If you log in to the system by telephone lines and service is interrupted for any reason, use the ATTACH command to restore the connection. If you do not do this within the time limit set by the system manager (usually five minutes), your job will be logged out automatically and you will have to log in again.

Warning

Attaching Attached Jobs

The system will ask you to confirm your choice with a carriage return before attaching to your terminal a job that is attached elsewhere. If you attach an attached job that is running a program, that program may be sent one or more CTRL/Cs, which can affect programs that handle CTRL/C themselves. To avoid this possibility you must give a DETACH command from the terminal to which the program's job is attached, then attach this job to your terminal with an ATTACH command.

Effect on Memory, Terminal, and Job

The ATTACH command affects neither memory nor the job that you are attaching (but see Warning, above), and leaves your terminal at TOPS-20 command level unless a program is being run by the job. If a program is being run, your terminal is left at command level, if any, in the job. Your terminal's characteristics will be those established in the job from which you gave the ATTACH command; if you were not logged in, they will be reset to system default characteristics.

Related Commands

SYSTAT	for finding out the user name and job number associated with any job
DETACH	for disengaging a job from your own terminal
UNATTACH	for disengaging a job from any other terminal

Examples

1. Attach your only job, which is presently detached.

```
@ATTACH LATTA
Password:___
```

2. Attach one of several detached jobs.

```
@ATTACH LATTA
  Job 37, Detached, Running DETACH
  Job 54, Detached, Running EXEC
Job:37
Password:___
```

3. Check your jobs with the SYSTAT command (your current job is marked with an asterisk [*]), then attach the only detached job. Verify the system's action.

```
@SYSTAT LATTA
  37    26  NEWRUN  LATTA
  58   DET  EXEC    LATTA
  59*  231  EXEC    LATTA
@ATTACH LATTA
 Detaching job # 59
@INFORMATION JOB-STATUS
 Host AURORA  Job 58, TTY314 kilpa.TOPS20.dec.com(TCP)
 User LATTA, PS:<LATTA>, Account 341
```

4. Start a program in one job. Then detach and continue it, and attach another of your jobs.

```
@INFORMATION JOB-STATUS
 Host AURORA  Job 9, TTY26 kilpa.TOPS20.dec.com(TCP)
 User LATTA, PS:<LATTA>, Account 341
@RUN FFACTOR

^C
@DETACH CONTINUE
 Detaching job # 9
^C  !Not displayed on terminal
 TEDDY, Controller Dept. TOPS-20 Monitor 7(7)
@SYSTAT LATTA
 9     DET   FFACTO    LATTA
45*     41   SYSTAT    LATTA
@ATTACH LATTA 45
 [Attached to TTY41, confirm]
Password:___
```

2.7 BACKSPACE

Moves a magnetic tape set backward over a specified number of files or records.

Format

@**BACKSPACE** (DEVICE) **dev: n units**

where:

dev: is the name of the tape set or magnetic tape drive that you want to move backward. The colon after the device name is optional.

n is the number of files or records over which you want to backspace.
Default n - 1

units is either FILES or RECORDS.
Default units - FILES

Restrictions

BACKSPACE with Open Files

If you have given a CTRL/C to exit from a program that has opened a magnetic tape drive and you then give the BACKSPACE command for that tape drive, the system will first ask if you want to close the associated file. You must do so for BACKSPACE to succeed, but you will probably be unable to continue the program from that point because the file will now be closed.

RECORDS Argument Used for Unlabeled Tapes Only

You cannot give the RECORDS argument to the BACKSPACE command when using a labeled tape, because read and write operations for labeled tapes always move the tape to the beginning of a file first.

Related Commands

SKIP for moving a magnetic tape set forward

REWIND for backspacing a tape volume or tape set to its logical beginning (the beginning of the first file)

UNLOAD	for completely rewinding a magnetic tape onto the source reel

Examples

1. Backspace your magnetic tape one file.

   @BACKSPACE MTA0: 1 FILE

2.8 BLANK

Clears your video terminal screen.

Format

@**BLANK** (SCREEN)

Characteristics

This command moves the cursor to line 1 of the screen, providing you with a clean area for typing commands and receiving system output.

Restriction

The BLANK command functions only if you have set your terminal type with the TERMINAL command or the unsupported TTYINI program.

2.9 BREAK

Breaks the communication link made between terminals with the TALK command.

Format

@**BREAK** (LINKS WITH) **argument**

where:

argument — is a user name or line number.
Default – all communication links

Restrictions

Does Not End Advice

The BREAK command, given at either of two terminals joined by the ADVISE command, does not end advice. To end an ADVISE link, type CTRL/E.

Related Commands

TALK	for establishing communication links with another terminal
REFUSE LINKS	for preventing future communication links with your terminal

Examples

1. Use the BREAK command to end a TALK session with another user.

   @BREAK

2. Use the TALK command to speak to another user, then use BREAK to end the conversation.

   @TALK PORADA
   LINK FROM LATTA, TTY 41
   @!HI. CAN YOU TELL ME WHERE THE PROJECT ESTIMATES ARE STORED?
   @!SURE: THEY'RE ON TAPE LS2.0 IN THE LIBRARY.
   @!THANKS. BYE
   @BREAK

2.10 BUILD

Creates, modifies, or deletes a subdirectory to a directory to which you have write access.

Format

@BUILD (DIRECTORY NAME) **str:<directory>**
@@subcommand

where

str:	is the name of the (mounted) structure containing the directory you are building.
directory	is the name of the directory you are building. The directory name can contain 39 or fewer alphanumeric characters, including the following special characters: '_', '-' and '$'. The name must be enclosed in angle brackets <> or square brackets [].
@@subcommand	indicates that you automatically enter subcommand mode after completing the BUILD command line.

Summary of BUILD Subcommands (defaults in boldface)

	ABORT	
|	ABSOLUTE-INTERNET-SOCKETS	
	ACCOUNT-DEFAULT account	
	ARCHIVE-ONLINE-EXPIRED-FILES	
	CONFIDENTIAL	
	DECNET-ACCESS	
	DEFAULT-FILE-PROTECTION octal protection code	**Default** n - 777700
	DIRECTORY-GROUP group number	
	DISABLE	
	ENABLE	
	ENQ-DEQ	
|	EXPIRATION-OF-PASSWORD	
|	EXPIRE	
	FILES-ONLY	
	GENERATIONS n	**Default** n - 1
	IPCF	
|	INTERNET-ACCESS	
|	INTERNET-WIZARD	
	KILL	

```
            ---
            | NAME-ONLY
        LIST  | FAST
            | VERBOSE
            ---
        MAINTENANCE
        MAXIMUM-SUBDIRECTORIES n                          Default n - 0

            ---
|           | ABSOLUTE-INTERNET-SOCKETS
            | ARCHIVE-ONLINE-EXPIRED-FILES
            | CONFIDENTIAL
            | DECNET-ACCESS
            | DIRECTORY-GROUP group number
            | ENQ-DEQ
|           | EXPIRATION-OF-PASSWORD
            | FILES-ONLY
        NOT   | IPCF
|           | INTERNET-ACCESS
|           | INTERNET-WIZARD
            | KILL
            | MAINTENANCE
            | OPERATOR
            | REPEAT-LOGIN-MESSAGES
|           | SECURE
            | SEMI-OPERATOR
            | SUBDIRECTORY-USER-GROUP group number
            | USER-OF-GROUP group number
            | WHEEL
            ---

        NUMBER octal directory number
        OFFLINE-EXPIRATION-DEFAULT date or +n             Default n - 90
        ONLINE-EXPIRATION-DEFAULT date or +n              Default n - 60
        OPERATOR
        PASSWORD 1- to 39-character word
                 ---
        PERMANENT | pages                                 Default n - 250
                  | INFINITY
                  ---
        PRESERVE
        PROTECTION octal protection code                  Default n -
                                                                    777700
        PUSH
        REPEAT-LOGIN-MESSAGES
|       SECURE
        SEMI-OPERATOR
        SUBDIRECTORY-USER-GROUP group number
        TOPS10-PROJECT-PROGRAMMER-NUMBER n,n
        USER-OF-GROUP group number
        WHEEL
```

WORKING | pages **Default** n - 250
| INFINITY

BUILD Subcommands

ABORT — cancels all work done during current BUILD command. If directory was new, it does not exist; if old, it remains unchanged.

| ABSOLUTE-INTERNET-SOCKETS — allows the directory owner to establish | INTERNET Protocol network connections | using 32-bit absolute socket numbers; users with Wheel or Operator capabilities can also perform this function. For use only with systems | that are connected to a TCP/IP network. Requires WHEEL or OPERATOR capabilities.

ACCOUNT-DEFAULT account — causes the specified account to be charged for a terminal session whenever the user does not include an account in his LOGIN command.

ARCHIVE-ONLINE-EXPIRED-FILES — causes on-line files that have expired to be marked for archiving.

CONFIDENTIAL — grants the directory owner confidential information access capabilities, allowing him to obtain confidential information within the system through certain monitor calls. See the TOPS-20 Monitor Calls Reference Manual for details. Requires WHEEL or OPERATOR capabilities.

DECNET-ACCESS — allows the directory owner to establish DECNET network connections. This subcommand works in conjunction with pre-established system manager controls. Requires WHEEL or OPERATOR capabilities.

DEFAULT-FILE-PROTECTION octal protection code — assigns this number as default for the

protection code of each file subsequently placed in the directory.

The protection code is constructed (by addition) from the octal values shown below:

77 full access to the file
40 read the file
20 write and delete the file
10 execute the program contained in the file
04 append to the file
02 list the files specification using DIRECTORY-class commands
00 no access to the file
Default code - 777700

See the TOPS-20 User's Guide for more information about protection codes.

DIRECTORY-GROUP group number
places the directory in a group, thereby allowing users in the same group access to it according to the middle two digits of the protection code, and access to files in the directory according to the middle two digits of each file's protection code. You can assign up to 40 directory group numbers to each directory, with values ranging from 1 through 262143 (2**18 - 1). See the TOPS-20 System Manager's Guide for a discussion of groups.

DISABLE
suspends any special capabilities that you may have activated with the ENABLE subcommand or the ENABLE TOPS-20 command.

ENABLE
allows you to activate any privileged capabilities that the system manager has given you and that you may need during the BUILD session.

ENQ-DEQ
grants the directory owner the ability to perform global Enqueue and Dequeue functions; these are discussed in the TOPS-20 Monitor Calls Reference Manual. Requires WHEEL or OPERATOR capabilities.

EXPIRATION-OF-PASSWORD n	sets the password expiration date for the directory. You can specify the date and time to expire the password. The user can log in again once after the password expires and is prompted for a new password.
EXPIRE	sets the password expiration date to -1, which means that the user cannot login to the account because the account is expired.
FILES-ONLY	declares the directory to be a files-only directory; one not associated with a user. See Restrictions - Files-only Directories, below.
GENERATIONS n	specifies a default for the number of successive generations of files to be retained in the directory. This number must be from 0 to 15, with 0 meaning an infinite number. **Default** n - 1
IPCF	allows the directory owner to execute all privileged IPCF functions; these are discussed in the TOPS-20 Monitor Calls Reference Manual. Requires WHEEL or OPERATOR capabilities.
INTERNET-ACCESS	allows the directory owner to establish INTERNET Protocol network connections. This subcommand works in conjunction with pre-established system manager controls. Requires WHEEL or OPERATOR capabilities.
INTERNET-WIZARD	allows the directory owner to use special queues for sending and receiving information using the INTERNET Network Protocol. For use only with systems that are connected to a TCP/IP network. Requires WHEEL or OPERATOR capabilities.
KILL	eliminates the directory and any files it contains from the system; you must confirm this subcommand with an extra RETURN.

	NAME-ONLY	
LIST	FAST	provides a listing at your terminal of parameter values set for the directory by TOPS-20 commands, BUILD subcommands, or by default. The FAST listing always includes the subdirectory's name, working and permanent storage limits, and, if they have been set, directory number, account default, maximum number of subdirectories allowed to this directory, the date and time of last log-in, group memberships, and user group numbers assignable by this directory. The VERBOSE listing adds the other values that can be assigned by BUILD subcommands, while NAME-ONLY restricts output to the directory name. **Default** - FAST
	VERBOSE	

MAINTENANCE		allows the directory owner to execute certain system maintenance functions or monitor calls; discussed in the TOPS-20 Monitor Calls Reference Manual. Requires WHEEL or OPERATOR capabilities.
MAXIMUM-SUBDIRECTORIES n		allows the owner of this directory to build up to n subdirectories of his own, and subtracts an equal number from the value of this parameter for the superior directory.

NOT	ABSOLUTE-INTERNET-SOCKETS	withdraws the specified subcommand
	ARCHIVE-ONLINE-EXPIRED-FILES	
	CONFIDENTIAL	
	DECNET-ACCESS	
	DIRECTORY-GROUP group number	
	ENQ-DEQ	
	EXPIRATION-OF-PASSWORD	
	FILES-ONLY	
	IPCF	
	INTERNET-ACCESS	
	INTERNET-WIZARD	
	KILL	
	MAINTENANCE	
	TOPS10-PROJECT-PROGRAMMER-NUMBER	
	OPERATOR	
	REPEAT-LOGIN-MESSAGES	
	SECURE	
	SEMI-OPERATOR	

| SUBDIRECTORY-USER-GROUP group number
| USER-OF-GROUP group number
| WHEEL

NUMBER octal directory number
assigns a specific directory number to a new directory (note: usually the default is adequate). Directory numbers 1 through 17 must never be assigned by users, as they are reserved for system use.

Default directory number - assigned by system

OFFLINE-EXPIRATION-DEFAULT date or +n
establishes the tape expiration date for files that are to go off line because of migration or archiving. If you specify "+n", the expiration date will be n days from the date the files are moved off line.

The default date cannot exceed the system maximum. Check the system maximum with the command INFORMATION (ABOUT) SYSTEM-STATUS.

ONLINE-EXPIRATION-DEFAULT date or +n
establishes the disk expiration date for files that are to be created in the directory. If you specify "+n", the expiration date will be n days from the creation date.

OPERATOR
grants Operator capabilities to the owner of the directory; these are discussed further in the TOPS-20 Operator's Guide. Requires WHEEL or OPERATOR capabilities.

PASSWORD 1- to 39-character word
assigns a password, consisting of alphanumeric characters and hyphens (-), to the directory. You can include any special characters (except '@', ';', '!' and '?') in a password by typing CTRL/V before each special character. Unlike special characters in file specifications, CTRL/V is required only when creating the password, not when

using it.

PERMANENT | --- pages | INFINITY ---

allocates permanent disk storage capacity (in pages) to the directory, and subtracts an equal number from the permanent disk storage capacity of the superior directory. INFINITY allows users with WHEEL or OPERATOR capabilities to allocate an unlimited number of pages to the directory. The permanent disk storage of the superior directory must also be INFINITY. The number of pages that can be used is limited to the number of free pages on the structure.

The INFINITY argument is intended for special system directories; it is not intended for general users.

Default pages - 250

PRESERVE

preserves the values of the superior directory's PERMANENT, WORKING and MAXIMUM-SUBDIRECTORIES parameters. PRESERVE stops the PERMANENT, WORKING, and MAXIMUM-SUBDIRECTORIES subcommands from subtracting the values from the quotas in the superior directory. Requires WHEEL or OPERATOR capabilities.

PROTECTION octal protection code

assigns the given directory protection code to the directory. The protection code is constructed (by addition) from the octal values shown below:

77 full access to the directory
40 access to files in the directory (including expunging individual files), consistent with the file protection of the files
10 connect to the directory without giving a password, undelete files, expunge the entire directory, and change times, dates, and accounting information for files. All other access is governed by the file protection of each file.
04 create files in the directory
00 no access to the directory

Default code - 777700

See the TOPS-20 User's Guide for more information about protection codes.

PUSH — creates an EXEC level inferior to the one from which you issued the BUILD command and leaves your terminal at this new level. You can then issue TOPS-20 commands to create conditions or obtain information that you may need during the BUILD session. Give the POP command to return to BUILD. See Example 6.

This subcommand refers to the EXEC defined by the logical name DEFAULT-EXEC:. You can use the DEFINE command to define the job logical name, DEFAULT-EXEC:, with the name of the EXEC you want to create each time you PUSH.

REPEAT-LOGIN-MESSAGES — causes all system messages (mail sent by privileged users to all users, contained in the file, PS:<SYSTEM>MAIL.TXT) to be printed on the user's terminal each time he logs in to this directory. If this subcommand is not given, only those system messages created since the last time he logged in are printed.

| SECURE — sets any new files created in the specified directory secure by default. When a file is secure, the Access Control Job checks to see if the user has access to that file before the user can read, write, append, rename, delete, set secure, or set unsecure that file.

SEMI-OPERATOR — creates or modifies directories to include the SEMI-OPERATOR privileges, which allows unprivileged users to run OPR and execute certain OPR commands. These commands are strictly for accessing information and controlling certain devices. See the TOPS-20 Operator's Command Language Manual for more information on SEMI-OPERATOR. Requires WHEEL or OPERATOR privileges.

SUBDIRECTORY-USER-GROUP group number — allows propagation of any or all of the group numbers in a directory's user group list to the subdirectories of that

directory. Issuing this subcommand is the first step required in establishing subdirectory group rights. You complete the process by issuing the USER-OF-GROUP subcommand for each subdirectory. You can assign up to 40 subdirectory user group numbers to each directory, with values ranging from 1 to 262143 (2**18 - 1).

TOPS10-PROJECT-PROGRAMMER-NUMBER project number, programmer number

allows TOPS-10 programs that require a project-programmer number (PPN) to create subjobs to be compatible with TOPS-20. The project number is an octal number in the range 10-377777. The programmer number is a six-digit octal number. Separate the project number from the programmer number with a comma, for example 17,76.

USER-OF-GROUP number — assigns the directory owner to the given group. You can assign up to 40 group numbers to each directory, with values ranging from 1 to 262143 (2**18 - 1). See the TOPS-20 System Manager's Guide for a discussion of groups.

WHEEL — grants WHEEL capabilities to the owner of the directory, allowing him to perform all the privileged functions available on the system; these are discussed further in the TOPS-20 Operator's Guide. Requires WHEEL or OPERATOR capabilties.

WORKING { pages | INFINITY } — allocates working disk storage capacity (in pages) to the directory, and subtracts an equal number from the working disk storage capacity of the superior directory. This working space is temporary and is allocated to the directory only while the user is logged in. Ordinarily, working and permanent storage limits are equal.

INFINITY allows users with WHEEL or OPERATOR capabilities to allocate an unlimited number of pages to the directory. The working disk storage of

the superior directory must also be INFINITY. The number of pages that can be used is limited to the number of free pages on the structure.

The INFINITY argument is intended for special system directories; it is not intended for general users.

Default pages - 250

Characteristics

BUILD and ^ECREATE

The BUILD command is identical in format to the privileged ^ECREATE command. If you use BUILD with WHEEL or OPERATOR capabilities enabled, it has the same power as ^ECREATE, namely, to create directories and modify the parameters of any directory on the system. Without these capabilities, you can use BUILD to modify a more restricted set of directories: you can modify a directory if you have write access to the immediately superior directory. The LOGIN, CONNECT, or ACCESS command obtains write access to the superior directory; or, if you have sufficient group rights to the superior directory, you can use BUILD to modify its subdirectories.

More Information

For a description of using ^ECREATE to create directories, see the TOPS-20 Operator's Command Language Reference Manual.

Quotas Subtracted from the Superior Directory's Allotments

Working and permanent disk storage page limits, and the maximum number of subdirectories allowed to a subdirectory are subtracted from the quotas allocated to the immediately superior directory. This subtraction occurs at the time of their allotment to a subdirectory. If the superior directory's quota is not sufficient, the BUILD command will fail. (Note that if you have enabled WHEEL or OPERATOR capabilities, you can stop the subtraction of quotas from the superior directory with the PRESERVE subcommand.)

To increase the superior directory's quota or any of these quantities you must either kill some of its subdirectories or reduce their allotments of the quantity. Or you can ask the system manager to increase the allotment of the superior directory. Remember that unless you specify working and

permanent page limits, they will assume a default value of 250 pages. The BUILD command will fail in this case if there are not at least 250 pages free in the immediately superior directory.

Assigning Infinite Quotas

If you have WHEEL or OPERATOR capabilities enabled, you can assign the maximum storage limit of 34359738367 (2**35-1) to a directory. This will appear in the response to an INFORMATION DIRECTORY command as +INF, denoting infinite storage capacity. If you then use the BUILD command to construct subdirectories to this directory, any disk storage capacity assigned, even the maximum, will not be subtracted from the superior directory. You can use this feature to assign infinite storage capacity to a number of users sharing a private structure. Then these users may use storage space on the structure without limit until the disk pack fills up.

Hints

Keeping Track of Subdirectories

Subdirectories appear as files of type .DIRECTORY in the immediately superior directory, so the DIRECTORY *.DIRECTORY command for the superior directory will indicate any existing subdirectories. To suppress the listing of these files you can use the SET FILE PROTECTION command to give them a protection of 000000, but then you must specify the files completely (including generation number) to access them in the future.

If there are two or more levels of subdirectories below a superior directory, you can do something else to allow a listing of them: put each subdirectory into a group of which the owner of the highest-level superior directory is a member. Then, if you obtain the group rights of this owner (by using the LOGIN or ACCESS command if the superior directory is on the public structure, or ACCESS if it is on another structure), the INFORMATION DIRECTORY <directory.*> command with the NAME-ONLY subcommand will produce a listing of subdirectories at every level beneath the superior directory. For this feature to operate properly the group field of each subdirectory's protection code must be at least 40.

Modifying Subdirectories Easily

By following the above procedure, that is, by making subdirectories at every level members of groups of which the

owner of the highest-level superior directory is also a member, you make the modification of these subdirectories much easier. You can use the BUILD command to modify these subdirectories or read and write to them, as long as you have the group memberships of this owner. You need not connect to each subdirectory's immediate superior to make modifications.

Restrictions

Giving Capabilities to Subdirectory Owners

To give capabilities (WHEEL, OPERATOR, SEMI-OPERATOR, ABSOLUTE-ARPANET-SOCKETS, ARPANET-WIZARD, CONFIDENTIAL, ENQ-DEQ, IPCF, or MAINTENANCE) to a subdirectory owner, you must have these capabilities yourself, and they must be enabled at the time of the BUILD command. WHEEL and OPERATOR capabilities allow you to assign any capabilities. The INFORMATION DIRECTORY command for your log-in directory tells you which capabilities you have, if any.

Modifying Other Directories

Unless you have WHEEL or OPERATOR capabilities enabled, you can use the BUILD command to modify the parameters of only those directories subordinate to a directory to which you have write access. (See Characteristics - BUILD and ^ECREATE, and Hints - Modifying Subdirectories Easily, above.) If your installation allows it, you can use the SET DIRECTORY command to change some parameters of these directories.

Files-only Directories

By giving the FILES-ONLY subcommand you make the directory a files-only directory (see Figure 1). A files-only directory is not associated with a user and so should not be given capabilities or user group memberships. Although a files-only directory can have subdirectories, none of these can be a user directory. You cannot give the ACCESS or LOGIN command for a files-only directory.

Killing Directories

You cannot kill a directory that has subdirectories; first you must kill those subdirectories one by one. (When you kill a directory, the files it contains are deleted and expunged.) Also, you cannot kill a directory if you are logged into it or connected to it, or there are open files on it.

Restricted Use of BUILD Command

Your system manager may make the BUILD and SET DIRECTORY commands available only to users with enabled Wheel or Operator capabilities.

Related Commands

INFORMATION DIRECTORY	for examining the parameters established for a directory
INFORMATION DISK-USAGE	for determining how much of a directory's disk space is already assigned to files
SET DIRECTORY	for changing certain directory parameters

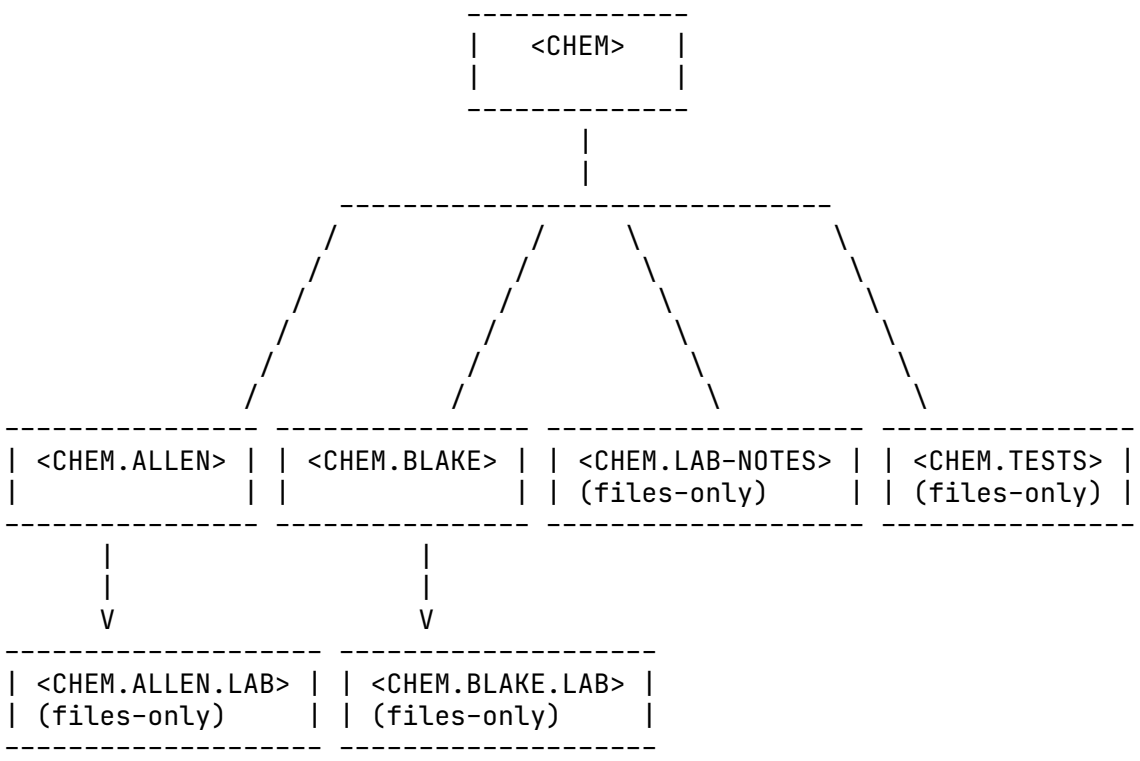


Figure 2-1: Directories and Subdirectories

Examples

The examples show how a user with a directory named <CHEM> builds subdirectories in the pattern shown in Figure 1.

1. Build directories for two of your students or employees, assigning disk space and passwords and placing them in one of your directory groups; check their parameters.

```
@BUILD <CHEM.ALLEN>
[New]
@@WORKING 50
@@PERMANENT 50
@@PASSWORD 619JIM
@@DIRECTORY-GROUP 2391
@@LIST
 NAME <CHEM.ALLEN>
 Working disk storage page limit 50
 Permanent disk storage page limit 50
 Account default for LOGIN - none set
 Directory groups 2391

@@
```

```
@BUILD <CHEM.BLAKE>
[New]
@@WORKING 50
@@PERMANENT 50
@@PASSWORD 127BIL
@@DIRECTORY-GROUP 2391
@@LIST
 Name <CHEM.BLAKE>
 Working disk storage page limit 50
 Permanent disk storage page limit 50
 Account default for LOGIN - none set
 Directory groups 2391
@@
```

2. Modify Blake's directory to allow him to create two subdirectories.

```
@BUILD <CHEM.BLAKE>
[Old]
@@MAXIMUM-SUBDIRECTORIES 2
```

3. Build a files-only directory to store examination questions.

```
@BUILD <CHEM.TESTS>
[New]
@@FILES-ONLY
@@WORKING 10
@@PERMANENT 10
@@PASSWORD MINERVA
@@DIRECTORY-GROUP 2391
```

4. Build a files-only directory as a library directory for your subdirectory owners. Place the directory and these users in the same group.

```
@BUILD <CHEM.LAB-NOTES>
[New]
@@FILES-ONLY
@@WORKING 25
@@PERMANENT 25
@@PROTECTION 774000
@@DEFAULT-FILE-PROTECTION 775200
@@DIRECTORY-GROUP 2392
@@
@BUILD <CHEM.ALLEN>
[Old]
@@USER-GROUP 2392
@@
@BUILD <CHEM.BLAKE>
[Old]
@@USER-GROUP 2392
```

5. User Blake quits. Delete his directory.

```
@BUILD <CHEM.BLAKE>
[Old]
@@KILL
[Confirm]
```

6. Modify a subdirectory so that the subdirectory's owner will have 350 disk pages available.

```
@BUILD <TUCKER.TEST>
[Old]
@@PERMANENT 350
@@
?Request exceeds superior directory permanent quota.
Please fix incorrect subcommands.
```

The action above produced an error message. To correct the error, PUSH out of the BUILD session to learn what the superior directory's permanent quota is.

```
@@PUSH

 TOPS-20 Command processor 7(70)
@INFORMATION DISK-USAGE
 PS:<TUCKER>
 70 Pages assigned
 261 Working pages, 261 Permanent pages allowed
 7546 Pages free on PS:, 144454 pages used.
```

Then return to the BUILD session, and specify a permanent quota that is less than the superior directory's quota of 261 disk pages.

```
@POP
[Continuing BUILD of directory PS:<TUCKER.TEST>]
@@PERMANENT 170
@@
```

2.11 CANCEL

Cancels requests made with a queue-class command.

Format

@**CANCEL** (REQUEST TYPE) **queue** (ID) **identifier/switch(es)**

where:

queue — is the name of the queue, chosen from the following list:

- ARCHIVE — for requests made using the ARCHIVE command
- BATCH — for requests made using the SUBMIT command
- CARDS — for requests made using the PUNCH CARDS command
- MOUNT — for requests made using the MOUNT STRUCTURE or MOUNT TAPE command
- PAPER-TAPE — for requests made using the PUNCH PAPER-TAPE command
- PLOT — for requests made using the PLOT command
- PRINT — for requests made using the PRINT command
- RETRIEVE — for requests made using the RETRIEVE command

request ID number
: the unique identifier assigned by the system to your request. This is the number appearing under the heading "Req#" in the list of requests shown by the appropriate INFORMATION command (see Related Commands, below). To cancel archival requests, use "filespec" argument instead.

jobname
: the jobname of the request,

either:

- the first six characters of the first filename in the request or the argument you supplied to a /JOBNAME switch when making the request (for output and batch requests), or
- the first six characters of each filename in the request (for retrieval requests), or
- the first six characters of the structure alias or tape set n. This is the name appearing under the heading "Name", "Req Name", or "Job Name" in the list of requests shown by the appropriate INFORMATION command (see Related Commands, below).

filespec — the specification of a file. Use this argument to cancel archival requests.

* — the asterisk identifier cancels all your requests in the specified queue.

/switch — is one or more of the following switches:

/DESTINATION-NODE:node
which lets you cancel a print request to a remote printer in the same TOPS-20 cluster as the requesting node. This switch cancels only the print requests that were made from the local node. Other print requests made on the remote node are not affected.

/JOBNAME:jobname
which gives the jobname of the request you want to cancel.

See Special Cases - /JOBNAME Switch, below.

/SEQUENCE:n — which gives the sequence number of the batch or output request you want to cancel. The INFORMATION BATCH-REQUESTS or INFORMATION OUTPUT-REQUESTS command with the /ALL switch gives the sequence number assigned to these requests. Use this switch in CANCEL commands placed within batch jobs; then you can cancel requests made earlier in the batch job even though you do not know the request ID number.

/USER:user name — which cancels the specified request entered under the given user name. Use an asterisk (*) both for request ID number and as argument to this switch to cancel all requests of all users in the specified queue. For privileged users only. /USER is required to modify or cancel requests from users other than yourself.

Output

When you complete a CANCEL command removing a request, the system responds with "[1 Job Canceled]" and makes the appropriate deletion from the indicated queue. If the job is being processed, the response is "[1 Job Canceled (1 was in progress)]", but if the job is already finished, it is simply "[No Jobs Killed]".

Characteristics

Request ID or Jobname as Argument to CANCEL

You can cancel a single queue request (those made with Queue-class commands - MOUNT, PLOT, PRINT, PUNCH, RETRIEVE, or SUBMIT) by giving either its request ID number or its jobname as the second argument of a CANCEL command. This

argument is interpreted as a request ID number unless it includes one or more non-numeric characters. If the argument includes non-numeric characters it is interpreted as a jobname. By giving a jobname as the second argument of a CANCEL command, you cancel all your requests of that jobname in the specified queue. But see also Special Cases - /JOBNAME Switch, below.

Special Cases

/SPOOLED-OUTPUT Switch

You can give the special switch, /SPOOLED-OUTPUT, after the CARDS, PAPER-TAPE, PLOT, or PRINT argument to the CANCEL command. By doing so you cause any accumulated requests in the spooler queue for the appropriate output device (CDP:, PTP:, PLT:, or LPT:, respectively) to be canceled, rather than filled when you log out. Do not give any further arguments to a "CANCEL queue /SPOOLED-OUTPUT" command.

If any spooled file print requests have been routed to a remote node, use the /DESTINATION-NODE switch to cancel them.

/JOBNAME Switch

In the singular case when you want to cancel several queue requests of the same jobname using a single command, and that jobname is purely numerical (for example, 5045), you must use the /JOBNAME:jobname switch as the second argument to the CANCEL command. Do not also give the request ID or jobname as a command argument if you give the /JOBNAME:jobname switch.

Restrictions

Cannot Cancel Filled Tape-mount Requests

You cannot use the CANCEL command to withdraw a MOUNT TAPE request once the first volume of tape has been mounted (once you have received a message of the form, [setname defined as MTn:]). Use the DISMOUNT command to give up your tape resource in this case. Note that the DEASSIGN or LOGOUT command will also dismount the tape set.

Cannot Cancel Certain Archival Requests

You cannot use the CANCEL command to withdraw an archival request once the operator has initiated archival procedures. Thus, even though files remain on disk between the

operator's first and second archive runs, you cannot cancel a request during this time. If you try to cancel a request after archiving has begun, you receive the error message:

?File has archive status: filename

Note that this error does not terminate a multifile CANCEL ARCHIVE command (for example, CANCEL ARCHIVE *.*); the TOPS-20 command processor continues processing each remaining filename in the request. Cancel requests for these remaining files are judged individually.

Related Commands

ARCHIVE	for requesting archival of a file
INFORMATION ARCHIVE-STATUS	for finding out the archival status of files
INFORMATION BATCH-REQUESTS	for examining requests in the batch input queue
INFORMATION MOUNT-REQUESTS	for examining requests in the structure- and tape-mount queue
INFORMATION OUTPUT-REQUESTS	for examining requests in the line printer, plotter, card punch, and paper tape punch queues
INFORMATION RETRIEVAL-REQUESTS	for examining requests in the retrieval queue
MODIFY	for changing requests without removing them
MOUNT	for placing requests in the structure- or tape-mount queue
PLOT	for placing requests in a plotter queue
PRINT	for placing requests in a line printer queue
PUNCH	for placing requests in the card- or paper-tape-punch queue
RETRIEVE	for placing requests in the retrieval queue

SUBMIT	for placing requests in the batch input queue

Examples

1. Cancel a specific print request.

```
@CANCEL PRINT REMAX
[1 Job canceled]
```

2. Cancel all your batch requests.

```
@CANCEL BATCH *
[3 Jobs canceled]
```

3. Find out what line printer requests you have made, then cancel one of two jobs bearing the same jobname.

```
@INFORMATION OUTPUT-REQUESTS /USER

Printer Queue:
Job Name    Req#   Limit   User
--------    ----   -----   -------
  MYCOPY     142      81   LATTA  /Lower  /After: 8-Nov-79 18:00
  MYCOPY     143      81   LATTA  /After: 8-Nov-79 18:00
  MYCOPY     144      81   LATTA  /After: 8-Nov-79 18:00
  HOLMAX     141     200   LATTA  /After: 8-Nov-79 17:00
  HOLMAX     140     200   LATTA  /After: 8-Nov-79 18:00
There are 5 Jobs in the Queue (None in Progress)

@CANCEL PRINT  141
[1 Job Canceled]
```

4. Get a list of your printing jobs (and some of the switches you gave), then cancel three of them.

```
@INFORMATION OUTPUT-REQUESTS /ALL/USER

Printer Queue:
Job Name   Req#   Limit   User
--------   ----   -----   -------
  MYCOPY    142      81   LATTA  /Lower  /After: 8-Nov-79 18:00
           /Prio:20   /Seq:1728
  MYCOPY    143      81   LATTA  /After: 8-Nov-79 18:00
                               /Seq:1729
  MYCOPY    144      81   LATTA  /After: 8-Nov-79 18:00
                               /Seq:1730
  HOLMAX    140     200   LATTA  /After: 8-Nov-79 18:00
                               /Seq:1726
There are 4 Jobs in the Queue (None in Progress)
```

```
@CANCEL PRINT MYCOPY
[3 Jobs Canceled]
```

5. Cancel a plotter request.

```
@CANCEL PLOT 94
[1 Job Canceled]
```

6. Cancel a mount request for a structure or a tape set.

```
@CANCEL MOUNT 24
[1 mount request canceled]
```

7. Find out what requests are in the mount queue. Cancel your structure mount request.

```
@INFORMATION MOUNT-REQUESTS

Tape/Disk Mount Queue:
Volume    Status  Type   Write   Req Name  Req#  Job#  User
-------  -------  ----  -------  --------  ----  ----  ----------
MARK     MTA1     Tape  Enabled  MARK       126    60  HOVSEPIAN
TAPE     MTA3     Tape  Enabled  TAPE       148    13  WALLACE
LATB     Waiting  Disk           LATB       157    65  LATTA
There are 3 Requests in the Queue

@CANCEL MOUNT 157
[1 mount request canceled]
```

8. Find out what requests are in the mount queue. Cancel a mount request that has not yet been filled, and dismount a tape that has been mounted.

```
@INFORMATION MOUNT-REQUESTS

Tape/Disk Mount Queue:
Volume    Status  Type   Write   Req Name  Req#  Job#  User
-------  -------  ----  -------  --------  ----  ----  ----------
MARK     MTA1     Tape  Enabled  MARK        29    15  HOVSEPIAN
DBL02    MTA0     Tape  Enabled  LAT         31    24  LATTA
NCV19    Waiting  Tape  Enabled  NCV         32    24  LATTA
There are 3 Requests in the Queue

@CANCEL MOUNT 32
[1 mount request canceled]
@DISMOUNT TAPE LAT:
[Tape dismounted, logical name LAT: deleted]
```

9. Cancel a remote print request.

```
@CANCEL PRINT SUM7/DESTINATION-NODE::LEZAH
```

```
[1 print request canceled]
```

2.12 CLOSE

Closes open files in your job and releases their JFNs.

Format

@CLOSE (JFN) **n**

where:

n is the JFN (Job File Number) of a file.
Default n - all JFNs for open and closed files

Output

When the CLOSE command is completed, the system prints a message on your terminal for each JFN that it has closed or attempted to close.

Characteristics

CLOSE Usually Unnecessary

The CLOSE command is used to preserve the contents of a file after the abnormal termination of the program that opened it. Under ordinary conditions you do not need to use the CLOSE command.

Special Cases

CLOSE For Closed Files

If you give the CLOSE command for an existing but closed JFN, the JFN is released.

Restrictions

Closing Mapped Files

You cannot close files that are mapped into memory; in this case you may give the RESET command, which clears memory. (Note, however, that RESET will delete and expunge a mapped file if the file is new, rather than save it as CLOSE would. To save a new mapped file, give the SET PAGE-ACCESS 0:777 NONEXISTENT command, and follow this with CLOSE.) If RESET by itself does not close the file, you may first have to

give the POP command to return to a higher level of the TOPS-20 command processor; then give the RESET command.

Once memory has been cleared, all files are closed for processes at the current and lower levels of TOPS-20.

Related Commands

INFORMATION FILE-STATUS	for determining which files are open in your job
INFORMATION MEMORY-USAGE	for determining which files are mapped
RESET	for closing mapped files and clearing memory
SET PAGE NONEXISTENT	for removing specified pages from memory

Examples

1. Close an open file.

```
@CLOSE 4
 4  EDIT-BUFFER.OUT.100036   [OK]
```

2. Try to close all your open files. Give the RESET command to close those that are mapped. (Note that the file EXEC.EXE, containing the TOPS-20 command processor, cannot be closed.)

```
@CLOSE
 3  TEST1.CBL.1   [OK]
 2  PS:<TEST>EDIT.EXE.4  Can't close file - File still mapped
 1  PS:<SYSTEM>EXEC.EXE.3  Can't close file - File still mapped
@RESET
@CLOSE
 1  PS:<SYSTEM>EXEC.EXE.3  Can't close file - File still mapped
```

2.13 COMPILE

Translates source files into object (relocatable binary) files.

Format

@COMPILE (FROM) **/switch(es) source/switch(es) object,...**

where:

switches — are keywords chosen from the list below, indicating your choice of COMPILE command options. They have different effects depending on their position in the command line: placed before all files in the command, they act as defaults for all; otherwise, they affect only the nearest preceding file.
Defaults are shown in the list of switches

source — is the file specification of a source program. The filename must be of 6 or fewer characters, and the file type of 3 or fewer characters; you cannot use a generation number.
Default - arguments you gave in your last LOAD-class command

object — is the filename you choose for the object file; it must be of 6 or fewer characters.
Default - filename of the source file (file type is .REL)

Summary of COMPILE Command Switches (defaults in boldface)

/10-BLISS
/36-BLISS
/68-COBOL
/74-COBOL
/ABORT
/ALGOL
/BINARY
/COBOL
/COMPILE
/CREF
/CROSS-REFERENCE
/DDT
/DEBUG
/FAIL
/FLAG-NON-STANDARD
/FORTRAN
/LANGUAGE-SWITCHES:"/switch(es)"

/LIBRARY
/LIST
/MAC
/MACHINE-CODE
/MACRO
/MAP
/NOBINARY
/NOCOMPILE
/NOCREF
/NOCROSS-REFERENCE
/NODBUG
/NOFLAG-NON-STANDARD
/NOLIST
/NOLIBRARY
/NOMACHINE-CODE
/NOOPTIMIZE
/NOSEARCH
/NOSTAY
/NOSYMBOLS
/NOWARNINGS
/OPTIMIZE
/PASCAL
/RELOCATABLE
/SAIL
/SEARCH
/SIMULA
/SNOBOL
/STAY
/SYMBOLS
/WARNINGS

Descriptions of these switches are given below. Although the system will not reject switches described under any of the LOAD-class commands, only those switches commonly associated with COMPILE are described here.

COMPILE Command Switches

/10-BLISS compiles the file using the BLISS-10 compiler.
Default for files of type .B10 and .BLI

/36-BLISS compiles the file using the BLISS-36 compiler.
Default for files of type .B36

/68-COBOL compiles the file using the COBOL-68 compiler.
Default for files of type .C68 or .68C

/74-COBOL compiles the file using the COBOL-74 compiler.
Default for files of type .C74 or .74C

/ABORT stops a compile if a fatal error is detected and returns your terminal to TOPS-20 command level.

/ALGOL compiles the file using the ALGOL compiler.
Default for files of type .ALG

/BINARY allows generation of an object (binary) file for each source file given. Use this switch to cancel a /NOBINARY switch.
Default

/COBOL compiles the file using the COBOL compiler, either COBOL-68 or COBOL-74, that your installation has stored in the file SYS:COBOL.EXE.
Default for files of type .CBL

/COMPILE forces compilation of the source file even if a current object file already exists. Use this switch along with the /LIST or /CREF switch to obtain listings when you have current object files. (See also the /NOBINARY switch.)

/CREF same as /CROSS-REFERENCE

/CROSS-REFERENCE
creates a file containing cross-reference information for each compilation. The filename is that of the object file; the file type is .CRF. Use the CREF command to obtain a listing of the file. (For COBOL files, this switch automatically produces a cross-reference listing.) See the TOPS-20 User Utilities Guide for information about the CREF program.

/DDT loads the DDT debugging program along with your object file.

/DEBUG produces an object file containing debugging information beyond what is usually inserted during a compilation. (For FORTRAN programs only, and only if you have not given the /OPTIMIZE switch.)

/FAIL compiles the file using the FAIL compiler.
Default for files of type .FAI

/FLAG-NON-STANDARD
indicates non-standard syntax in file.

/FORTRAN compiles the file using the FORTRAN compiler
Default in the absence of a standard source file type and a language switch

Default for files of type .FOR

/LANGUAGE-SWITCHES:"/switch(es)"
passes the specified switches to the compiler that will process the file(s) to which this switch applies. You must include the switches in double quotation marks (" ").

/LIBRARY same as /SEARCH.

/LIST prints a listing of the program in ASCII format; the name of this listing is the filename of the object file. The /CREF switch overrides /LIST when they both apply to the same file.

/MAC same as /MACRO

/MACHINE-CODE produces a file containing the generated machine code. The filename is that of the object file; the file type is .LST. For high-level languages.
Default

/MACRO assembles the file using the MACRO assembler.
Default for files of type .MAC

/MAP produces a loader map and stores it in the file object.MAP, where object is the name of the module containing the start address, or (if no start address) nnnLNK.MAP, where nnn is your job number.

/NOBINARY prevents generation of an object (binary) file. Use this switch along with /LIST or /CREF to allow these switches to take effect without producing a new object file.

/NOCREF same as /NOCROSS-REFERENCE.

/NOCOMPILE prevents compilation if the associated object file is current; otherwise it forces compilation. Cancels /COMPILE or /RELOCATABLE. See Characteristics - Compiling New Sources Only, below.
Default

/NOCROSS-REFERENCE
prevents the creation of a cross-reference file.
Default

/NODEBUG excludes special debugging information from your object file. (For FORTRAN programs only.)
Default

/NOFLAG-NON-STANDARD
prevents the flagging of nonstandard syntax in the file.
Default

/NOLIBRARY same as /NOSEARCH.

/NOLIST prevents a line printer listing of the program.
Default

/NOMACHINE-CODE
prevents generation of a file containing machine code.
Default

/NOOPTIMIZE prevents the generation of a globally optimized object file. (For FORTRAN programs only.)
Default

/NOSEARCH requires all modules in the object file library (the file accompanied by this switch in the command line) to be loaded, even if they are not called by your program. Cancels the /SEARCH switch.
Default

/NOSTAY stops the compiler from being placed in a background fork. Use when /STAY is set as a default for the compiler.

/NOSYMBOLS prevents a symbol table from being loaded along with the object file.

/NOWARNINGS prevents display of warnings for nonfatal errors.

/OPTIMIZE calls for generation of a globally optimized object file, that is, one that runs as quickly as possible. (For FORTRAN programs only, and only if you have not given the /DEBUG switch.)

/PASCAL compiles the file using the PASCAL compiler.
Default for files of type .PAS

/RELOCATABLE identifies the input file as an object file (regardless of its extension) and prevents compilation of the source file, forcing use of an existing object file even if the object file is out of date.
Default for files of type .REL

/SAIL compiles the file using the SAIL compiler.

Default for files of type .SAI

/SEARCH requires that the object file library (the file accompanied by this switch in the command line) be searched for modules called by your program or by a program subroutine. Only these modules are loaded, along with modules called from the system libraries, which are always searched.

/SIMULA compiles the file using the SIMULA compiler.
Default for files of type .SIM

/SNOBOL compiles the file using the SNOBOL compiler
Default for files of type .SNO

/STAY returns your terminal to TOPS-20 command level so that you can perform other work while the system continues to compile your program. You immediately receive the TOPS-20 prompt (@ or $), and can then issue any user command. Be careful not to send incorrect data to programs expecting terminal input. (See the CONTINUE command, Restrictions: Programs Competing for Terminal Input.)

This switch saves you from having to: issue a ^T to make sure the compiler has begun; give a ^C to halt compilation; and issue a CONTINUE /STAY command to remain at command level during compilation.

/SYMBOLS loads a symbol table along with the object file; helpful for debugging a program.
Default

/WARNINGS displays warnings for nonfatal errors.
Default

Characteristics

Compiling New Sources Only

The system usually compiles only those sources for which there are no current object files, that is, sources whose write dates are more recent than those of the object files of the same name.

However, sources for which you supply a new object filename are compiled even if there are current object files. You can always force compilation with the /COMPILE switch.

Default Switches Not Passed to Compiler

Only switches specified in a LOAD-class command are passed to the compiler; default switches are not passed. Instead, the system assumes that the defaults for the compiler are the same as the defaults for the LOAD-class command.

Using Standard File Types

If you specify source files with standard types (.FOR, .MAC, .CBL, or .ALG) in a COMPILE command, the system automatically calls the appropriate compiler when compilation is necessary. If you specify source files by filename only, the system searches your connected directory in the above order for a file of this name and a standard type. To compile programs from sources that have nonstandard file types, give a switch to indicate the proper compiler (/FORTRAN, /MACRO, /COBOL, or /ALGOL). A switch will take precedence over a standard file type if they indicate different languages. If no compiler is indicated with either a switch or a standard file type, the FORTRAN compiler is used.

Hints

Plus Signs Between Filespecs

If you give two or more filespecs separated by plus signs (+) as arguments to COMPILE, they are compiled together as if they were a single file. Their object module is stored under any filename given as the "object" argument of the command, or (if none) under the last filename in the group and file type .REL.

Indirect Files as Arguments

You can store the arguments (source and object filespecs, switches) of a COMPILE command in an indirect file, and specify them by typing an at sign (@) and its filespec as a COMPILE command argument.

Establishing Default Arguments with the SET Command

You can issue the SET DEFAULT COMPILE-SWITCHES command to set up default global arguments to the COMPILE command. Insert this SET command in your COMAND.CMD file to change your own defaults permanently.

Wildcards Illegal with COMPILE

The COMPILE command does not accept wildcard characters (* and %) in a file specification.

Effect on Memory

The COMPILE command clears any unkept forks from memory and loads the appropriate compiler.

Related Commands

LOAD, EXECUTE, and DEBUG	other LOAD-class commands for performing related functions

Examples

1. Compile a FORTRAN program.

```
@COMPILE RSD2.FOR
FORTRAN: RSD2
MAIN.
```

2. Do the same thing, using a switch to indicate the proper compiler. Use the /STAY switch to return immediately to TOPS-20 command level.

```
@COMPILE RSD2/FORTRAN/STAY
@PUSH

 TOPS-20 Command processor 7(1015)
@EDIT LOGIN.CMD
```

3. Create an indirect file using the EDIT editor. Use it to compile several programs, forcing a compilation of the last one and storing its object file under a new name.

```
@CREATE UPDATE.CMD
Input: UPDATE.CMD.1
00100   /COBOL FSTQ, SNDQ, THDQ, FTHQ/COMPILE ANNUAL
00200   $
*E

[UPDATE.CMD.1]
@COMPILE @UPDATE.CMD
COBOL:  DMN     [FSTQ.CBL]
COBOL:  DMN     [SNDQ.CBL]
COBOL:  DMN     [THDQ.CBL]
COBOL:  DMN     [FTHQ.CBL]
```

```
EXIT
@DIRECTORY,
@@CHRONOLOGICAL WRITE
@@REVERSE
@@
   PS:<LATTA>
 ANNUAL.REL.1
 THDQ.REL.1
 SNDQ.REL.1
 SNDQ.CBL
 FSTQ.CBL

Total of 5 files
```

4. Produce a cross-reference (.CRF) file for a FORTRAN program although you already have a current object file; prevent the generation of a new object program. Check for the output file.

```
@COMPILE /CREF/FORTRAN/COMPILE/NOBINARY RSD2
FORTRAN: RSD2
MAIN.
@DIRECTORY,
@@CHRONOLOGICAL WRITE
@@REVERSE
@@

   PS:<LATTA>
 RSD2.CRF.1
 017CRE.TMP.100017;T
 RSD2.REL.1
 RSD2.FOR

Total of 4 files
```

2.14 CONNECT

Connects your job to a directory.

Format

@CONNECT (TO DIRECTORY) **dev:<directory>**
PASSWORD:**password**

where:

dev:<directory> is the directory to which you want to connect.

Default dev: - your connected structure

Default <directory> - the directory (on the specified structure) of the same name as your connected directory

Default (if no arguments are given) - your log-in directory on the public structure

password is the password of the directory (not requested for your log-in directory or a directory to which you already have ownership or sufficient group rights).

Characteristics

Capabilities

Your capabilities (WHEEL, OPERATOR, SEMI-OPERATOR, CONFIDENTIAL) are associated with your log-in user name only. If you connect to a directory whose owner has Wheel capabilities, you do not gain these capabilities.

Hints

Obtaining Group Rights

You can obtain group rights equal to those of the owner of a directory by giving the ACCESS command instead of, or in addition to, CONNECT for that directory.

Restrictions

Features Not Affected

For some system features, CONNECT does not affect the directory used:

System Accounting

The SET ACCOUNT command allows arguments valid for your log-in user name only. Generally, charges for system use are made to your log-in user name.

Queue-class Commands

The Queue-class commands charge processing requests to your log-in user name only.

Related Commands

ACCESS	for obtaining group as well as ownership rights equal to those of the owner of a directory
MOUNT	for making a structure available for connecting and ensuring the continued availability of the structure

Examples

1. Connect to another user's directory.

```
@CONNECT <HOLLAND>
Password:___
```

2. Try to access a directory. Discovering that it is a files-only directory, connect to it instead.

```
@ACCESS <MANUALS>
?Directory is "files-only" and cannot be accessed
@CONNECT <MANUALS>
Password:___
```

3. Connect to another user's directory on a different file structure and then to your directory on that structure; then return to your log-in directory on PS:. Give INFORMATION JOB-STATUS commands as you go along to check which is your connected directory.

```
@INFORMATION JOB-STATUS
 Host AURORA
```

```
 Job 36, TTY207, User LATTA
 Account 341
@MOUNT STRUCTURE SNARK:
Structure SNARK: mounted
@CONNECT SNARK:<HOLLAND>
Password:___
@INFORMATION JOB-STATUS
 Host  AURORA
 Job 36, TTY207, User LATTA, SNARK:<HOLLAND>
 Account 341
@CONNECT <LATTA>
@INFORMATION JOB
 Host AURORA
 Job 36, TTY207, User LATTA, SNARK:<LATTA>
 Account 341
@CONNECT
@INFORMATION JOB-STATUS
 Host AURORA
 Job 36, TTY207, User LATTA
 Account 341
@DISMOUNT STRUCTURE SNARK:
Structure SNARK: dismounted
```

4. Connect to your directory on another structure and obtain your full rights to it. After giving an INFORMATION JOB-STATUS command to verify your connected directory, give a command that depends on these rights. Then return to your log-in directory on PS:.

```
@MOUNT STRUCTURE SNARK:
Structure SNARK: mounted
@ACCESS SNARK:
@CONNECT SNARK:
@INFORMATION JOB

 Host AURORA
 Job 36, TTY207, User LATTA, SNARK:<LATTA>
 Account 341
@INFORMATION DIRECTORY <LATTA.*>,
@@NAME-ONLY
@@
 Name SNARK:<LATTA.ALLEN>
 Name SNARK:<LATTA.BLAKE>
 Name SNARK:<LATTA.LAB-NOTES>
 Name SNARK:<LATTA.TESTS>
@END-ACCESS SNARK:<LATTA>
@CONNECT
@DISMOUNT STRUCTURE SNARK:
Structure SNARK: dismounted
```

2.15 CONTINUE

Continues execution of a fork that was halted.

Format

@**CONTINUE** (FORK) **argument /switch**

where:

argument — is the fork name or fork number.
Default - the current fork

/switch — is a keyword, chosen from the list below, indicating your choice of CONTINUE command options.

CONTINUE Command Switches

/BACKGROUND — keeps your terminal at TOPS-20 command level and continues execution of the program in a "background" fork. When the program attempts to do terminal input or output, it halts and displays the message [FORK-NAME wants the TTY].

/NORMALLY — restores your terminal to command level (if any) within the program.
Default

/STAY — keeps your terminal at TOPS-20 command level and continues execution of the program in a "background" fork. Output from the program is sent to the terminal and is intermixed with whatever output is currently displayed. When the program attempts to read from the terminal, it can randomly intercept input intended for the EXEC or another program. Therefore, use this switch with programs that, once started, do not request further terminal input.

Characteristics

Continuing a Noncurrent Fork

When you continue a noncurrent fork, by including the fork-name argument in a CONTINUE command, the specified fork becomes your current fork.

Hints

Stopping a Background Program

To stop a background program, give the FREEZE command.

Providing Input to a Background Program

A background program, continued with CONTINUE /BACKGROUND, sends the message [FORK-NAME wants the TTY] when it wants input. A background program, continued with CONTINUE /STAY, prints the program prompt, for example PASCAL>, when it wants input. To (provide input to a program that is running in a background fork, return to program command level with CONTINUE /NORMALLY. (Some programs require you to type an extra RETURN after CONTINUE /NORMALLY to display the program prompt). Then, type the required program input. (See below, Restrictions, Programs Competing for Terminal Input.)

Monitoring your Program

CONTINUE /STAY and CONTINUE /BACKGROUND, by keeping your terminal at TOPS-20 command level (EXEC), let you use TOPS-20 commands to monitor the progress of your program while it is running. Use the INFORMATION FORK-STATUS command to display the CPU time used and the kept and RUN status of each fork belonging to the current EXEC level. More commands for monitoring your programs are listed below in Related Commands.

Running Multiple Programs Simultaneously

To simultaneously run multiple programs in background forks or use commands that affect memory, use one or a combination of the following methods after placing a fork in the background with CONTINUE /STAY or CONTINUE /BACKGROUND:

PUSHing to an Inferior EXEC Level

Type the PUSH command to create an inferior EXEC level and a fresh copy of memory (address space). Then run another program and return to EXEC command level with CONTINUE /STAY or CONTINUE /BACKGROUND. The new program does not affect the background program since both programs are at a different EXEC level. However, see Restrictions below. PUSH to a new EXEC before running each new program.

KEEPing the Fork

Type the KEEP command to give the background fork a

"kept" status. (A kept fork is not cleared from memory when another program is loaded.) Then run another program and return to EXEC command level with CONTINUE /STAY or CONTINUE /BACKGROUND. KEEP each background fork before running another program. Check the status of your forks with INFORMATION FORK-STATUS.

Continuing Forks Using the Fork Name

You can continue a fork by typing the fork name as if it were the CONTINUE command. To function as the CONTINUE command, the fork must be "kept" with the KEEP or the SET PROGRAM KEEP command. For more information, refer to the descriptions of these commands.

More Information

The CONTINUE command is one of the TOPS-20 multiforking-class commands. For more information about multiforking, see the section named, Running Multiple Programs, in the TOPS-20 User's Guide.

Restrictions

Similar Programs Competing for Files

If you have two similar programs running simultaneously, they may try to access the same files at the same time (for example, temporary files labeled by job number, used by compilers). This may cause unpredictable situations to develop. To avoid the possibility, run different kinds of programs.

Programs Competing for Terminal Input

If you use CONTINUE /STAY to run a program in a background fork, the program can request input from the terminal while you are giving input to the EXEC or another program. This input can be randomly intercepted by the background program when it requests terminal input. Usually though, the EXEC or the current program receives the input.

When terminal input is intercepted by the background program, the program will usually type input error messages. To give input to the program, stop the program by typing two CTRL/Cs or the program's exit command. Then, if the background program is at a higher EXEC command level, give POP commands to return to the EXEC level that holds the background program. (POP terminates the current EXEC and erases programs in its memory.) Finally, give the CONTINUE

/NORMALLY command; this puts you at program command level so that you can give the requested input.

Remember, input is intercepted by the background program randomly. Therefore, you may have to type extra CTRL/Cs, program exit commands, and POPs. To reduce confusion about the direction of terminal input, it is recommended that you use CONTINUE /STAY only when you plan to work at the current EXEC level while a program runs in a background fork. You should also CONTINUE /STAY programs that simply end without requesting terminal input. Use CONTINUE /BACKGROUND when you plan to work at a lower EXEC level or at another program command level.

When a program started with CONTINUE /BACKGROUND requests terminal input, it sends the message, [FORK-NAME wants the TTY]. No input is taken by the background program until you return to program command level with CONTINUE /NORMALLY. You should CONTINUE /BACKGROUND programs that request terminal input.

Maintaining Access to Directories

While a fork is running in the background, use caution in using the CONNECT, ACCESS and END-ACCESS commands. Changing your directory access could leave the fork unable to reference certain files.

No I/O Control with Some Programs

Most programs read and write data to the terminal through standard input and output designators. Some programs however, use different methods of communicating with the terminal. Therefore, when you use /BACKGROUND and /STAY to control terminal input and output from a background fork, the input and output behavior of programs with nonstandard designators can be unpredictable.

Continued Programs Do Not Prompt for input

When you continue a program, the program continues from exactly where it was interrupted. If the program was waiting for input, it will simply continue to wait for input; it won't prompt you again. For example, assume you are running the DECmail/MS program and you press CTRL/C at the MS> prompt. Next, you CONTINUE MS. The cursor moves to the next line but no MS> prompt appears. This is because MS has continued to do the last thing it was doing when you interrrupted it with CTRL/C - waiting for a command at the MS> prompt. MS does not know that its prompt is no longer displayed before the cursor.

So, when you continue a program and nothing happens, consider what you were doing when you CTRL/C'd the program. If you were at the MS> prompt, type an MS command or, press RETURN again to redisplay the MS> prompt. If you had typed a portion of and MS command, press CTRL/R to redisplay the command. If you had typed a portion of a mail message, press CTRL/K to redisplay the message.

Effect on Memory and Terminal

The CONTINUE /NORMALLY command resumes processing the program in memory, and leaves your terminal at program command level (if any). The CONTINUE /BACKGROUND and CONTINUE /STAY commands resume processing the program in memory, but leave your terminal at TOPS-20 command level.

Related Commands

DETACH CONTINUE	for disengaging your current job from your terminal and continuing the program that the job is running
FORK	for changing the current fork
FREEZE	for halting a program in a background fork
INFORMATION FILE-STATUS	for monitoring files being written by your program
INFORMATION FORK-STATUS	for displaying the number and the status of each fork in your job
INFORMATION MEMORY-USAGE	for monitoring your program's use of memory
INFORMATION PROGRAM-STATUS	for monitoring your program's use of CPU time
KEEP	for giving a fork a kept status
PUSH	for obtaining a lower TOPS-20 command level (and a fresh copy of memory)
REENTER	for starting your current program at its alternate entry point (if any)

START	for starting your current program at the beginning
RESET, SET NAME, SET PROGRAM, UNKEEP	other multiforking-class commands for performing related functions

Examples

1. Display the fork status with the INFORMATION FORK-STATUS command. Notice that the arrow points to the current fork. Then, give the CONTINUE command to continue the program in the current, halted fork.

```
@INFORMATION FORK-STATUS
   EDIT (1): Kept, HALT at 6253, 0:02:54.4
=> DUMPER (2): HALT at 700304, 0:01:19.3
@CONTINUE
DUMPER>
```

2. Run the DSR program and then halt it by typing two CTRL/Cs. Give the CONTINUE /BACKGROUND command to continue DSR in a background fork and return to EXEC command level. Then, give the KEEP command so that you can load another program without clearing the running, background, DSR fork. Check the status of DSR with the INFORMATION FORK-STATUS command.

```
@RUN DSR
DSR>TEST.RNO
^C
@CONTINUE /BACKGROUND
@KEEP
[Keeping DSR]
@INFORMATION FORK-STATUS
=> DSR (1): Kept, Background, Running at 413160, 0:00:00.8
```

   Now begin editing a file with the EDIT program. During your editing session the system notifies you that the background fork wants input. To return to DSR command level, first exit the edit program. Then check the fork status with the INFORMATION FORK-STATUS command. Notice that DSR is in a terminal I/O wait state and that EDIT is now the current fork. Since the fork you want to continue is not the current fork, you must specify the fork name with the CONTINUE command. Now type CONTINUE DSR.

```
@EDIT COMAND.CMD
Edit: COMAND.CMD.2
*P
00100   SET DEFAULT PRINT /NOHEADER /NOTIFY:YES
00200   SET PROGRAM MS KEEP START
```

```
00300   SET PROGRAM HOST KEEP CONTINUE
00400   SET DEFAULT COMPILE-SWTICHES PAS /NOFLAG-NON-STANDARD
00500   INFO MAIL
00600   TAKE
*I350
00350   SET PROGRAM DUMPER KEEP CONTINUE
DSR>[DSR: wants the TTY]
*E

[COMAND.CMD.3]
@INFORMATION FORK-STATUS
=> EDIT (1): Kept, HALT at 6253, 0:00:51.4
   DSR (1): Kept, Background, TTY I/O wait at 4404426,
0:00:00.8
@CONTINUE DSR

DSR>
```

3. Begin editing a long file, giving the F (find) command to EDIT. Give a CTRL/C and then the M command to return to TOPS-20 command level. Give the CONTINUE /STAY command and then INFORMATION FILE-STATUS commands to check the progress of EDIT as it searches through the file. (Notice that the byte position shown in response to successive INFORMATION FILE-STATUS commands grows larger.) Finally, give the CONTINUE command to return to EDIT so you can give more EDIT program commands.

```
@EDIT DOC-PLAN.MEM
Edit: DOC-PLAN.MEM.1
*FABCD$
Yes? (Type H for help): M
@CONTINUE /STAY
@INFORMATION FILE-STATUS
 Connected to PS:<LATTA>,  JFNS:
 4   <LOADTEST>EDIT.EXE.4        Read, Execute
 3   EDIT-BUFFER.OUT.100046      Read, Write, 0.(7)
 2   DOC-PLAN.MEM.1      Read, 43520.(7)
 1   <SYSTEM>EXEC.EXE.153        Read, Execute

 Device assigned to/opened by this job: TTY222

@INFORMATION FILE-STATUS 2
 2   DOC-PLAN.MEM.1      Read, 112640.(7)
@INFORMATION FILE-STATUS 2
 2   DOC-PLAN.MEM.1      Read, 130560.(7)
@CONTINUE

*
```

4. Start compiling a long file. After compilation has begun, type two CTRL/Cs to stop the compilation and return to the EXEC command level. Use the CONTINUE /STAY command to resume compilation, and then PUSH to a new EXEC command level. Edit a text file at this lower level, then give the POP and CONTINUE commands to return to the compilation in progress. The compiler finishes, in this case, after you have done so.

```
@COMPILE DUMPER.MAC
MACRO:  DUMPER
^C
@CONTINUE /STAY
@PUSH

 TOPS-20 Command processor 7(55)
@EDIT PROFIL.TXT
Edit: PROFIL.TXT.2
*SAPRIL$JUNE$^:*
00100   JUNE 19, 1987
00500   JUNE 12
00750   JUNE 5
00900   JUNE 18
01400   JUNE 21
*E

[PROFIL.TXT.3}
@POP
@CONTINUE

EXIT
```

2.16 COPY

Creates a copy of a file.

Format

@COPY (FROM) **source filespec** (TO) **destination filespec,**
@@subcommand

where:

source filespec — is the specification of the file or device whose contents you want to copy.

destination filespec — is the specification of the file or device in which you want to store a copy of the file.
Default – same as source filespec but in your connected directory, if necessary using the next higher generation number

@@subcommand — means that after a final comma you can specify the mode and format of the transfer with one of the following subcommands:

COPY Subcommands

(when used with the paper tape reader
or paper tape punch – PTR: or PTP:)

ASCII — specifies that the file being copied is written in ASCII mode, with 36-bit words each consisting of five 7-bit bytes and a parity bit; the parity bit means that the eighth hole of the paper tape is never punched.

BINARY — specifies that the file being copied is composed of 36-bit words, each consisting of six 6-bit bytes with the seventh hole of the paper tape set always to 0 and the eighth hole set always to 1; causes a checksum calculation.

BYTE n — specifies that the byte size of the destination file is to be n (any decimal number). If you do not give the BYTE

subcommand, the destination file will have the same byte size as the source file. See also Hints - Viewing Display Screen Data below.

IMAGE — specifies that the file being copied is composed of 36-bit words, each consisting of one 8-bit byte; the 28 most significant bits are set to 0 on input and are lost on output.

IMAGE BINARY — same as BINARY, but lacking the checksum calculation.

COPY Subcommands

(when used with devices other than the paper tape reader or paper tape punch)

ASCII — specifies that the file being copied is written in ASCII mode, with 36-bit words each consisting of five 7-bit bytes and a parity bit; the parity bit means that the least significant bit is set to 0 on input and is lost on output.

BINARY — calls for a direct transfer of data in 36-bit bytes.

BYTE n — specifies that the byte size of the destination file is to be n (any decimal number). If you do not give the BYTE subcommand, the destination file will have the same byte size as the source file. See also Hints - Viewing Display Screen Data, below.

IMAGE — same as BINARY.

IMAGE BINARY — same as BINARY.

SUPERSEDE | ALWAYS | NEVER | NEWER | OLDER — sets the condition under which COPY overwrites the destination file of the same name.

ALWAYS — allows the source file to be copied to the destination file.
Default for COPY command

NEVER — does not copy the file if the destination file already exists.

OLDER — allows the source file to be copied to the destination file if

- no version of the destination file exists, or
- the generation number is less than or equal to the generation number specified in the destination file and the file's write date is older than the source file.

NEWER — same as OLDER except allows the copy if the file's write date is "newer" than the source file.

Output

As each file is copied, the system prints the specifications of the source and destination files and the word [OK]. The delay before you see this [OK] indicates how long it took to copy the file. If you use recognition on the destination file specification, the system prints, !New Generation!, !New File!, or !Superseding!, to indicate the status of disk files, or !OK!, if the file is copied to a non-disk device.

Characteristics

Optional Subcommands With Paper Tape

Each subcommand, when used to copy information from the paper tape reader (PTR:), specifies an interpretation of eight-bit bytes, represented as eight-hole lines on paper tape. When used with the paper tape punch (PTP:), each subcommand specifies a mapping of information to the eight-bit bytes of paper tape.

Optional Subcommands With Other Devices

Each subcommand can be used under particular conditions, for example, when transferring files over network facilities (using DCN: and SRV:),to specify the byte size of information being copied. In general, you can use COPY command subcommands whenever you need to specify the byte size of information being copied.

Hints

RENAME Faster Than COPY for Transferring Files

For moving a set of files from one directory to another on the same structure, the RENAME command is a faster and more efficient means than COPY. This is because RENAME only changes the file specifications; it does not copy the contents of the files. Also, a file transfer with the RENAME command leaves only one set of files, while a transfer with the COPY command leaves two sets: the original copies and the destination copies. The original copies are often unnecessary and must be deleted.

Using Devices as Source and/or Destination Filespecs

By specifying a device as the source and/or destination filespec, you can use the COPY command to transfer information between card- or paper-tape-handling devices, magnetic tape drives, line printers, terminals, or other output devices. However, the PLOT, PRINT, PUNCH and TYPE commands, and appropriate utility programs (such as DUMPER and EDIT), offer more flexibility for most applications.

Copying To or From TTY:

You can simulate the action of the CREATE command for creating files by copying from device TTY: to a new filespec, ending your input with a CTRL/Z; use CTRL/U, CTRL/R, CTRL/W, and the DELETE key to edit the current line of terminal input. You can simulate the action of the TYPE command for displaying files by copying from an existing filespec to device TTY:.

Viewing Display Screen Data

If you specify TTY: as the destination filespec and then give the BYTE 8 subcommand, characters in the source file will be sent literally to your terminal. Do this to examine special display screen data (for 8-bit ASCII files only).

Erasing the Contents of a File While Keeping the Filespec

You can erase the contents of a file by copying from device NUL: to the file. NUL: is a receptacle for unwanted program output and a supplier of null input.

Spooled Output Action

If you send information to output devices using the

COPY command, your request is processed according to the status of the SPOOLED-OUTPUT-ACTION parameter, which you set with the SET SPOOLED-OUTPUT-ACTION command.

Using Wildcards in Source and/or Destination Filespecs

You can use wildcard characters (* and %) in source and/or destination filespecs to copy many files at a time. Default values will be assumed for filespec fields you do not specify. Note that if you use wildcard characters to copy more than one source file into a single destination file on disk, the contents of each source file will appear in a different generation of the destination file; the highest generation will contain a copy of the last source file only. Use the APPEND command to put the contents of several files into a single file.

Specifying a New Account or Protection Number

The COPY command lets you specify the new file's protection number, and the account to which storage fees for it will be charged. Follow the new file specification with a semicolon (;) and the letter P before giving a new 6-digit protection number, and with a semicolon and the letter A before giving a new account (which must be valid for your user name). If you do not specify an account for a new file, it will take as a default the account you gave in your most recent LOGIN or SET ACCOUNT command. However, non-default protection numbers will be maintained for higher generations of existing files, unless you specify otherwise in the COPY command that creates that higher generation.

Restrictions

Copying Archived Files

You can make a copy of an archived file by specifying it as the first (or source) argument in a COPY command, and specifying a file of different name or type as destination. You can edit the new file, because it does not have archive status although it has the same contents as the original file. However, you cannot give the specification of an archived file as the second (or destination) argument of a COPY command, as this would replace the file's contents. If you attempt to do so, whatever source argument you supply will be copied into the next higher generation of the

archived file, leaving the archived file intact. And, if you include the generation number when specifying an archived file as the second argument of a COPY command, the command will fail.

Warning

Destroying the Previous Contents of Files

If you give a destination file specification that includes a generation number, the source file will be copied into that file, replacing any previous contents if that generation of the file already exists. Those contents cannot be recovered. But see Restrictions - Copying Archived Files, above.

Related Commands

APPEND	for adding information to a file or putting the contents of many files into a single file
RENAME	for changing only the specification of a file
SET SPOOLED-OUTPUT-ACTION	for changing the setting of the SPOOLED-OUTPUT-ACTION parameter, which determines when files copied to output devices are processed
DIRECTORY with the TIMES WRITE subcommand and VD	for displaying the date and time that the file was written

Examples

1. Make an extra copy of a file in your connected directory.

```
@COPY FORT.TXT BACKUP.TXT
 FORT.TXT.1 => BACKUP.TXT.3 [OK]
```

2. Copy a file from your directory into another user's directory, allowing the destination file to be labeled with default file specification (the source file specification).

```
@ACCESS <SARTINI>
Password:___
@COPY TEST1.CBL <SARTINI>
 TEST1.CBL.2 => <SARTINI>TEST1.CBL.2 [OK]
```

```
@END-ACCESS <SARTINI>
```

3. Use a wildcard character to copy several files from your directory on another structure to magnetic tape.

```
@ACCESS SNARK:
@COPY SNARK:NA*.TST MT2:
 SNARK:NACCESS.TST.2 => MT2:NACCESS.TST [OK]
 SNARK:NADVISE.TST.2 => MT2:NADVISE.TST [OK]
 SNARK:NAPPEND.TST.2 => MT2:NAPPEND.TST [OK]
 SNARK:NASSIGN.TST.2 => MT2:NASSIGN.TST [OK]
 SNARK:NATTACH.TST.2 => MT2:NATTACH.TST [OK]
@END-ACCESS SNARK:
```

4. Use the COPY command to create a short text file.

```
@COPY TTY: NEW-FILE.TXT
 TTY: => NEW-FILE.TXT.2

THIS FILE WAS CREATED USING THE COPY COMMAND.
^Z
@
@TYPE NEW-FILE.TXT
THIS FILE WAS CREATED USING THE COPY COMMAND.
```

5. Copy a file from your directory into another user's directory. Give the SUPERSEDE NEVER subcommand to cancel the COPY command if the other user already has a copy of the file.

```
@ACCESS <STEVENS>
Password:___
@COPY STATS.TXT <STEVENS>,
@@SUPERSEDE NEVER
@@
 STATS.TXT.1 => <STEVENS>STATS.TXT.3
%Not superseding current file
```

2.17 CREATE

Invokes your defined editor to create a file.

Format

@CREATE (FILE) **/switch(es) filespec**

where:

switches — are keywords, chosen from the list below, indicating your choice of CREATE command options.
Defaults are shown in the list of switches

filespec — is a specification for the file you want to create.
Default – the last file specification and associated switches you gave in a CREATE or EDIT command during the current terminal session

Summary of CREATE Command Switches (defaults in boldface)

NOTE

These switches are valid only if you have defined logical name EDITOR: to be the EDIT program.

/BAK
/C128
/C64
/DECIDE
/DPY
/EXPERT
/INCREMENT:n **Default** n - 100
/ISAVE:n
/LOWER
/M33
/M37
/NOBAK
/NODECIDE
/NONSEPARATORS
/NONUMBER
/NOVICE
/NUMBER
/OLD
/OPTION:name
/PLINES:n **Default** n - 16

/R	
/READONLY	
/RONLY	
/RUN:filespec	**Default** file type - .EXE
/SAVE:n	
/SEPARATORS	
/SEQUENCE	
/START:n	**Default** n - argument of /INCREMENT switch
/STEP:n	**Default** n - 100
/UNSEQUENCE	
/UPPER	
/WINDOW:n	**Default** n - 10

CREATE Command Switches

/BAK causes an unedited copy of the file to be saved at the end of an editing session under specification name.Qyp, where name.typ is the file's original specification.
Default

/C128 calls for a 128-character alphabet, allowing insertion of control characters in an alternate format. See the TOPS-20 EDIT Reference Manual for details.

/C64 calls for a 64-character alphabet, disallowing use of an alternate format for insertion of control characters.
Default

/DECIDE lets you decide whether to accept or reject each change caused by the operation of the S (substitute) command of the EDIT program.

/DPY has no effect in the current monitor.

/EXPERT tells the EDIT program that you need only abbreviated error messages, and fewer warnings and reminders.

/INCREMENT:n specifies the value to add to each line number of the file to obtain the next line number.
Default n - 100

/ISAVE:n instructs the EDIT program to update the backup file of specification name.Qyp after every n lines you insert.

/LOWER — specifies that all alphabetic characters you type should be considered lowercase characters; give uppercase characters by preceding the corresponding lowercase character with a single quotation mark (').

/M33 — has no effect in the current monitor.

/M37 — has no effect in the current monitor.

/NOBAK — prevents an unedited copy of the file from being saved at the end of an editing session under specification name.Qyp, where name.typ is the file's original specification.

/NODECIDE — ensures the automatic operation of the S (substitute) command of the EDIT program.
Default

/NONSEPARATORS — specifies that the characters . (period), $ (dollar sign), and % (percent sign) are ordinary textual characters and not field delimiters (separators) in the accompanying file.
Default

/NONUMBER — suppresses the printing of line numbers with each line of a file.

/NOVICE — tells the EDIT program that you want to see complete error messages and all appropriate warnings; opposite of /EXPERT switch.
Default

/NUMBER — prints a line number for each line of the file.
Default

/OLD — causes the first backup file to be saved under the specification name.Zyp, where name.typ is the file's original specification.

/OPTION:name — sets any EDIT switches contained in lines of the SWITCH.INI file in your log-in directory labeled with name (of 6 or fewer characters). See the TOPS-20 EDIT Reference Manual for more information about SWITCH.INI files.

/PLINES:n — specifies how many lines to print in response to each P (print) command of the EDIT program.
Default n - 16

/R — same as /READONLY.

/READONLY — prevents any changes to the file during the current session of the EDIT program, i.e., makes it a read-only session. This switch cannot be given in the SWITCH.INI file.

/RONLY — same as /READONLY

/RUN:filespec — specifies an executable program to be run when you end the current session of the EDIT program with the G command.
Default file type - .EXE

/SAVE:n — instructs the EDIT program to update the backup file (of specification name.Qyp) after every n EDIT program commands that modify the file.

/SEPARATORS — notifies the EDIT program that the characters . (period), $ (dollar sign), and % (percent sign), are not ordinary textual characters but are field separators in the accompanying file.

/SEQUENCE — tells the EDIT program not to strip the line numbers from the file when the EDIT session ends.
Default

/START:n — specifies the first line number for the EDIT program to use when numbering the file.
Default n - argument of /INCREMENT switch

/STEP:n — same as /INCREMENT

/UNSEQUENCE — tells the EDIT program to strip the line numbers from the file when the EDIT session ends.

/UPPER — specifies that all alphabetic characters you type should be considered uppercase characters; give lowercase characters by preceding the corresponding uppercase character with a single quotation mark (').
Default

/WINDOW:n — specifies the number n (between 10 and 99) of pages to be held in memory during the EDIT session.
Default n - 10

Characteristics

Input Mode and Edit Mode

The CREATE command runs the EDIT system program, first in Input mode and then in Edit mode. (However, see also Special Cases, below.) Input mode automatically begins each line with a line number (unless you have given the /NONUMBER switch), and allows you to put any alphabetic or numeric information into the file. When you have finished doing this and press the ESCAPE key, the EDIT program puts you into Edit mode and prompts you with an asterisk (*), just as if you had typed the EDIT command with the specifications of the newly-created file as argument. If you want to save the file in its present state, give the E (for end) command to the EDIT program. Otherwise, you can give any other EDIT command to change or add to the file before saving it.

Hints

Saving Backup Files Periodically

Give the /ISAVE:n switch to save an updated copy of the file you are creating after every n lines inserted. Then you will lose only a few lines of input in the event of a system failure. The similar /SAVE:n switch is useful for the CREATE command only in Edit mode, where it saves an updated copy of the file after every n EDIT program commands that modify the file.

SWITCH.INI File

If there is a group of CREATE command switches that you always or often use with CREATE or EDIT commands, put them into a file of specification SWITCH.INI in your log-in directory, in a line of that file beginning with EDIT:abc, where abc is any set of characters you choose to identify the line. Then if you include the single switch /OPTION:abc when you give a CREATE or EDIT command, all these switches will be in effect.

Further Information

For more information about the EDIT program, see the TOPS-20 EDIT Reference Manual.

Special Cases

Using an Editor Other than EDIT

The CREATE, EDIT, and PERUSE command descriptions in this manual assume that these commands call on the EDIT program for their action. If your job uses another editing program,

for example EDT, the switches and examples shown here will not be applicable.

The editor used by CREATE, EDIT, and PERUSE is specified by logical name EDITOR:, so you can find out the name of this program by giving the command, INFORMATION LOGICAL-NAMES EDITOR:. The job-wide definition (if any) will be given first, followed by the system-wide definition; the job-wide definition prevails if both exist. If the definition of EDITOR: is SYS:EDIT.EXE, the CREATE, EDIT and PERUSE commands will function as described in this manual. Otherwise, you must consult the appropriate manual (for example, the EDT-20 Reference Manual) for information.

You can use the DEFINE command to define logical name EDITOR: to be any editing program available at your installation. Then this editor will be in effect when you give the CREATE or EDIT command.

Effect on Memory

The CREATE command clears any unkept forks from memory, then loads the editor program defined by the logical name EDITOR.

Related Commands

DIRECTORY-class commands	for getting lists of existing files
EDIT	for modifying existing files
PERUSE	for reading existing files (same as EDIT/READONLY)

Examples

1. Create a file.

```
@CREATE FILE1.TXT
Input: FILE1.TXT.1
00100   !THIS IS A SHORT TEXT FILE.
00200   $
*E

[FILE.TXT.1]
```

2. Create and edit (using the P and R commands to the EDIT system program) another file.

```
@CREATE FILEB.TXT
Input:FILEB.TXT.1
00100   !THIS IS ANOTHER SHORT TEXT FILE.
00200   $
*P
00100   !THIS IS ANOTHER SHORT TEXT FILE.
*R100
00100   !THIS IS A SECOND TEXT FILE.
00200   $
1 Lines (00100/1) deleted
*P
00100   !THIS IS A SECOND TEXT FILE.
*E

[FILEB.TXT.1]
```

3. Create, then execute, a FORTRAN program.

```
@CREATE FILEE.FOR
Input: FILEE.FOR.1
00100   C       THIS IS A SHORT TEST PROGRAM.
00200           TYPE 101
00300   101     FORMAT ( ' THIS IS ONLY A FORTRAN TEST.')
00400           END
00500
*E

[FILEE.FOR.1]
@EXECUTE FILEE.FOR
FORTRAN: FILEE
MAIN.
LINK:   Loading
[LNKXCT FILEE Execution]

THIS IS ONLY A FORTRAN TEST.
```

```
END OF EXECUTION.
CPU TIME: 0.04  ELAPSED TIME: 0.44
EXIT
```

2.18 CREF

Runs the CREF program, which produces cross-reference listings from files of type .CRF.

Format

@CREF destination-filespec=source-filespec /switch(es)

where:

destination-filespec — is the name of the file or device to which you want to send the processed contents of the .CRF file.
Default - LPT:

source-filespec — is the name of the .CRF file you want to process.
Default - the names of all files of type .CRF produced during the current terminal session

/switch — is one or more keywords from the following list.

CREF Program Switches

/A — Advances magnetic tape reel by one file. You can type this switch more than once in the command string.

/B — Backspaces magnetic tape reel by one file. You can type this switch more than once in the command string.

/C — Cancels the processing of any switches in your SWITCH.INI file.

/D — Restores the processing of any default switches in your SWITCH.INI file.

/H — Types the CREF help file. /H is illegal in a SWITCH.INI file.

/K — Suppresses Regular Symbol Table in the CREF listing.

/M — Suppresses OPDEF/Macro Table in the CREF listing.

/O Includes the Op Code Table in the CREF listing.

/P Preserves an input file with the file type .CRF or .LST. These types of input files are normally deleted.

/R Requests the line number at which the CREF listing is to start. CREF types out "RESTART LISTING AT LINE:", after which you type the line number and press RETURN. If you use an indirect file, CREF looks for the number in the indirect file. /R is most useful for physical (non-spooled) line printers, and is illegal in a SWITCH.INI file.

/S Suppresses the program listing and lists only the tables you select.

/T Skips to the logical end of the magnetic tape.

/W Rewinds the magnetic tape.

/Z Zeroes the DECtape directory. This is a historical switch, and is illegal.

Characteristics

Current .CRF Files

If you have files of type .CRF produced by LOAD-class commands during the current terminal session, the unmodified command CREF produces listings of them and deletes the files. By supplying an argument of the form shown in the Format section above, you can copy the listing for a current .CRF file to another file or device. To run the CREF program yourself when you have current .CRF files, give the command R CREF instead.

.CRF Files From a Previous Session

If your only files of type .CRF (created by the CREF program or one of the LOAD-class commands) were produced during a previous terminal session, the command CREF puts your terminal at command level in the CREF program, symbolized by an asterisk (*). Thus it is equivalent to the command R CREF in this case. See Hints - Further Information, below, for advice on how to proceed.

Hints

Producing .CRF Files

You can produce cross-reference files by including the /CREF switch in any LOAD-class command that actually causes a compilation (i.e., is not prevented from doing so by a /RELOCATABLE switch or by the existence of current object files).

Preserving .CRF Files After Processing

Give the /P switch immediately after the CREF command to preserve .CRF files. Ordinarily they are deleted after being sent to an output device or copied into another file.

Further Information

For a brief on-line description of the CREF program, give the HELP CREF command. For more detailed information, see the TOPS-20 User Utilities Guide.

Effect on Memory and Terminal

The CREF command replaces the contents of any unkept forks in memory with the CREF program and leaves your terminal at TOPS-20 command level, or at command level within CREF (denoted by an asterisk prompt [*]).

Related Commands

LOAD-class commands for producing .CRF files

Examples

1. Give the CREF command to obtain a listing of your .CRF file.

```
@CREF
CREF:   TESTF1
```

2. Compile two FORTRAN programs, using the /CREF switch to produce .CRF files. Then give the CREF command to obtain listings of these, and use the /P switch to preserve the .CRF files.

```
@COMPILE /CREF TESTF1.FOR, TESTF2.FOR
FORTRAN: TESTF1
MAIN.
FORTRAN: TESTF2
```

```
MAIN.
@CREF/P
CREF:   TESTF1
CREF:   TESTF2
```

3. Determine what .CRF files you have, then mount a tape. Give the CREF command, and once within the CREF program, have the cross-reference listing produced from one of these files copied onto tape. (The .CRF files are not processed automatically when you give the CREF command because they were produced during a previous terminal session.)

```
@DIRECTORY *.CRF

   PS:<LATTA>
 TESTM1.CRF.2
 TESTM2.CRF.1

 Total of 2 files
@MOUNT TAPE CRFMAC:/WRITE-ENABLED
[Mount Request CRFMAC Queued, Request-ID 128]
[Tape set CRFMAC, volume CRFMAC mounted]
[CRFMAC defined as MT3:]
@CREF

*MT3:=TESTM2
[CRFXKC 4K core]
*^C
@DISMOUNT TAPE CRFMAC:
[Tape dismounted, logical name CRFMAC: deleted]
```

2.19 CSAVE

Makes a non-sharable copy of the program in memory and stores it in a file, in compressed executable format.

Format

@**CSAVE** (ON FILE) **filespec** (WORDS FROM) **loc1** (TO) **loc2, loc3 loc4,...**

where:

filespec — is the file specification under which you want to store the program.

Default filespec - program name.EXE

loc1 loc2, loc3 loc4, ... — are pairs of octal numbers or symbolic expressions that specify the span(s) of memory locations you want to save.

Default loc1 loc2 - 20 to last location occupied by program

Caution

Inefficiency of CSAVE Compared to SAVE

The CSAVE command saves in a compressed-formatted file whatever program the system finds in memory. When the file is returned to memory, this format prevents other users from sharing the in-memory copy of the file. Therefore you should ordinarily use the SAVE command instead for storing programs in executable format.

Related Commands

GET — for putting a saved file into memory

LOAD — for putting source or output files into memory

RUN — for running executable programs

SAVE — usual command for saving programs in executable format

Examples

1. Save your currently loaded program in compressed executable format.

```
@CSAVE
 DMN.EXE.1 SAVED
```

2. Mount a magnetic tape set. Then load an ALGOL program and save it in three places in executable format: once in a disk file under the same filename, again in a disk file under a new filename, and once on magnetic tape.

```
@MOUNT TAPE TAPBAK: /WRITE-ENABLED
[Mount Request TAPBAK Queued, Request-ID 140]
[Tape set TAPBAK, volume TAPBAK mounted]
[TAPBAK defined as MT2:]
@LOAD TESTA1
LINK:   Loading

EXIT
@CSAVE
 TESTA1.EXE.1 Saved
@CSAVE BAK
 BAK.EXE.1 Saved
@CSAVE MT2:
 MT2:BAK Saved
@DISMOUNT TAPE TAPBAK:
[Tape dismounted, logical name TAPBAK: deleted]
```

2.20 DAYTIME

Displays the current day, date, and time on your terminal.

Format

@DAYTIME

Hints

Using DAYTIME

The DAYTIME command, which does not require you to be logged in, lets you check the system's clock against your own. If you are saving the output from a hard-copy terminal, use this command to make a record of the date and time.

Examples

1. Give the DAYTIME command.

```
@DAYTIME
Friday, April 20, 1984 09:21:19
```

2.21 DDT

Loads or merges a debugging program into memory (unless one is already there), then starts it.

Format

@DDT/switch(es)

where:

/switch is one or more of the following:

/OVERLAY	allows pages of the DDT program to be loaded over pages occupied by the existing program in memory
/USE-SECTION:n	specifies the memory section (from 0 to 37 octal) into which the debugging program is to be loaded, run, or merged

Characteristics

If a Debugging Program is Already Loaded

If you have already loaded a debugging program into memory along with your program, the DDT command starts the debugging program.

If Your Program, But Not a Debugging Program, is Already Loaded

If a program containing symbols is in memory without a debugging program, the DDT command merges SYS:XDDT.EXE into memory, then starts this debugging program. However, if some of XDDT's pages include some of the same pages as the existing program in memory, then the DDT program is not placed into memory, and you receive the error message, "?Illegal to overlay existing pages." To force the pages to be overlaid, reissue the DDT command using the /OVERLAY switch.

If There is No Current Program

If you do not have a program in memory, or if no program in memory is in the current fork, or if your program does not contain symbols, the DDT command puts SYS:XDDT.EXE into memory and starts it.

Compatibility with Previous DDT Versions

The UDDT program run by the DDT command has been replaced by the XDDT program. For compatibility with programs that reference UDDT, a stub program named UDDT references the XDDT program.

Hints

Using DDT to Create a Program

You can use DDT to begin typing instructions directly into memory, without first putting the instructions into a file for later compilation and loading. Give a RESET . (period) command to clear the current fork, then the DDT command. This will load the SYS:XDDT.EXE program. Then you can give commands within XDDT to create your own program. When using the XDDT program, you can use all the symbols in the system parameter file MONSYM.MAC.

See the TOPS-20 Monitor Calls Reference Manual for more information about MONSYM.MAC. This method of writing a program is most useful for testing special cases, or for learning to use TOPS-20 monitor calls.

Special Cases

Using COBDDT

If you put COBDDT into memory along with a COBOL program, the DDT command starts the UDDT program, not COBDDT. Use the REENTER command to start COBDDT in this case.

Effect on Memory and Terminal

The DDT command merges the SYS:UDDT.EXE program into the current fork and starts it, or loads and starts SYS:XDDT.EXE. If you have already loaded a debugging program, the DDT command starts this program.

Related Commands

DEBUG	for loading your program along with a particular debugging program (such as FORDDT or COBDDT).
FORK	for selecting the current fork.

INFORMATION MEMORY-USAGE
for displaying the numbers of pages occupied by the program in memory.

Examples

1. Give the DDT command to begin debugging a program in the current fork in memory.

```
@DDT DDT
```

2. Give the DEBUG command to debug a FORTRAN program; type a CTRL/C to return to TOPS-20 command level so you can find out the current load averages and number of jobs for the system. Return to your debugging program (FORDDT in this case) by giving the DDT command.

```
@DEBUG TESTF1
LINK:   Loading
[LNKDEB FORDDT Execution]

STARTING FORTRAN DDT

>> ^C
@SYSTAT SYSTEM
 Fri 20-Apr-79 13:50:01  Up 36:47:55
 35+14 Jobs   Load av (class 0)   0.72   0.81   1.33

@DDT

STARTING FORTRAN DDT

>> START

THIS IS A TEST.

END OF EXECUTION
CPU TIME: 0.04  ELAPSED TIME: 0.33
EXIT
@INFORMATION MEMORY-USAGE

 66. pages, Entry vector loc 0 len 254000
0-12     Private   R, W, E
400      Private   R, W, E
401-466  <SUBSYS>FOROTS.EXE.3  3-70   R, CW, E
```

3. Display the programs in memory with the INFORMATION FORK-STATUS command. Then use the FORK command to make the CLIP program the new current fork. Verify this with INFORMATION FORK-STATUS and then merge the UDDT program with the CLIP program.

```
@INFORMATION FORK-STATUS
    EMACS (1): Kept, HALT at 50340, 0:00:03.6
    CLIP (2): Kept, HALT at 70363, 0:00:00.2
```

```
 => UNITS (3): HALT at 162, 0:00:00.9
@FORK CLIP
@INFORMATION FORK-STATUS
    EMACS (1): Kept, HALT at 50340, 0:00:03.6
=>  CLIP (2): Kept, HALT at 70363, 0:00:00.2
    UNITS (3): HALT at 162, 0:00:00.9
@DDT
DDT
```

2.22 DEASSIGN

Releases a device from your job and places the device back in the pool of available devices.

Format

@**DEASSIGN** (DEVICE) **dev:**

where:

dev: is the name of the device you want to deassign; an asterisk (*) deassigns all devices (except your log-in terminal) assigned to your job. The colon after the device name is optional.

Restrictions

Open Files

The DEASSIGN command will not deassign a device that is accessing an open file. An error is generated, and the device is not deassigned until that file is closed or until you log out. When you log out, all devices are deassigned.

Related Commands

ASSIGN	for assigning a particular device to your job
INFORMATION AVAILABLE DEVICES	for finding out which devices are available, and which ones have already been assigned to your job
MOUNT	for mounting a structure or magnetic tape and assigning the first available disk drive or tape drive to your job

Examples

1. Deassign a device (in this case a card reader).

   @DEASSIGN PCDR0:

2. Find out which devices are assigned to your job, then deassign all of these. Verify that this was done. (Note that your terminal, in this case TTY222:, is never deassigned.)

```
@INFORMATION AVAILABLE DEVICES
Devices available to this job:
 DSK, PS, SNARK, PACK, FTN20, MTA2, MT0, LPT, LPT0, LPT1
 CDR, PCDR0, CDP, FE1-15, PTY15-61, NUL, PLT, PLT0, DCN
 SRV
Devices assigned to/opened by this job: MTA2, MT0, PCDR0
 TTY222, PTY15
@DEASSIGN *
@INFORMATION AVAILABLE DEVICES
Devices available to this job:
 DSK, PS, SNARK, PACK, FNT20, MTA2, LPT, LPT0, LPT1
 CDR, PCDR0, CDP, FE1-15, PTY15-61, NUL, PLT, PLT0
 DCN, SRV
Devices assigned to/opened by this job: TTY222
```

2.23 DEBUG

Loads your program into memory along with a debugging program, compiling the source file first if necessary. Then it starts the debugging program.

Format

@DEBUG (FROM) **/switch(es) source/switch(es) object,...**

where:

switches — are keywords chosen from the list below, indicating your choice of DEBUG command options. They have different effects depending on their position in the command line: placed before all files in the command, they act as defaults for all; otherwise, they affect only the nearest preceding file.
Defaults are shown in the list of switches

source — is the file specification of the source program. The filename must be of 6 or fewer characters, and the file type of 3 or fewer characters; you cannot use a generation number. This argument is not necessary if you supply an object filespec.

object — is the file specification of the object program. The filename must be of six or fewer characters, and the file type must be .REL; you cannot use a generation number. This argument is not necessary if you supply a source filespec.
Default (if you give neither source nor object filespecs) - last filespecs and associated switches you gave in a LOAD-class command

Summary of DEBUG Command Switches (defaults in boldface)

/10-BLISS
/36-BLISS
/68-COBOL
/74-COBOL
/ABORT
/ALGOL
/BINARY
/COBOL
/COMPILE
/CREF
/CROSS-REFERENCE

/DDT
/DEBUG
/FAIL
/FLAG-NON-STANDARD
/FORTRAN
/LANGUAGE-SWITCHES:"/switch(es)"
/LIBRARY
/LIST
/MAC
/MACHINE-CODE
/MACRO
/MAP
/NOBINARY
/NOCOMPILE
/NOCREF
/NOCROSS-REFERENCE
/NODEBUG
/NOFLAG-NON-STANDARD
/NOLIBRARY
/NOLIST
/NOMACHINE-CODE
/NOOPTIMIZE
/NOSEARCH
/NOSTAY
/NOSYMBOLS
/NOWARNINGS
/OPTIMIZE
/PASCAL
/RELOCATABLE
/SAIL
/SEARCH
/SIMULA
/SNOBOL
/STAY
/SYMBOLS
/WARNINGS

DEBUG Command Switches

/10-BLISS compiles the file using the BLISS-10 compiler.
Default for files of type .B10 and .BLI

/36-BLISS compiles the file using the BLISS-36 compiler.
Default for files of type .B36

/68-COBOL compiles the file using the COBOL-68 compiler.
Default for files of type .C68

/74-COBOL compiles the file using the COBOL-74 compiler.
Default for files of type .C74

/ABORT stops a compile if a fatal error is detected and returns your terminal to TOPS-20 command level.

/ALGOL compiles the file using the ALGOL compiler.
Default for files of type .ALG

/BINARY allows generation of an object (binary) file for each source file given.
Default

/COBOL compiles the file using the COBOL compiler, either COBOL-68 or COBOL-74, that your installation has stored in the file SYS:COBOL.EXE.
Default for files of type .CBL

/COMPILE forces compilation of the source file even if a current object file already exists. Use this switch along with a /LIST or /CREF switch to obtain listings when you have current object files.

/CREF same as /CROSS-REFERENCE.

/CROSS-REFERENCE
creates a file containing cross-reference information for each compilation. The filename is that of the object file; the file type is .CRF. Use the CREF command to obtain a listing of the file. (For COBOL files, this switch automatically produces a cross-reference listing.) See the TOPS-20 User Utilities Guide for more information about the CREF program.

/DDT loads the DDT debugging program along with your object file.

/DEBUG produces an object file containing debugging information beyond what is usually inserted during compilation. (For FORTRAN programs only, and only if you have not given the /OPTIMIZE switch).

/FAIL compiles the file using the FAIL compiler.
Default for files of type .FAI

/FLAG-NON-STANDARD
indicates nonstandard syntax in file.

/FORTRAN compiles the file using the FORTRAN compiler.
Default in the absence of a standard source file type and a language switch
Default for files of type .FOR

/LANGUAGE-SWITCHES:"/switch(es)"
passes the specified switches to the compiler that will process the file(s) to which this switch applies. You must include the switches in double quotation marks (" ").

/LIBRARY same as /SEARCH.

/LIST prints a line printer listing of the program in ASCII format; the name of this listing is the filename of the object file. The /CREF switch overrides /LIST when they both apply to the same file.

/MAC same as /MACRO.

/MACHINE-CODE produces a file containing the generated machine code. The filename is that of the object file; the file type is .LST. For high level languages.
Default

/MACRO assembles the files using the MACRO assembler.
Default for files of type .MAC

/MAP produces a loader map and stores it in the file object.MAP, where object is the name of the module containing the start address; or (if no start address) nnnLNK.MAP, where nnn is your job number.

/NOBINARY prevents generation of an object (binary) file. Use this switch along with /LIST or /CREF to allow these switches to take effect without producing a new object file.

/NOCOMPILE prevents compilation if the associated object file is current; otherwise it forces compilation. Cancels the /COMPILE or /RELOCATABLE switch.
Default

/NOCREF same as /NOCROSS-REFERENCE.

/NOCROSS-REFERENCE
prevents the creation of a cross-reference file.

Default

/NODEBUG excludes special debugging information from your object file. (For FORTRAN programs only.)

/NOFLAG-NON-STANDARD
prevents a line printer listing of a cross-reference file.
Default

/NOLIBRARY same as /NOSEARCH.

/NOLIST prevents a line printer listing of the program.
Default

/NOMACHINE-CODE prevents generation of a file containing machine code.
Default

/NOOPTIMIZE prevents the generation of a globally optimized object file (for FORTRAN programs only).
Default

/NOSEARCH requires all modules in the object file library (the file accompanied by this switch in the command line) to be loaded even if they are not called by your program. Cancels the /SEARCH switch.
Default

/NOSTAY stops the compiler from being placed in a background fork. Use when /STAY is set as a default for the compiler.

/NOSYMBOLS prevents a symbol table from being loaded along with the object file.

/NOWARNINGS prevents display of warnings for nonfatal errors.

/OPTIMIZE calls for generation of a globally optimized object file, that is, one that runs as quickly as possible. (For FORTRAN programs only, and only if you do not also give the /DEBUG switch).

/PASCAL compiles the file using the PASCAL compiler.
Default for files of type .PAS

/RELOCATABLE identifies the input file as an object file (regardless of its extension) and prevents compilation of the source file, forcing use of

an existing object file even if the object file is out of date.
Default for files of type .REL

/SAIL — compiles the file using the SAIL compiler.
Default for files of type .SAI

/SEARCH — requires that the object file library (the file accompanied by this switch in the command line) be searched for modules called by your program or by a program subroutine. Only these modules are loaded, along with modules called from the system libraries, which are always searched.

/SIMULA — compiles the file using the SIMULA compiler.
Default for files of type .SIM

/SNOBOL — compiles the file using the SNOBOL compiler.
Default for files of type .SNO

/STAY — returns your terminal to TOPS-20 command level so that you can perform other work while the system continues executing the DEBUG command. You immediately receive the TOPS-20 prompt (@ or $) and can then issue any user command. Be careful not to send incorrect data to programs expecting terminal input. (Refer to the CONTINUE command, Restrictions: Programs competing for terminal input.)

This switch saves you from having to: issue a ^T to make sure the debugger has begun; give a ^C to halt debugging; and issue a CONTINUE /STAY command to remain at command level during debugging.

/SYMBOLS — loads a symbol table along with the object file; helpful for debugging a program.
Default

/WARNINGS — displays warnings for nonfatal errors.
Default

Characteristics

Compiling New Sources Only

Before debugging programs, the system ordinarily compiles any source (and only those sources) whose write date is more recent than that of the object file of the same name. You

can override this action with the /COMPILE or /RELOCATABLE switch. Note that the DDT debugging program is used when /RELOCATABLE prevents a new compilation.

Default Switches Not Passed to Compiler

Only switches specified in a LOAD-class command are passed to the compiler; default switches are not passed. Instead, the system assumes that the defaults for the compiler are the same as the defaults for the LOAD-class command.

Using Standard File Types

If you specify source files with standard types (.FOR, .MAC, .CBL, or. ALG) in a DEBUG command, the system automatically calls the appropriate compiler when compilation is necessary. If you specify source files by filename only, the system searches your connected directory in the above order for a file of this name and a standard type. To debug programs from sources that have nonstandard file types, give a switch to indicate the proper compiler (/FORTRAN, /MACRO, /COBOL, or /ALGOL). A switch will take precedence over a standard file type if they indicate different languages. If no compiler is indicated with either a switch or a standard file type, the FORTRAN compiler is used.

Name of Debugging Program Loaded by DEBUG

Ordinarily the DEBUG command causes the appropriate debugging program to be loaded along with your program (FORDDT with FORTRAN programs. COBDDT with COBOL programs, DDT with MACRO and ALGOL programs). Use the /DDT switch to specify that DDT be used.

Hints

Commas Between Filespecs

If you give two or more filespecs separated by commas as arguments to DEBUG, the loaded programs exist in memory at the same time and will act as a single program. You can use this feature to substitute one module for another under varying conditions or for different applications.

Plus Signs Between Filespecs

If you give two or more filespecs separated by plus signs (+) as arguments to DEBUG, they are treated as a single file by compilers. Their object module is stored under any filename given as the "object" argument of the command, or

(if none) under the last filename in the group and file type .REL.

Indirect Files as Arguments

You can store the arguments (source and object filespecs, switches) of a DEBUG command in an indirect file, and specify them by typing an at sign (@) and its filespec as a DEBUG command argument.

Establishing Default Arguments with the SET Command

You can issue the SET DEFAULT COMPILE-SWITCHES command to set up default global arguments to the DEBUG command. Insert this SET command in your COMAND.CMD file to change your own defaults permanently.

Including all FORTRAN Debugging Information

If you are debugging a FORTRAN program and you wish to examine line numbers or DO loops, or use statement tracing or array dimension checking, give the /DEBUG and /COMPILE switches with the DEBUG command to include the necessary information.

Running LINK Directly

The DEBUG command automatically runs LINK, the system's loader program, but if you require control of the loading process you can run LINK directly. See the TOPS-20 LINK Reference Manual.

Wildcards Illegal with DEBUG

The DEBUG command does not accept wildcard characters (* and %) in a file specification.

Effect on Memory

The DEBUG command clears any unkept forks from memory, loads the appropriate compiler if necessary, then loads your program and a compatible debugging program.

Related Commands

COMPILE, LOAD, and EXECUTE	other LOAD-class commands for performing related functions
DDT	for loading and starting the DDT

debugging program, or for starting the debugging program you have already loaded

Examples

1. Debug a FORTRAN program.

```
@DEBUG FORT.FOR
FORTRAN: FORT
MAIN.
LINK:   Loading
[LNKDEB FOR DDT Execution]
STARTING FORTRAN DDT

>>
```

2. Debug a FORTRAN program using the /COMPILE switch to force compilation and the /DEBUG switch to generate additional debugging information.

```
@DEBUG /COMPILE /DEBUG FORT.FOR
FORTRAN:FORT
MAIN.
LINK:    Loading
[LNKDEB FORDDT Execution]
STARTING FORTRAN DDT

>>
```

3. Using incompatible switches, try to debug a program. (The system ignores one of them and continues.)

```
@DEBUG/COMPILE/OPTIMIZE/DEBUG FORT
FORTRAN: FORT
%ERROR IS GLOBAL OPTIMIZATION NOT SUPPORTED WITH
/DEBUG - /OPT IGNORED
MAIN.
LINK:   Loading
[LNKDEB FORDDT Execution]

STARTING FORTRAN DDT

>>
```

4. Get a time-ordered list of TEST1 files in your directory. Debug an old version of it.

```
@TDIRECTORY TEST1.*
                    WRITE
```

```
   PS:<LATTA>
 TEST1.CBL.2         5-Jan-85 13:10:57
   .LST.1            6-Jan-85 14:22:00
   .REL.1            6-Dec-84 10:08:17

 Total of 3 files
@DEBUG TEST1/RELOCATABLE
LINK:   Loading
[LNKDEB DDT Execution]
DDT
```

2.24 DEFINE

Establishes or cancels logical names for your job.

Format

@**DEFINE** (LOGICAL NAME) **name: list**

where:

name: is any combination of up to 39 alphanumeric characters that you want to use as a logical name. Use an asterisk (*) for this argument to withdraw all logical names. The colon after the logical name is optional.

list is a series of devices, file structures, directories, file specifications, and/or other logical names; each item should be separated from the others by commas.

Default – not specifying a list withdraws the logical name definition

Characteristics

Colon Designates a Logical Name

Normally, when you give a logical name to an EXEC command in the place of a file specification, structure, or directory name, a colon must follow the logical name. However, for the DEFINE and INFORMATION LOGICAL-NAMES commands, where the argument can only be a logical name, a colon after the logical name is optional.

Hints

DEFINE in LOGIN.CMD File

Your DEFINE command is valid for the current terminal session only. If there are logical names that you always want to use, put DEFINE commands into a LOGIN.CMD (or, for batch jobs started by SUBMIT commands within the control files of other batch jobs, a BATCH.CMD) file in your log-in directory.

Redefining System Logical Names

You can use the DEFINE command to redefine any system

logical name for your own job. By repeating a system logical name in its own search list you expand its definition to include the other items, in the order you specify. Consider the system logical name SYS:, which is searched whenever you give a program name in place of a TOPS-20 command. If you redefine SYS: to be str:<directory>, SYS: you can run programs in str:<directory> by typing just their names. This will work as long as the program names are not the same as TOPS-20 commands.

Logical Names as Dummy File Specifications

You can use logical names as dummies for file specifications or devices when writing programs. Then, just before running such a program, use the DEFINE command to define these as real file specifications or devices, without changing the program itself.

More Information

For more information about using logical names, see the TOPS-20 User's Guide.

Special Cases

Using Recognition in the File Specifications

Normally, when you attempt to use recognition in a nonexistent filename, the system rings the terminal bell. However, for the DEFINE command, instead of ringing the terminal bell, the system may append part(s) of the default file specification, .0 or ..0, to the logical name definition. This is because DEFINE allows you to define a directory or file specifications that may not yet exist.

Note that a logical name definition that includes .0 or ..0 may not work for your use of the logical name; it is recommended that you specify the complete directory name or file specifications.

Restriction

Adding Comments to a DEFINE Command Line

You can add a comment to the end of any TOPS-20 command by preceding the comment with one of the comment characters: an exclamation point (!) or a semicolon (;). However, only the exclamation point can be used with the DEFINE command.

Using Short Logical Names Only

Although logical names can be up to 39 characters long and can include dollar signs ($), hyphens (-), and underlines (_), some commands and programs (such as programs originally written for the TOPS-10 operating system) accept a more limited set of logical names. These can be no more than 6 characters long and cannot include any special symbols. If all your logical names are of this kind, they will be acceptable to any TOPS-20 programs and commands.

Related Commands

INFORMATION LOGICAL-NAMES	for finding out the current definitions of logical names

Examples

1. Define a logical name for your job.

   ```
   @DEFINE LGN:  <MANUALS>, <SARTINI>
   ```

2. Withdraw the logical name.

   ```
   @DEFINE LGN:
   ```

3. Define a logical name to be a set of directories to which you have access. Then use the logical name to copy a file from one of them into your connected directory.

   ```
   @DEFINE MSM: <MANUALS>, <SARTINI>, <MCELMOYLE>
   @COPY MSM:4-UPED.TXT
    <MCELMOYLE>4-UPED.TXT.1 => 4-UPED.TXT.1 [OK]
   ```

4. Add one of your own directories to the definition of SYS: so that you can run .EXE files in that directory by typing just the program name.

   ```
   @DEFINE SYS: SYS:,AURORA:<WHITING.TOOLS>
   ```

2.25 DELETE

Marks a file(s) for eventual erasure.

Format

@DELETE (FILES) **filespec,...,**
@@subcommand

where:

filespec — is the specification of a file that you want to delete.
Default .gen - all generations of the specified files

@@subcommand — means that after a final comma you can give one of the following subcommands:

DELETE Subcommands

ARCHIVE — both deletes the disk copy (if any) and gives up the tape copy of specified archived files.

BEFORE — deletes the specified files that were created prior to the time and date indicated.

CONTENTS-ONLY — deletes and immediately expunges only the disk copy of files that also have a tape copy. Note that you must use the RETRIEVE command, not UNDELETE, to restore such files to disk.

DIRECTORY — deletes and immediately expunges a subdirectory without making the disk space available to the files of other users. Subdirectories appear as files with the type .DIRECTORY in the immediately superior directory. For users with enabled WHEEL or OPERATOR capabilities only.

EXPUNGE — immediately and permanently erases the specified files from the directory.

FORGET — deletes and immediately expunges the specified files without making their disk space available to the files of other users; for users with enabled WHEEL or OPERATOR

capabilities only.

KEEP n — saves the n most recent generations of the specified files while deleting the rest.
Default n - 1

LARGER n — deletes the specified files that are larger than n number of pages.

SINCE — deletes the specified files that were created since the time and date indicated.

SMALLER n — deletes the specified files that are smaller than n number of pages.

Output

Notice of Deleted Archived Files

Whenever an archived file is completely expunged as a result of your DELETE command (that is, when you also give the ARCHIVE subcommand), the operator sends a mail message notifying the owner of the directory from which the file was taken.

Characteristics

Privileged Subcommands to DELETE

The DIRECTORY and FORGET subcommands to the DELETE command are intended for privileged users only, and only as a last resort, because they withhold freed disk space from system use. Users with enabled WHEEL or OPERATOR capabilities can run the CHECKD program to recover this disk space.

DIRECTORY should not be used unless the KILL subcommand to a BUILD command fails to delete the directory. FORGET is for removing damaged files from directories, and should not be used unless DELETE without subcommands fails to delete the file.

Hints

Removing Open Files

If DELETE with the EXPUNGE subcommand fails to erase a file, it may be that some job in the system has opened it. The INFORMATION FILE-STATUS command tells whether your own job

has done so. If it has, give the CLOSE or (if the file is mapped) RESET command before repeating DELETE and EXPUNGE.

Recovering Deleted Archived Files

If you have given the DELETE command with the ARCHIVE subcommand to delete an archived file, and the disk copy has already been expunged, you may still be able to recover the tape copy. The operator will send a MAIL message (see Output, above) concerning the discarded tape copy of the deleted file. Use this information, along with Hints - Undoing DISCARD, in the DISCARD command description, to attempt recovery of the deleted file.

Special Cases

Files With the "Permanent" Attribute

The system erases only the contents of any files that have the Permanent attribute (for example, MAIL.TXT in your log-in directory) when you include them in a DELETE command. Their file specifications remain among your deleted files, and cannot be removed by TOPS-20 commands.

Restrictions

Using Logical Names When Specifying Files for Deletion

If you include a logical name when specifying arguments to a DELETE command, the system will search for the specified file in only the first directory of the logical name's definition. This restriction prevents the accidental deletion of another file if the file you intended to delete has already been deleted.

Using the LARGER/SMALLER and BEFORE/SINCE Subcommands Together

You can use size-related and time-related subcommands together. However, error messages are displayed if you use the LARGER/SMALLER pairing and the SMALLER number of pages exceeds the LARGER number, or you use the BEFORE/SINCE pairing and the SINCE date is later than the BEFORE date.

Warning

Erasure of Deleted Files

Ordinarily an UNDELETE command given during the same

terminal session as an original deletion will recover the deleted files, unless you included the EXPUNGE subcommand to DELETE or gave a subsequent EXPUNGE command. However, if any user or a batch job logs out while connected to your directory, all deleted files are permanently erased. Also, if available disk space is low on the system, the operator or the system itself may expunge all deleted files from a structure even though you have not logged out. A warning message is usually sent before this happens.

Related Commands

DIRECTORY, with the DELETED subcommand	for displaying a list of deleted files
DISCARD	for giving up only the tape copy of on-line files
EXPUNGE	for permanently removing deleted files
INFORMATION DISK-USAGE	for finding out how much disk space is available, and how much is associated with deleted files
UNDELETE	for recovering deleted files

Examples

1. Delete two of your files.

```
@DELETE TTY.SCM, VERCBL.BAT
 TTY.SCM.1 [OK]
 VERCBL.BAT.2 [OK]
```

2. Delete all your object files and all your backup files produced by the EDIT program. Then log out (this will expunge them).

```
@DELETE *.REL, *.Q*
 TESTA1.REL.1 [OK]
 TESTF1.REL.1 [OK]
 TESTC1.QBL.2 [OK]
 TESTF1.QOR.4 [OK]
@LOGOUT
Killed Job 32, User J.L.PAGE, Account 341, TTy 41
  at 25-Apr-84 10:15:51, Used 0:1:46 in 1:23:59
```

3. Delete some files, and check what files are currently deleted in your connected directory. Give the UNDELETE command for two of these, then expunge the remaining deleted files and verify that they are gone.

```
@DELETE *.QXT
 4-UPED.QXT.7 [OK]
 MAIL.QXT.1 [OK]
 REMARK.QXT.3 [OK]
@DIRECTORY,
@@DELETED
@@

   PS:<J.L.PAGE>
 4-UPED.QXT.1,2,3,4,5,6,7
   .TXT.7,8,9
 MAIL.QXT.1
 MEMO.QMD.1
 REMARK.QXT.1,2,3
 TTY.SCM.1
 VERCBL.BAT.2

Total of 17 files
@UNDELETE TTY.SCM, VERCBL.BAT
 TTY.SCM.1 [OK]
 VERCBL.BAT.2 [OK]
@EXPUNGE
 PS:<J.L.PAGE> [8 pages freed]
@DIRECTORY,
@@DELETED
```

4. Delete the files with the .MAC extension that were created between 2-Feb-87 and 3-Mar-87.

```
@DELETE *.MAC,
@@BEFORE 3-MAR-87 11:00:04
@@SINCE 2-FEB-87 23:30
@@
@
```

5. Delete the files named MYFILE.MEM that are between 50 and 150 pages.

```
@DELETE MYFILE.MEM,
@@LARGER 50
@@SMALLER 150
@@
@
```

2.26 DEPOSIT

Modifies the contents of a specific memory location.

Format

@**DEPOSIT** (MEMORY LOCATION) **address** (CONTENTS) **data**

where:

address is an octal number or a symbol.

data is a symbolic or numerical expression.

Output

Status of Pages

When you complete a DEPOSIT command, the system gives you a message indicating the status of the page you are trying to change: "[New]" for previously nonexistent pages, "[Shared]" for those having Copy-on-Write status, or "?Can't write that page" for other pages. (See also Hints - Setting the Page-access of Memory Pages, below.) However, no message is printed for deposits made to private pages.

Hints

Using Symbols

For symbols that are defined in multiple modules of a program, you can be specific by giving the module name followed by an ampersand (&) and the symbol name.

Using DDT Instead

Usually the DEPOSIT command is unnecessary, as the DDT program provides more powerful methods for modifying the contents of memory.

Abbreviating DEPOSIT Arguments

The contents of each memory location are represented as two 6-digit octal numbers. By inserting a pair of commas between these two numbers, you can abbreviate them. For example, to deposit 000004000050 into memory location 151003, use the command

```
@DEPOSIT 151003 4,,50
```

This is the same as

```
@DEPOSIT 151003 4000050
```

Note that you can also insert commas between expressions. For example, the command

```
@DEPOSIT 1 1+3,, 5+7
```

deposits 000004000014 into memory location 1. (Expressions are considered to be octal unless they contain an 8 or a 9, in which case they are considered to be decimal and are translated to octal.)

The DEPOSIT command itself can be abbreviated by the single letter D.

Deposit Address Defaults to the One Examined, and Vice Versa

The first argument of a DEPOSIT command defaults to the address examined by your most recent EXAMINE command. (You must press the ESCAPE key to take this default.) The argument of an EXAMINE command defaults to the address whose contents were modified by your most recent DEPOSIT command. Therefore you can examine a memory location, deposit a new value in it, and verify your action, while specifying the location only once. If you give DEPOSIT commands without intervening EXAMINE commands (or vice versa), the default address increases by 1 for each subsequent command.

Setting the Page-access of Memory Pages

If the system responds to a DEPOSIT command with an error message of the form, "?Can't write that page", give the SET PAGE-ACCESS COPY-ON-WRITE command for the page. Then give DEPOSIT again. If the system allows it, you will be given your own copy of the page to modify.

Using DEPOSIT With Inferior Processes

To modify memory for a process inferior to the one immediately below the TOPS-20 command processor, you must give the FORK command to specify this process before using DEPOSIT. Remember that for an inferior process to run, all superior processes must be running too. INFORMATION PROGRAM-STATUS tells you which processes these are.

Effect on Memory

The DEPOSIT command changes one location in memory.

Related Commands

DDT	for calling a debugging program, allowing more efficient modification of memory
EXAMINE	for displaying the contents of a specific memory location
FORK	for selecting the process whose memory you want to modify
INFORMATION MEMORY-USAGE	for displaying a list of memory pages, their contents and status
SET PAGE-ACCESS	for making it possible to write to specified pages

Examples

1. Deposit a value in a memory location.

   @DEPOSIT 1500 21

2. Modify a memory location, using symbols. Then examine the location.

   @DEPOSIT T3+1 P+2
   @EXAMINE T3+1
   T3+1/ P+2 (4/ 21)

3. Try to deposit a number into a page of memory that does not allow it. Examine memory, set the page to Copy-on-Write status, and try again (succeeding this time).

   @DEPOSIT 716505 0
   ?Can't write that page
   @INFORMATION MEMORY-USAGE

    216. pages, Entry vector loc 462207 len 254000

     Section 0   R, W, E, Private
   0-11   Private   R, W, E
   20   Private   R, W, E
   400-401   Private   R, W, E

```
402-660       <FIELD-IMAGE>FORTRA.EXE.3   13-271   R, CW, E
700-730       <NEXT-RELEASE>PA1050.EXE.4   1-31   R, E
731-733       Private   R, W, E

@SET PAGE-ACCESS 716 COPY-ON-WRITE
@DEPOSIT 716505 0
 [Shared]
@EXAMINE 716505
716505/  0
```

4. Check your program status (the arrow [=>] indicates your current process [fork]). Select an inferior process, deposit a value into a memory location, and verify that memory for the superior process is not changed to this.

```
@INFORMATION PROGRAM-STATUS
 Used 0:00:05 in 0:10:11
 TOPS-20: 0:00:03.5
 SET UUO-SIMULATION (FOR PROGRAM)
 SET CONTROL-C-CAPABILITY (OF PROGRAM)
 => MACRO (1): ^C from IO wait at 775701, 0:00:00.3
      Fork 2: HALT at 472052, 0:00:00.1
@FORK 2
@DEPOSIT 3500 12
@EXAMINE 3500
3500/   12
@FORK 1
@EXAMINE 3500
3500/   202200,,1136
```

2.27 DETACH

Disconnects your job from your terminal.

Format

@**DETACH** (AND) **argument**

where:

argument — can be one of these:

CONTINUE – directs the current program to proceed, just as if you had typed the CONTINUE command.

REENTER – reenters the current program, just as if you had typed the REENTER command.

START – starts the current program, just as if you had typed the START command.

Default – the program is left in its present state

Characteristics

Effects of Detached Jobs

Detached jobs use scarce system resources (such as swapping space, process slots, job slots) and can prevent new users from logging in.

Warning

Programs Writing to the Controlling Terminal (Device TTY:)

If a program running in a detached job attempts to write to device TTY:, the job will wait until it is again attached to a terminal.

Effect on Terminal

The DETACH command leaves your terminal detached from the system in the state before log-in.

| If the system manager enables the hangup-on-DETACH feature, then

| if a user DETACHes from a DECSYSTEM-20, the system hangs up the terminal line that user was connected to. The job still remains detached, but the user no longer retains a terminal line.

Related Commands

ATTACH	for joining a detached job to your terminal
SYSTAT	for finding out which jobs are detached
SUBMIT	for running independent jobs
UNATTACH	for disengaging another job from its terminal

Examples

1. Detach your job.

```
@DETACH
Detaching job # 16
```

2. Detach your job while starting the program in memory, then log in again.

```
@DETACH START
 Detaching job # 45

 BOSTON, TOPS-20 Development System, TOPS-20 Monitor 7(21002)
@LOGIN LATTA 341
```

3. Log in and put a program in memory; detach the job while starting this program, and repeat the entire procedure. Log in a third time and begin execution of a third program. Interrupt this execution with CTRL/C, then detach this third job while continuing its program. Now you have three jobs running at once. Instead of logging in again, attach the first job (specifying the job number) and verify the system's action.

```
@LOGIN LATTA 341
 Job 5 on TTY230 26-Mar-87 14::09, Last Login 26-Mar-87 11:36:12

@GET DMN
@DETACH START
 Detaching job # 5

 BOSTON, TOPS-20 Development System, TOPS-20 Monitor 7(21002)
@LOGIN LATTA 341
 Job 22 on TTY222 26-Mar-87 14:42:03, Last Login 26-Mar-87 14:38:09
```

```
@GET TESTA1
@DETACH START
 Detaching job # 22

 BOSTON, TOPS-20 Development System, TOPS-20 Monitor 7(21002)
@LOGIN LATTA 341
 Job 53 on TTY222 26-Mar-87 14:44:02, Last Login 26-Mar-87 14:42:03

@EXECUTE TESTF1
FORTRAN: TESTF1
MAIN.
LINK:   Loading
[LNKXCT  TESTF1 Execution]
^C

@DETACH CONTINUE
 Detaching job # 53

 BOSTON, TOPS-20 Development System, TOPS-20 Monitor 7(21002)
@ATTACH LATTA
|    Job 22, Detached, Running DETACH
|    Job 53, Detached, Running DETACH
|  Job: 5
Password:___

EXIT
@SYSTAT LATTA
   5*   222   EXEC      LATTA
  22    DET   TESTA1    LATTA
  53    DET   TESTF1    LATTA
```

2.28 DIRECTORY

Displays information about the files in a directory.

Format

@DIRECTORY (OF FILES) **filespec,...,**
@@subcommand

where:

filespec	is the specification of a file about which you want information. **Default** filespec - *.*.*
@@subcommand	means that, after a final comma, you can give one or more subcommands on successive lines.

Summary of DIRECTORY Subcommands (defaults in boldface)

ACCOUNT
ALPHABETICALLY
ARCHIVED
BEFORE date and/or time

CHECKSUM	**BY-PAGES** SEQUENTIALLY
CHRONOLOGICAL	**WRITE** CREATION READ TAPE-WRITE

COMPLETE
CRAM

DATES	**WRITE** CREATION OFFLINE-EXPIRATION ONLINE-EXPIRATION READ TAPE-WRITE

DELETED

```
DOUBLESPACE
EVERYTHING
FIND number of generations     Default number - 1
GENERATION-RETENTION-COUNT
HEADING
INVISIBLE
LARGER number of pages
LENGTH
LPT

      ---
      | ACCOUNT
      | CHECKSUM
      | CRAM
      | DATES
      | DOUBLESPACE
      | FILE-LINES
      | GENERATION-RETENTION-COUNT
      | HEADING
NO    | LENGTH
      | LPT
      | PROTECTION
      | REVERSE
      | SEPARATE
      | SIZE
      | SUMMARY-LINES
      | TIMES
      | USER
      ---

OFFLINE
ONLINE
OUTPUT filespec                Default filespec - DIR.DIR
PROHIBIT-MIGRATION
PROTECTION
RESIST-MIGRATION
REVERSE
SEPARATE
SINCE date and/or time
SIZE
SMALLER number of pages

        ---
        | WRITE
        | CREATION
TIMES   | OFFLINE-EXPIRATION
        | ONLINE-EXPIRATION
        | READ
        | TAPE-WRITE
        ---
```

```
      ---
USER  | WROTE
      | CREATED
      ---
```

DIRECTORY Subcommands

ACCOUNT	prints the account to which storage fees for the files are charged.
ALPHABETICALLY	lists the files in alphabetical order. **Default**
ARCHIVED	restricts the listing to archived files only, visible and invisible, offline and online.
BEFORE date and time **or** day of week (or TODAY) and time	restricts listing to files last written before the date and time given.
CHECKSUM (SEQUENTIALLY | BY-PAGES)	computes and prints 6-digit octal checksums for the files, either sequentially and without going beyond the EOF (end-of-file) mark, or by pages on disk, accounting for holes in files and pages beyond the EOF mark; output will be followed by letter P in this case. **Default** - BY-PAGES
CHRONOLOGICAL (CREATION | WRITE | READ | TAPE-WRITE)	lists files in order (oldest first) according to o date of creation, or o date they were last changed, or o date they were last read, or o date their tape copy was created,

Default – WRITE

COMPLETE	prints the complete file specification, which includes the structure and directory names.
CRAM	compresses formats to reduce printing space and time.
DATES --- | CREATION | WRITE | READ | TAPE-WRITE | OFFLINE-EXPIRATION | ONLINE-EXPIRATION ---	lists for the specified files, the following: o date of creation, or o date they were last changed, or o date they were last read, or o date the tape copy was created, or o date of expiration **Default** – WRITE
DELETED	limits descriptions to deleted files that have not yet been expunged.
DOUBLESPACE	double-spaces the DIRECTORY command output.
EVERYTHING	prints, in this order, the following information about the files: o file specification o protection o account number o size in pages and in bytes (and associated byte size) o generation retention count o date and time of creation, of last change (Write), last time read, and of the creation of any tape copy o the name of the user who created the file, and of the

	user who last wrote in the file.
FIND n	prints the specifications of all but the n most recent generations of the files, omitting any files having n or fewer generations. **Default** n - 1
GENERATION-RETENTION-COUNT	tells the number of generations of each file the system will retain in the given directory.
HEADING	prints a headline labeling each category of information supplied by the command. **Default**
INVISIBLE	restricts the listing to invisible files only, both on-line and off-line.
LARGER n	lists only files of size greater than n pages.
LENGTH	gives the file length in bytes and the associated byte size.
LPT	directs the command output to LPT: instead of to your terminal.

| ACCOUNT
| CHECKSUM
| CRAM
| DATES
| DOUBLESPACE
| FILE-LINES
| GENERATION-RETENTION-COUNT
| HEADING
NO | LENGTH
| LPT
| PROTECTION
| REVERSE
| SEPARATE
| SIZE
| SUMMARY-LINES
| TIMES
| USER
---

suppresses the action or information associated with the specified subcommand. (FILE-LINES refers to the information pertaining to the individual files, which is the bulk of the DIRECTORY command output. SUMMARY-LINES refers to the information following the file lines, giving a total file-count, and total page-count and total checksum if required by

	subcommands.) **Default** - HEADING
OFFLINE	restricts the listing to (visible) off-line files only, both archived and not archived.
ONLINE	restricts the listing to on-line files.
OUTPUT filespec	directs the command output to the specified file rather than to your terminal. **Default** filespec - DIR.DIR
PROHIBIT-MIGRATION	restricts the listing to files that are never migrated, because they were specified in a SET FILE PROHIBIT command.
PROTECTION	prints the protection code (protection number) of the file.
RESIST-MIGRATION	restricts the listing to files that the system considers last for migration. These files were specified in a SET FILE RESIST command.
REVERSE	causes an ordering subcommand, such as ALPHABETICALLY or CHRONOLOGICAL, to arrange its output in reverse.
SEPARATE	lists the complete specification for each file on a separate line (instead of listing successive generation numbers of the file on the same line, separated by commas; and instead of listing files of the same name and different type by file type only, indented under the first complete file specification).
SINCE date and time **or** day of week (or TODAY) and time	limits listing to files last written after the date (or day of week) and time given.
SIZE	prints the size of the files in pages.

SMALLER n	lists only files of size less than n pages.
TIMES --- | CREATION | WRITE | READ | TAPE-WRITE | OFFLINE-EXPIRATION | ONLINE-EXPIRATION ---	lists, for the specified files, the following: o time and date of creation, or o time and date they were last changed, or o time and date they were last read, or o time and date the tape copy was created, or o time and date of expiration **Default** - WRITE
USER --- | CREATED | WROTE ---	gives the name of the user who created the file, or changed the file last. **Default** - WRITE

Hints

Listing Unneeded Files

In preparation for deleting files so that your directory will fall within disk quotas, you can get a list of your largest files by using the LARGER and/or SIZE subcommands, and of your oldest or least-used files with DATES, TIMES, and BEFORE. With FIND you can discover extra generations of files.

Finding Files of a Particular Age or Size

To examine only files of a certain age or size, give the pair of subcommands BEFORE and SINCE, or LARGER and SMALLER, with appropriate arguments.

Comparing Checksums of Files

You can use the numbers reported by the CHECKSUM subcommand to compare two files: if they have differing contents they will almost certainly yield different values; and identical files will have the same checksums. The CHECKSUM subcommand causes a checksum of checksums as well.

Special Cases

Asterisks Appearing Before Filespecs

An asterisk (*) appearing before a filename in the response to a DIRECTORY command indicates a possible hardware-related error, one caused by a read operation at a marginally functional area of disk. To test whether there actually is an error in the file, use the COPY command to copy the file to a new specification. If the COPY command succeeds, there is no error in the file, and no asterisk will precede the new file specification. If the COPY command fails, you should move the disk to another drive and repeat the command. If it still fails, you may have to write your own program to recover everything but the missing part of the file (usually, just one page).

DIRECTORY-class Commands for Labeled Magnetic Tapes

The FDIRECTORY, TDIRECTORY, and VDIRECTORY commands for labeled magnetic tapes are equivalent to the DIRECTORY command for labeled magnetic tapes. All these commands rewind the tape set to the beginning of the first volume, print a directory of files, then rewind the set again. You can give only these subcommands when using DIRECTORY-class commands with labeled magnetic tapes: ALPHABETICALLY, DOUBLESPACE, HEADING, LPT, NO, OUTPUT, REVERSE, and SEPARATE.

Related Commands

FDIRECTORY (Full DIRECTORY)
TDIRECTORY (Time-ordered DIRECTORY)
VDIRECTORY (Verbose DIRECTORY)

other DIRECTORY-class commands for performing related functions

Examples

1. Obtain a listing of your files.

```
@DIRECTORY

  PS:<HERRICK>
4-UPED.TXT.13
ACCT20.FOR.1
DUMPER.MAC.1
F-O.DIRECTORY.1
FORD.CTL.2,3,4,5,6
```

```
 MEMO.CMD.1
   .FIL.1
   .FRM.2
 MULTIP.FOR.2

 Total of 13 files
```

2. Use a DIRECTORY command with a filespec consisting of wildcard characters and the account attribute (;A) to find out which files' storage fees are being charged to account MONITOR.

```
@DIRECTORY *.*;AMONITOR

   PS:<HERRICK>
 FORD.CTL.2
 MEMO.FRM.2

 Total of 2 files
```

3. Find out what files of type .TXT there are in your connected directory and in one to which you have group rights.

```
@DIRECTORY *.TXT, <SARTINI>*.TXT

   PS:<HERRICK>
 4-UPED.TXT.13
 MAIL.TXT.1
 REMARK.TXT.4

 Total of 3 files

   PS:<SARTINI>
 CHAP21.TXT.33
 CHAPT2.TXT.16
 CHAPT3.TXT.8
 PRIVATE.TXT.1
 TEST.TXT.1

 Total of 5 files

 Grand total of 8 files
```

4. Give a DIRECTORY command with the BEFORE and SINCE subcommands to find out which files were changed during the week of March 6, 1985.

```
@DIRECTORY,
@@BEFORE 3-12-85
@@SINCE 3-5-85
@@
```

```
  PS:<HERRICK>
DIVIDE.FOR.4
MULTIP.FOR.2
QUOTNT.EXE.1
SQUARE.EXE.1

Total of 4 files
```

5. Give the DIRECTORY command to list all the files in a directory that you have access to.

```
@DIR WORK:<HERRICK>

  WORK:<HERRICK>
CALENDAR.TXT.26
COMAND.MIC.3
QUERY.DAVE.1
     .GENE.2
RESULT.SCM.1
WEEKLY.STA.15

Total of 6 files
```

2.29 DISABLE

Disables any special capabilities, such as those of WHEEL or OPERATOR, that you have enabled.

Format

```
$DISABLE (CAPABILITIES)
@
```

Characteristics

Resumption of Standard Prompt

The DISABLE command causes the system to resume the standard at sign prompt (@) in place of the dollar sign prompt ($), which indicated enabled capabilities.

Warning

Disabling Promptly

Be sure to disable your capabilities as soon as you have finished using them, so that you or a program you run cannot accidentally damage the system.

Related Commands

ENABLE for activating any capabilities that the system manager has given you

Examples

1. Disable your capabilities.

```
$DISABLE
@
```

2. Try copying a file from a directory to which you have no access. Then enable your capabilities (assuming you have been granted capabilities), copy the file, and give up your capabilities with the DISABLE command.

```
@COPY <MON-SPCS>FFE.SPC
?Directory access privileges required
@ENABLE
```

```
$COPY <MON-SPCS>FFE.SPC
 <MON-SPCS>FFE.SPC.1 => FFE.SPC.2 [OK]
$DISABLE
@
```

2.30 DISCARD

Gives up the tape copy of specified on-line files.

Format

@**DISCARD** (TAPE INFORMATION FOR FILES) **filespec,...**

where:

filespec	is the specification of a file whose tape copy you want to discard.

Characteristics

Action of DISCARD

The DISCARD command causes the tape pointer in the FDB (File Descriptor Block) of the specified file to be deleted. It does not actually destroy the tape copy of the file. The tape copy is destroyed when the tape is recycled by the operator. (See also Hints - Undoing DISCARD, below.)

Withdrawing Archive Status of Files

When you give the DISCARD command for an on-line archived file, you withdraw archive status from the disk copy of the file. That is, the file becomes an ordinary disk file, which you can edit or delete if you wish.

DISCARD for Non-archived Files

You can also use the DISCARD command to give up the tape copy of files that have been migrated to tape (automatically, by the system) and then retrieved using the RETRIEVE command.

Hints

Undoing DISCARD

You receive a mail message from the operator for every file whose tape copy you discard. The message gives the tape number, saveset number, and file number within the saveset of each tape copy. If you have given the DISCARD command for a file and later wish to use the tape copy, you may be able to recover it using this information, as long as the tapes have not yet been recycled.

Related Commands

ARCHIVE	for requesting that a permanent tape copy of specified files be made
DELETE (with CONTENTS-ONLY subcommand)	for deleting only the disk copy of files that also have a tape copy
RETRIEVE	for requesting that an off-line file be restored to disk

Examples

1. Discard the tape copy of a file.

```
@DISCARD TESTER.EXE
 TESTER.EXE.1 [OK]
```

2. Attempt to alter an archived file. When you find out it has archive status, discard its tape copy (this revokes its archive status) and perform the alteration. Archive the resulting file and check its status.

```
@APPEND FOO.LOG DRCHIVE.TXT
  FOO.LOG.1
?File has archive status, modification is prohibited:
DRCHIVE.TXT.1
@DISCARD DRCHIVE.TXT
  DRCHIVE.TXT.1 [OK]
@APPEND FOO.LOG DRCHIVE.TXT
  FOO.LOG.1 [OK]
@ARCHIVE DRCHIVE.TXT
  DRCHIVE.TXT.1 [Requested]
@INFORMATION ARCHIVE-STATUS DRCHIVE.TXT
  DRCHIVE.TXT.1 Archive requested
```

2.31 DISMOUNT

Gives up access to the specified structure or tape set.

Format

@DISMOUNT medium (NAME) **dev: /switch(es)**

where:

medium — is one of the following optional arguments:

STRUCTURE - for dismounting file structures (disk packs)

TAPE - for dismounting magnetic tapes

dev: — is either the structure identification (or alias), or a logical name referring to the tape set (the tape setname specified in your previous MOUNT command), or a logical device name of the form MTn:. The colon after the device name is optional.

/switches — are keywords, chosen from the list below, indicating your choice of DISMOUNT command options.

DISMOUNT Command Switches
(for use with argument STRUCTURE only)

/NOWAIT — tells the system to return your terminal to TOPS-20 command level as soon as you give the DISMOUNT command, and to send a message to your terminal when the request has been processed. Otherwise, your terminal waits for the message.

/REMARK:"remark" — sends the specified remark (of 119 or fewer characters, which must be enclosed in quotation marks [" "]) to the operator when he is notified of your request. The remark is sent only if you also include the /REMOVE switch.

/REMOVE — tells the operator that you want him to physically dismount the structure from the drive; makes the structure unavailable for further mount requests. See also Hints – Action of DISMOUNT Command Including /REMOVE Switch, below.

/STRUCTURE-ID:structure identification

gives the name of the structure as recorded in the disk(s); used when you give some alias as argument dev:, above. See Hints – Using the /STRUCTURE-ID Switch, below.

Characteristics

Action of DISMOUNT STRUCTURE Command

Ordinary DISMOUNT STRUCTURE

The DISMOUNT STRUCTURE command reduces by 1 the mount count of the specified structure (the number of users who have given a MOUNT but not a DISMOUNT command for the structure) if you had given a previous MOUNT command for it. An ordinary DISMOUNT STRUCTURE command does not withdraw the structure from system use.

Including /REMOVE Switch

If you include the /REMOVE switch when giving the DISMOUNT STRUCTURE command, the specified structure is made unavailable for further mount requests. The operator is informed of your dismount request, and any further action depends on him. If he denies your request, the structure is again made available to other users; if he grants your request, the structure remains unavailable for further mount requests, and is taken off line and physically removed. Under extreme conditions the operator may take a structure off line and physically remove it even though some users have not dismounted the structure. Before doing so, he will usually send a message to such users to allow them to close their files.

Action of DISMOUNT TAPE Command

The DISMOUNT TAPE command unloads the currently mounted volume of the specified tape set (i.e., rewinds it completely onto its source reel) so that it can be

physically removed by the operator, and returns the tape drive to the pool of available resources. (Note that if the /NOUNLOAD switch was given in your original MOUNT command, no volumes are unloaded by the system or removed by the operator, even after your DISMOUNT command is completed.) If a logical name (such as the setname of the tape set) is used in the DISMOUNT command to specify the tape set, the system also withdraws the definition of the logical name. Use DISMOUNT TAPE only for tape drives having device names of the form MTn:; drives obtained using the MOUNT command. Use UNLOAD to unload tapes from drives having device names of the form MTAn:.

Hints

Omitting "medium" Argument

If the dev: argument of your DISMOUNT command will be unambiguous (for example, you do not have both a structure and a tape set mounted using the same device name), you need not specify the medium. The shortened command, DISMOUNT dev:/switch(es), is sufficient.

Using the /STRUCTURE-ID Switch

The /STRUCTURE-ID switch gives the name of the structure as recorded in the disk(s) of the pack itself, where it is used by the system for identification. You may use this switch to dismount a structure that had been mounted using an alias different from its structure identification. (See Hints - Using the /STRUCTURE-ID Switch, in the MOUNT command description.)

Using INFORMATION VOLUMES before DISMOUNT TAPE

If you give the INFORMATION VOLUMES command before dismounting a tape set, the system will respond with a list of the volids for mounted volumes, including volids for any volumes newly added to the set. You should keep an up-to-date record of these for use with further MOUNT commands.

Effect on Terminal

The DISMOUNT command with the /NOWAIT switch, leaves your terminal at TOPS-20 command level. If you have not given the /NOWAIT switch, your terminal waits until the system has processed your request, or until you give a CTRL/C to return to TOPS-20 command level. This CTRL/C does not cancel your request.

Related Commands

CANCEL
: for withdrawing mount requests before they are processed

INFORMATION AVAILABLE DEVICES
: for finding which tape devices (if any) have been assigned to your job

INFORMATION MOUNT-REQUESTS
: for finding out information about pending mount requests for structures and tape sets, and currently mounted tape sets

INFORMATION STRUCTURE
: for finding out information about the specified mounted structure, including its mount count and the names of users who have given the MOUNT, CONNECT, and ACCESS commands for the structure

INFORMATION VOLUMES
: for finding out the volids of all mounted volumes (including newly created volumes) of a tape set

Examples

1. Dismount a magnetic tape set you have been using.

```
@DISMOUNT TAPE MT3:
[Tape dismounted]
```

2. Dismount the same tape set, referring to it by its setname.

```
@DISMOUNT TAPE LAT:
[Tape dismounted, logical name LAT: deleted]
```

3. Find out the volids of your tape set before dismounting it, in case the tape set has been extended to another volume.

```
@INFORMATION VOLUMES MT3:
Volumes of tape set LAT: LAT,00J16
@DISMOUNT MT3
[Tape dismounted]
```

4. Find out if you have any mount requests pending or any currently mounted tape sets. Dismount a currently mounted tape set (these display the actual device name (here, MTA0) in the column headed, Status).

```
@INFORMATION MOUNT-REQUEST/USER
```

```
Tape/Disk Mount Queue:
Volume    Status  Type   Write   Req Name  Req#  Job#  User
-------  -------  ----  -------  --------  ----  ----  ------
UNLBLD     MTA0  Tape   Locked   UNLBLD    128    55   LATTA
There is 1 Request in the Queue
@DISMOUNT TAPE UNLBLD:
[Tape dismounted, logical name UNLBLD: deleted]
```

5. Dismount a structure you have mounted named SNARK.

```
@DISMOUNT SNARK
Structure SNARK: dismounted
```

6. Find out whether your mount request for a structure has been satisfied yet (it has not). Use the CANCEL command to withdraw this request.

```
@INFORMATION MOUNT-REQUESTS

Tape/Disk Mount Queue:
Volume    Status  Type   Write   Req Name  Req#  Job#  User
-------  -------  ----  -------  --------  ----  ----  ----------
MARK     MTA1     Tape  Enabled  MARK       126    60  HOVSEPIAN
TAPE     MTA3     Tape  Enabled  TAPE       148    13  WALLACE
LATB     Waiting  Disk           LATB       157    65  LATTA
There are 3 Requests in the Queue

@CANCEL MOUNT 157
[1 mount request canceled]
```

7. Find out whether you can safely dismount and remove a structure you have mounted. Use the SEND command to ask another user to dismount the structure; then enable your capabilities and give a DISMOUNT command that will physically remove it.

```
@INFORMATION STRUCTURE LATB:
Status of structure LATB:
Mount count: 2, open file count: 0, units in structure: 1
Foreign
Users who have MOUNTed LATB: LATTA, GBLAINE
Users ACCESSing LATB: LATTA, GBLAINE
No users CONNECTed to LATB:
@SEND GBLAINE PLEASE DISMOUNT LATB: AS SOON AS CONVENIENT. -
I MUST REMOVE THE STRUCTURE.  THANKS.
@INFORMATION STRUCTURE LATB:
Status of structure LATB:
Mount count: 1, open file count: 0, units in structure: 1
Foreign
Users who have MOUNTed LATB: LATTA
Users ACCESSing LATB: LATTA
No users CONNECTed to LATB:
```

```
@END-ACCESS LATB:<OPERATOR>
@ENABLE
$DISMOUNT STRUCTURE LATB: /REMOVE/REMARK:"PLEASE LEAVE LATB: -
ON RP06 CABINET"
[Mount Request LATB Queued, Request-ID 164]
Structure LATB: removed
$DISABLE
@
```

2.32 EDIT

Invokes your defined editor to modify a file.

NOTE

This manual assumes that you are using the EDIT program to edit. See the Special Cases section below for information relating to other editors.

Format

@**EDIT** (FILE) **/switch(es) input filespec** (OUTPUT AS) **output filespec**

where:

switches — are keywords, chosen from the list below, indicating your choice of EDIT command options.
Defaults are shown in the list of switches

input filespec — is the specification of the file you want to edit.
Default – last file specification and associated switches you gave in a CREATE or EDIT command during the current terminal session

output filespec — is the specification with which you want to name the edited file.
Default – the input file specification, but with a generation number 1 higher than the highest existing generation number

Summary of EDIT Command Switches (Defaults in boldface)

NOTE

These switches are applicable only if you are using the EDIT editor.

/BAK
/C128
/C64
/DECIDE

/DPY	
/EXPERT	
/INCREMENT:n	**Default** n - 100
/ISAVE:n	
/LOWER	
/M33	
/M37	
/NOBAK	
/NODECIDE	
/NONSEPARATORS	
/NONUMBER	
/NOVICE	
/NUMBER	
/OLD	
/OPTION:name	
/PLINES:n	**Default** n - 16
/R	
/READONLY	
/RONLY	
/RUN:filespec	**Default** file type - .EXE
/SAVE:n	
/SEPARATORS	
/SEQUENCE	
/START:n	**Default** n - argument of INCREMENT switch
/STEP:n	**Default** n - 100
/UNSEQUENCE	
/UPPER	
/WINDOW:n	**Default** n - 10

EDIT Command Switches

/BAK — causes an unedited copy of the file to be saved at the end of an editing session under the specification name.Qyp, where name.typ is the file's original specification.
Default

/C128 — specifies a 128-character alphabet, allowing insertion of control characters in an alternate format. See the TOPS-20 EDIT Reference Manual for details.

/C64 — specifies a 64-character alphabet, disallowing use of an alternate format for insertion of control characters.
Default

/DECIDE — lets you decide whether to accept or reject each change caused by the operation of the S

(substitute) command of the EDIT program.

/DPY — has no effect in the current monitor.

/EXPERT — tells the EDIT program that you need only abbreviated error messages, and fewer warnings and reminders.

/INCREMENT:n — specifies the value that will be added to each line number of the file to obtain the next line number.
Default n - 100

/ISAVE:n — instructs the EDIT program to update the backup file of specification name.Qyp after every n lines you insert, instead of only at the end of the EDIT session.

/LOWER — specifies that all alphabetic characters you type should be considered lowercase characters; give uppercase characters by preceding the corresponding lowercase character with a single quotation mark (').

/M33 — has no effect in the current monitor.

/M37 — has no effect in the current monitor.

/NOBAK — prevents an unedited copy of the file from being saved at the end of an editing session under specification name.Qyp, where name.typ is the file's original specification.

/NODECIDE — ensures the automatic operation of the S (substitute) command of the EDIT program.
Default

/NONSEPARATORS — notifies the EDIT program that the characters . (period), $ (dollar sign), and % (percent sign) are to be regarded as ordinary textual characters and not as field delimiters (separators) in the file being edited.
Default

/NONUMBER — suppresses the printing of line numbers with each line of a file.

/NOVICE — tells the EDIT program that you want to see complete error messages and all appropriate warnings and reminders; opposite of /EXPERT switch.

Default

/NUMBER — prints a line number for each line of the file.
Default

/OLD — causes the first backup file to be saved under the specification name.Zyp, where name.typ is the file's original specification.

/OPTION:name — sets any EDIT switches contained in lines of your SWITCH.INI file labeled with name (of six or fewer characters). The system expects this file to be in your log-in directory.

/PLINES:n — specifies how many lines to print in response to each P (print) command of the EDIT program.
Default n - 16

/R — same as /READONLY.

/READONLY — prevents any changes to the file during the current session of the EDIT program, that is, makes it a read-only session. This switch cannot be given in the SWITCH.INI file.

/RONLY — same as /READONLY.

/RUN:filespec — specifies the program to be run when you end the current session of the EDIT program with the G command.
Default file type - .EXE

/SAVE:n — instructs the EDIT program to update the backup file of specification name.Qyp after every n EDIT program commands that modify the file.

/SEPARATORS — notifies the EDIT program that the characters . (period), $ (dollar sign), and % (percent sign) are not ordinary textual characters but are field separators in the accompanying file.

/SEQUENCE — tells the EDIT program not to strip the line numbers from the file when the EDIT session ends.
Default

/START:n — specifies the first line number for the EDIT program to use when numbering the file.
Default n - argument of /INCREMENT switch

/STEP:n — same as /INCREMENT.

/UNSEQUENCE — tells the EDIT program to strip the line numbers from the file when the EDIT session ends.

/UPPER — specifies that all alphabetic characters you type should be considered uppercase characters; give lowercase characters by preceding the corresponding lowercase character with a single quotation mark (').
Default

/WINDOW:n — specifies the number n (between 10 and 99) of pages to be held in memory during the EDIT session.
Default n - 10

Characteristics

Edit Mode or Input Mode

The EDIT command runs the EDIT system program in Edit mode, which uses an asterisk prompt (*). (However, see also Special Cases - Using an Editor Other than EDIT, below.) In Edit mode you can use any EDIT program commands to modify the specified file. If the EDIT program starts by printing the word Input instead of Edit, the specified file does not yet exist. You are then in Input mode, just as if you had used the CREATE command instead of EDIT. See the CREATE command description for details.

Saving Backup Files Periodically

Whenever you use EDIT, be sure to keep an extra copy of the file you are modifying, in case of a system failure. By default the system renames the unedited copy of your file to name.Qyp at the end of an editing session. By using the /SAVE:n switch you can have this backup file updated periodically during the editing session to reflect your edits.

SWITCH.INI File

If there is a group of EDIT command switches that you always or often use with EDIT or CREATE commands, put them into a file named SWITCH.INI in your log-in directory, in a line of that file beginning with "EDIT:abc", where abc is any set of characters you choose to identify the line. Then if you include the single switch /OPTION:abc when you give an EDIT

or CREATE command, all these switches will be in effect.

Hints

Debugging Your Programs and Editing the Sources

You can use EDIT to modify files containing source programs written in a programming language. The DDT and DEBUG commands run system programs that offer more efficient and powerful techniques for testing temporary corrections to your programs, but you should use the EDIT command to make final changes to the source files.

Further Information

For more information about the EDIT program, see the TOPS-20 EDIT Reference Manual.

Special Cases

Using an Editor Other than EDIT

The CREATE, EDIT, and PERUSE command descriptions in this manual assume that these commands call on the EDIT program for their action. If your job uses another editing program, for example EDT, the switches and examples shown here will not be applicable.

The editor used by CREATE, EDIT, and PERUSE is specified by logical name EDITOR:, so you can find out the name of this program by giving the command, INFORMATION LOGICAL-NAMES EDITOR:. The job-wide definition (if any) will be given first, followed by the system-wide definition; the job-wide definition prevails if both exist. If the definition of EDITOR: is SYS:EDIT.EXE, the CREATE and EDIT commands will function as described in this manual. Otherwise, you must consult the appropriate manual (for example, the EDT-20 Reference Manual) for information.

You can use the DEFINE command to define logical name EDITOR: to be any editing program available at your installation. Then this program will be in effect when you give the CREATE or EDIT command.

Attempting to Edit Archived Files

If you attempt to edit an on-line archived file, the system will let you produce an edited version of the archived file, but will retain the original (archived) file unchanged under

the specification name.Qyp (or name.Zyp if you included the /OLD switch in the EDIT command), where name.typ is the file's original specification. See also Hints - Editing Files of Type .Qyp, below.

Editing Files of Type .Qyp

If you edit a file of type .Qyp (any file whose type begins with the letter Q), the EDIT program does not save the unedited copy as a backup file. In such cases, give the /OLD switch to retain the unedited copy under file type .Zyp. If the file of type .Qyp is an archived file, you will not be allowed to produce an altered version using the EDIT command unless you include the /OLD switch.

Effect on Memory

The EDIT command clears any unkept forks from memory, then loads the editor program defined by the logical name EDITOR:.

Related Commands

CREATE	for creating new files
DIRECTORY-class commands	for getting lists of existing files
PERUSE	for editing files in read-only mode

Examples

1. Edit a file.

   @EDIT FILE.FOR
   Edit: FILE.FOR.1
   *

2. Edit a file using the EDIT editor, requesting that an updated copy of the file be saved after every three EDIT program commands; ask that the first such backup file be saved under specification FILE.ZOR. @EDIT /SAVE:3/OLD FILE.FOR
   Edit: FILE.FOR.1
   *

3. Edit a large text file, adjusting several EDIT program parameters as you begin, and give new specifications for the output file.

@EDIT /EXPERT/DECIDE/PLINES:50/WINDOW:99 REMARK.TXT REVISION.TXT

```
Edit: REMARK.TXT.18
*
```

4. Use the EDIT editor to create a SWITCH.INI file with one line for the switches used in Example 2, and one line for those in Example 3. Use this file to repeat Example 3.

```
@CREATE SWITCH.INI
Input: SWITCH.INI.1
00100   EDIT:ABC/SAVE:3/OLD
00200   EDIT:DEF/EXPERT/DECIDE/PLINES:50/WINDOW:99
00300
*E

[SWITCH.INI.1]
@EDIT /OPTION:DEF REMARK.TXT REVISION.TXT
Edit: REMARK.TXT.18
*
```

2.33 ENABLE

Enables any special capabilities you may have.

Format

```
@ENABLE (CAPABILITIES)
$
```

Characteristics

Dollar Sign Prompt

The ENABLE command causes the system to print a dollar sign prompt ($), indicating enabled capabilities, in place of the standard at sign prompt (@). The dollar sign prompt is printed after ENABLE even if you have not been granted any capabilities.

Capabilities of Log-In Directory Only

The ENABLE command activates only those capabilities that have been granted to the owner of your log-in directory. You do not receive any capabilities as a result of CONNECT or ACCESS commands or group memberships.

Hints

Displaying Capabilities

Capabilities are defined by your login directory but are a characteristic of your job and can be enabled in any directory to which you connect. To list your capabilities, give the INFORMATION DIRECTORY command for your login directory.

More Information

Capabilities are assigned with the BUILD command. See the BUILD command subcommands in this manual for brief descriptions of special capabilities.

Special Cases

Dollar Sign Prompt in Batch Jobs

Because a dollar sign placed in the location of a TOPS-20

prompt could be confused with a batch command, the system precedes the enabled prompt with a space for batch jobs.

Capabilities Changed While Logged In

Capabilities are given to your job when you log in. If your capabilities are changed while you are logged-in, you must log out and log in again for the change to take effect.

Warning

Disabling Capabilities Promptly

Because your commands are much more powerful if you have capabilities enabled, you should disable them as soon as you have finished using them. Otherwise you or a program that you run could accidentally damage the system.

Related Commands

DISABLE	for suspending any capabilities that the system manager has given you
INFORMATION DIRECTORY (for the login directory)	for finding out which capabilities, have been granted to you.

Examples

1. Enable your capabilities.

```
@ENABLE
$
```

2. Try to assign a tape drive to your job before taking it off line for repairs. But it is already assigned to another user, whose terminal is set to refuse links. Enable your capabilities and ask him to deassign the tape drive. Then disable capabilities.

```
@ASSIGN MTA2:
?MTA2: Already assigned to job 29
@SYSTAT 29
  29      53   EXEC    R.SCHNEIDER
@TALK R.SCHNEIDER
?Refused, send mail to the user instead
@ENABLE
$TALK R.SCHNEIDER
```

```
LINK FROM F.DOMINO, TTY 221
$;ROBIN - PLEASE DEASSIGN MTA2:.  IT MUST BE TAKEN OFF LINE
$;FOR MAINTENANCE.  USE MTA3: INSTEAD.  THANKS.
@;OKAY, SURE.
@DEASSIGN MTA2:
@
$BREAK
$DISABLE
@
```

2.34 END-ACCESS

Terminates your ownership rights to an accessed directory, as well as group rights borrowed from its owner.

Format

@END-ACCESS (TO DIRECTORY) **dev:<directory>**

where:

dev:<directory> is the directory to which you want to end access.

Default dev: - your connected structure

Default <directory> - the directory (on the specified structure) of the same name as your connected directory

Hints

Implicit END-ACCESS

You can access only one directory at a time on each mounted structure. Each ACCESS command ends access obtained by any previous ACCESS command for a directory on the specified structure. Therefore you do not need to give the END-ACCESS command if you access another directory on the structure, or if the structure is dismounted.

Restoring Previous Rights

END-ACCESS does not restore owner and group rights obtained by a previous ACCESS command for the specified structure. Give another ACCESS command to regain these. (Note that you must access your log-in directory to regain group rights obtained by the LOGIN command, lost by accessing another directory on the public structure.

Related Commands

ACCESS for obtaining ownership rights to a directory and the group rights of the owner

DISMOUNT for decrementing the mount count of a previously accessed structure

INFORMATION STRUCTURE
for finding out who is accessing a structure

Examples

1. Give up your access rights to another user's directory.

```
@END-ACCESS <HOLLAND>
```

2. Access another user's directory, copy a file from it, and give up your rights to it. Then give a command that depends on your own group rights. (It fails.) Access your own directory to establish these, and repeat the command, successfully this time.

```
@ACCESS <HOLLAND>
Password:___
@COPY <HOLLAND>DIST.LST
 <HOLLAND>DIST.LST.2 => DIST.LST.2 [OK]
@END-ACCESS <HOLLAND>
@INFORMATION DIRECTORY <LATTA.*>,
?No such directory
@ACCESS <LATTA>
@INFORMATION DIRECTORY <LATTA.*>,
@@NAME-ONLY
@@
 Name PS:<LATTA.A>
 Name PS:<LATTA.A.F-O>
```

3. Mount a structure, and access a user's directory there. Get a listing of his files of type .TXT. End the access and dismount the structure.

```
@MOUNT STRUCTURE SNARK:
Structure SNARK: mounted
@ACCESS SNARK:<HOLLAND>
Password:___
@DIRECTORY SNARK:<HOLLAND>*.TXT

   SNARK:<HOLLAND>
 ACCT.TXT.1
 MAIL.TXT.2
 REMARKS.TXT.1
 SYSTEM.TXT.1


 Total of 4 files
```

```
@END-ACCESS SNARK:<HOLLAND>
@DISMOUNT STRUCTURE SNARK:
Structure SNARK: dismounted
```

2.35 EOF

Writes an end-of-file mark on the specified magnetic tape. Use this command for unlabeled tapes only.

Format

@**EOF** (DEVICE) **dev:**

where:

dev: is the name of the magnetic tape drive on which you want to write an end-of-file mark. The colon after the device name is optional.

Hints

EOF Seldom Needed

Because tape-writing programs and commands automatically write end-of-file marks in the appropriate places, you do not ordinarily need the EOF command. But it can be useful if such a program is interrupted (by your CTRL/C or by a system failure and restart) and you want to preserve the information already written. Also, you can shorten files on an existing tape by giving an EOF command at the desired point.

Restrictions

EOF With Open Files

If you have given a CTRL/C to exit from a program that has opened a magnetic tape drive and you then give the EOF command for that tape drive, the system will first allow you to close the associated file. You must do so for the EOF command to succeed, but you will probably be unable to continue the program from that point, because the file will now be closed.

Related Commands

BACKSPACE REWIND SKIP UNLOAD	other TOPS-20 commands for controlling magnetic tape drives

Example

1. Put an end-of-file mark (EOF) on your magnetic tape.

```
@EOF MTA0:
```

2.36 ERUN

Runs a system program without disturbing the program already in memory by placing the program in an ephemeral fork.

Format

@ERUN (PROGRAM) **filespec**

where:

filespec — is the file specification of a system program.
Default dev:<directory> - SYS:

Default .typ - .EXE

Characteristics

Characteristics of an Ephemeral Fork

The ERUN command runs a program in an ephemeral fork. A program that runs in an ephemeral fork acts like an EXEC command (excluding those EXEC commands that run programs or otherwise affect memory). Ephemeral forks and EXEC commands share these characteristics:

- They do not affect programs in memory. For example, if you exit a program and issue a SYSTAT command, or run a program in an ephemeral fork, neither the command nor the ephemeral fork will disturb the program in memory.

  A program that runs ephemerally is always placed in a new fork.

- They disappear when interrupted or when processing completes. For example, if you stop execution of a SYSTAT command or an ephemeral fork with CTRL/C, neither the command nor the ephemeral fork can be continued.

  Whenever you stop or exit an ephemeral fork, the fork is automatically reset (cleared from memory).

Good candidates for ephemeral forks are programs that may have short execution times and simply display information in a manner similar to the INFORMATION and SYSTAT commands.

Related Commands

INFORMATION LOGICAL-NAMES	for examining the definition of SYS:
R	for running executable programs stored on SYS:
RUN	for running executable user programs
SET FILE EPHEMERAL	for giving a file a permanent ephemeral attribute
SET PROGRAM EPHEMERAL	for running a program in an ephemeral fork

Example

1. Display the status of the fork in memory with the INFORMATION FORK-STATUS command. Then, run a program in an ephemeral fork in order to preserve the state of your job's memory. Redisplay the fork status and note that the ephemeral program has been reset and has not replaced the original fork.

```
@INFORMATION FORK-STATUS
 => EMACS (1): HALT AT 50340, 0:00:03.6
@ERUN TERMSTAT
TERMSTAT>SHOW FREE TERMINALS

Lab  Terminals in use  Free terminals
---  ----------------  --------------
A          17                0
B          12                4

TERMSTAT>EXIT
@INFORMATION FORK-STATUS
 => EMACS (1): HALT at 50340, 0:00:03.6
```

2.37 EXAMINE

Displays the contents of a memory location.

Format

@**EXAMINE** (MEMORY LOCATION) **octal or symbolic address**

Output

Contents of Memory Location or Message

When you complete an EXAMINE command, the system prints the memory address examined, followed by a slash (/) and its contents. If you previously used the SET TYPEOUT MODE SYMBOLIC command, this information is both in symbolic and, in parentheses, numeric (octal) format. (The numeric information will always appear for this setting of the command; symbolic information will appear if the system finds that it is different from the numeric.)

Generally the numeric format shows two 6-digit octal numbers separated by a pair of commas (,,). If you do not see this pair of commas, only the right half of the memory location is being displayed; as the left half is 0. However, if you are not permitted to examine this location, the system prints only a message telling you of the restriction.

Hints

Using Symbols

For symbols that are defined in multiple modules of a program, you can be specific by giving the module name followed by an ampersand (&) and the symbol name.

Abbreviating EXAMINE

The EXAMINE command can be abbreviated by the single letter E.

Default Argument for EXAMINE

The argument of your current EXAMINE command defaults to a value greater by 1 than the last address examined, allowing you to inspect a section of memory with only a minimum of typing. But if you gave a more recent DEPOSIT command, the argument of your current EXAMINE command defaults to that address, allowing you to verify the deposit.

Using EXAMINE With Noncurrent and Inferior Forks

The EXAMINE command displays memory locations of the current fork. To examine the memory of noncurrent or inferior forks, make the fork the current fork by giving the FORK command with the fork name or number as an argument.

To run an inferior fork after examining it, you must ensure that all superior forks are running too. Give the CONTINUE command with the superior fork name or number as an argument to let the superior fork continue its inferiors.

Related Commands

DDT	for calling a debugging program, allowing more efficient examination of memory
DEPOSIT	for changing the contents of a specific memory location
FORK	for selecting the fork whose memory you want to examine
INFORMATION MEMORY-USAGE	for displaying a list of memory pages, their contents and status
SET TYPEOUT MODE	for displaying information in symbolic or numeric format

Examples

1. Examine location 550 of the current fork.

```
@EXAMINE 550
550/    74473,,414155
```

2. Examine location 20, first in numeric typeout mode, then in the symbolic mode.

```
@SET TYPEOUT MODE NUMERIC
@EXAMINE 20
20/ 104000,,56

@SET TYPEOUT MODE SYMBOLIC
@EXAMINE 20
P+1/ 104000,,.JBBLT+11   (20/ 104000,,56)
```

3. Put a program into memory and find out what pages it occupies. Examine a location on page 2, and then (using the

abbreviated form of the EXAMINE command) one on page 400.

```
@GET DMN
@INFORMATION MEMORY-USAGE

 5. pages, Entry vector loc 400010 len 254000

  Section 0    R, W, E,  Private
0-3           DMN.EXE.1  1-4   R, CW, E
 400          DMN.EXE.1  5   R, CW, E

@EXAMINE 2550
2550/   600170
@E 400550
400550/ 0
```

2.38 EXECUTE

Loads your program into memory, compiling the source file first if necessary. Then it starts the program.

Format

@EXECUTE (FROM) **/switch(es) source/switch(es) object,...**

where:

switches — are keywords chosen from the list below, indicating your choice of EXECUTE command options. They have different effects depending on their position in the command line: placed before all files in the command, they act on defaults for all; otherwise they affect only the nearest preceding file.
Defaults are shown in the list of switches

source — is the file specification of the source program. The filename must be of 6 or fewer characters, and the file type of 30 fewer characters; you cannot use a generation number. This argument is not necessary if you supply an object filespec.

object — is the file specification of the object program. The filename must be of six or fewer characters, and the file type must be .REL; you cannot use a generation number. This argument is not necessary if you supply a source.
Default (if you give neither source nor object filespecs) - last filespecs and associated switches you gave in a LOAD-class command

Summary of EXECUTE Command Switches (defaults in boldface)

/10-BLISS
/36-BLISSS
/68-COBOL
/74-COBOL
/ABORT
/ALGOL
/BINARY
/COBOL
/COMPILE
/CREF
/CROSS-REFERENCE
/DDT

/DEBUG
/FAIL
/FLAG-NON-STANDARD
/FORTRAN
/LANGUAGE-SWITCHES:"/switch(es)"
/LIBRARY
/LIST
/MAC
/MACHINE-CODE
/MACRO
/MAP
/NOBINARY
/NOCOMPILE
/NOCREF
/NOCROSS-REFERENCE
/NODEBUG
/NOFLAG-NON-STANDARD
/NOLIBRARY
/NOLIST
/NOOPTIMIZE
/NOSEARCH
/NOSTAY
/NOSYMBOLS
/NOWARNINGS
/OPTIMIZE
/PASCAL
/RELOCATABLE
/SAIL
/SEARCH
/SIMULA
/SNOBOL
/STAY
/SYMBOLS
/WARNINGS

Descriptions of these switches are given below. Although the system will not reject switches described under any of the LOAD-class commands, only those switches commonly associated with EXECUTE are described here.

EXECUTE Command Switches

/10-BLISS compiles the file using the BLISS-10 compiler.
Default for files of type .B10 and .BLI

/36-BLISS compiles the file using the BLISS-36 compiler.
Default for files of type .B36

/68-COBOL compiles the file using the COBOL-68 compiler.

Default for files of type .C68 or .68C

/74-COBOL compiles the file using the COBOL-74 compiler.
Default for files of type .C74 or .74C

/ABORT stops a compile if a fatal error is detected and returns your terminal to TOPS-20 command level.

/ALGOL compiles the file using the ALGOL compiler.
Default for files of type .ALG

/BINARY allows generation of an object (binary) file for each source file given.
Default

/COBOL compiles the file using the COBOL compiler, either COBOL-68 or COBOL-74, that your installation has stored in the file SYS:COBOL.EXE.
Default for files of type .CBL

/COMPILE forces compilation of the source file even if a current object file already exists. Use this switch along with a /LIST or /CREF switch to obtain listings when you have current object files.

/CREF same as /CROSS-REFERENCE.

/CROSS-REFERENCE
creates a file containing cross-reference information for each compilation. The file name is that of the object file; the file type is .CRF. Use the CREF command to obtain a listing of the file. (For COBOL files this switch automatically produces a cross-reference listing.)
Default

/DDT loads the DDT debugging program along with your object file.

/DEBUG produces an object file containing debugging information beyond what is usually inserted during compilation. (For FORTRAN programs only, and only if you have not given the /OPTIMIZE switch).

/FAIL compiles the file using the FAIL compiler.
Default for files of type .FAI

/FLAG-NON-STANDARD
indicates nonstandard syntax in a file

/FORTRAN compiles the file using the FORTRAN compiler.
Default in the absence of a standard source file type and a language switch
Default for files of type .FOR

/LANGUAGE-SWITCHES:"/switch(es)"
passes the specified switches to the compiler that will process the file(s) to which this switch applies. You must include the switches in double quotation marks (" ").

/LIBRARY same as /SEARCH.

/LIST prints a line printer listing of the program in ASCII format; the name of this listing is the filename of the object file. The /CREF switch overrides /LIST when they both apply to the same file.

/MAC same as /MACRO.

/MACHINE-CODE produces a file containing the generated machine code. The filename is that of the object file; the file type is .LST. For high-level languages.

/MACRO assembles the file using the MACRO assembler.
Default for files of type .MAC

/MAP produces a loader map and stores it in the file object.MAP, where object is the name of the module containing the start address; or (if no start address) nnnLNK.MAP, where nnn is your job number.

/NOBINARY prevents generation of an object (binary) file. Use this switch along with /LIST or /CREF to allow these switches to take effect without producing a new object file.

/NOCOMPILE prevents compilation if the object file is current; otherwise it forces compilation. Cancels the /COMPILE or /RELOCATABLE switch.

/NOCREF same as NOCROSS-REFERENCE.

/NOCROSS-REFERENCE
prevents the creation of a cross-reference file.
Default

/NODEBUG excludes special debugging information from your object file.
Default

/NOFLAG-NON-STANDARD
prevents the flagging of non-standard syntax in the file.
Default

/NOLIBRARY same as /NOSEARCH.

/NOLIST prevents a line printer listing of the program.
Default

/NOMACHINE-CODE prevents generation of a file containing machine code.
Default

/NOOPTIMIZE prevents the generation of a globally optimized object file (for FORTRAN programs only).
Default

/NOSEARCH requires all modules in the object file library (the file accompanied by this switch in the command line) to be loaded even if they are not called by your program. Cancels the /SEARCH switch.
Default

/NOSTAY stops the compiler from being placed in a background fork. Use when /STAY is set as a default for the compiler.

/NOSYMBOLS prevents a symbol table from being loaded along with the object file.

/NOWARNINGS prevents display of warnings for nonfatal errors.

/OPTIMIZE generates a globally optimized object file; one that runs as quickly as possible. (For FORTRAN programs only, and only if you do not also give the /DEBUG switch (see the DEBUG command description).)

/PASCAL compiles the file using the PASCAL compiler.

Default for files of type .PAS

/RELOCATABLE identifies the input file as an object file (regardless of its extension) and prevents compilation of the source file, forcing use of an existing object file even if the object file is out of date.
Default for files of type .REL

/SAIL compiles the file using the SAIL compiler.
Default for files of type .SAI

/SEARCH requires that the object file library (the file accompanied by this switch in the command line) be searched for modules called by your program or by a program subroutine. Only these modules are loaded, along with modules called from system libraries, which are always searched.

/SIMULA compiles the file using the SIMULA compiler.
Default for files of type .SIM

/SNOBOL compiles the file using the SNOBOL compiler.
Default for files of type .SNO

/STAY returns your terminal to TOPS-20 command level so that you can perform other work while the system continues to execute your program. You immediately receive the TOPS-20 prompt (@ or $), and can then issue any user command. Be careful not to send incorrect data to programs expecting terminal input. See the CONTINUE command, Restrictions: Programs Competing for Terminal Input.)

This switch saves you from having to: issue a ^T to make sure execution has begun; give a ^C to halt the job; and issue a CONTINUE /STAY command to remain at command level during execution.

/SYMBOLS loads a symbols table along with the object file (helpful for debugging a program).
Default

/WARNINGS displays warnings for nonfatal errors.
Default

Characteristics

Compiling New Sources Only

Before executing programs, the system ordinarily compiles any source (and only those sources) whose write date is more recent that that of the object file of the same name. You can override this action with the /COMPILE or /RELOCATABLE switch.

Using Standard File Types

If you specify source files with standard types (.FOR, .MAC, .CBL, or .ALG) in an EXECUTE command, the system automatically calls the appropriate compiler when compilation is necessary. If you specify source files by filename only, the system searches your connected directory in the above order for a file of this name and a standard type. To execute programs from sources that have nonstandard file types, give a switch to indicate the proper compiler (/FORTRAN, /MACRO, /COBOL, or /ALGOL). A switch will take precedence over a standard file type if they indicate different languages. If no compiler is indicated with either a switch or a standard file type, the FORTRAN compiler is used.

Default Switches Not Passed to Compiler

Only switches specified in a LOAD-class command are passed to the compiler; default switches are not passed. Instead, the system assumes that the defaults for the compiler are the same as the defaults for the LOAD-class command.

Hints

Commas Between Filespecs

If you give two or more filespecs separated by commas as arguments to EXECUTE, the loaded programs exist in memory at the same time and will act as a single program. You can use this feature to substitute one module for another under varying conditions or for different applications.

Plus Signs Between Filespecs

If you give two or more filespecs separated by plus signs (+) as arguments to EXECUTE, they are treated as a single file by compilers. Their object module is stored under any filename given as the "object" argument of the command, or (if none) under the last filename in the group and file type

.REL.

Indirect Files as Arguments

You can store the arguments (source and object filespecs, switches) of an EXECUTE command in an indirect file, and specify them by typing an at sign (@) and its filespec as an EXECUTE command argument.

Establishing Default Arguments with the SET Command

You can issue the SET DEFAULT COMPILE-SWITCHES command to set up default global arguments to the EXECUTE command. Insert this SET command in your COMAND.CMD file to change your own defaults permanently.

Running LINK Directly

The EXECUTE command automatically runs LINK, the system's loader program, but if you require control of the loading process you can run LINK directly. See the TOPS-20 LINK Reference Manual.

Wildcards Illegal with EXECUTE

The EXECUTE command does not accept wildcard characters (* and %) in a file specification.

Effect on Memory

The EXECUTE command clears any unkept forks from memory, loads the appropriate compiler if necessary, then loads and starts your program.

Related Commands

COMPILE, LOAD, and DEBUG	other LOAD-class commands for performing related functions
RUN	for running executable programs

Examples

1. Execute a program, indicating the language with a standard file type.

```
@EXECUTE CAFN.FOR
FORTRAN: CAFN
```

```
LINK:   LOADING
[LNKXCT CAFN EXECUTION]

END OF EXECUTION
CPU TIME: 0.04 ELAPSED TIME: 0.89

EXIT
```

2. Execute a program, indicating the language with a switch. Specify the /STAY switch to return immediately to TOPS-20 command level.

```
@EXECUTE CAFN/FORTRAN/STAY
```

3. Execute two programs, requesting a cross-reference file for one of them.

```
@EXECUTE CAFN, TAFN/CREF
FORTRAN: CAFN
MAIN.
FORTRAN: TAFN
MAIN.
LINK:   LOADING
[LNKXCT TAFN EXECUTION]

END OF EXECUTION
CPU TIME: 0.04 ELAPSED TIME: 0.15
EXIT
```

4. Combine two source programs into a single object program, and run this program.

```
@EXECUTE CAFN+TAFN
FORTRAN: CAFN
MAIN.
MAIN.
LINK:   LOADING
[LNKXCT TAFN EXECUTION]

END OF EXECUTION
CPU TIME: 0.04 ELAPSED TIME: 0.16
EXIT
```

5. Execute an ALGOL program, ensuring that the compilation includes required modules only; request a map.

```
@EXECUTE /COMPILE/MAP CALEND/ALGOL, ALGMOD.LBR/SEARCH
ALGOL: CALEND
LINK:   LOADING

EXIT
```

2.39 EXPUNGE

Permanently erases all the deleted files from a directory.

Format

@EXPUNGE (DIRECTORY) **dev:<directory>,**
@@subcommand

where:

dev:<directory> is the name of the directory you wish to expunge; you may use wildcard characters to expunge more than one directory.

Default dev: - your connected structure

Default <directory> - the directory (on the specified structure) of the same name as your connected directory

Default (if no arguments - your
are given) connected
directory

@@subcommand means that after a final comma you can give one or more optional subcommands on successive lines.

DELETE deletes and expunges temporary files (those with the Temporary (;T) attribute) created by some system programs to hold interim data. Do not use if you will have any further need of these files.

PURGE expunges all files which you have opened but not closed.

REBUILD rebuilds the symbol table of the directory named.

Output

After a successful EXPUNGE command, the system reports the number of disk pages freed with the message [n pages freed]. If deleted files are mapped, they will not be expunged, and so will not contribute to the number of pages freed. Occasionally the system

will report a negative number. This can mean that files were being written in the directory during the EXPUNGE, or (especially if you include the REBUILD subcommand) that previous computations of directory size had not adequately accounted for some files, for example, files written near the time of a system crash and reload.

Hints

Using the REBUILD Subcommand

The REBUILD subcommand is not needed under usual conditions, as the system performs this action automatically. Use REBUILD if a message is printed on your terminal advising you to rebuild the symbol table of a directory.

Using the PURGE Subcommand

The PURGE subcommand is useful chiefly for removing the remains of files that were being created at the time of a system crash or a structure dismount. Do not give it while anyone might be using the directory, because that user's program might be deprived of necessary files as a result.

Cannot CTRL/C

You cannot use <CTRL/C> to interrupt an EXPUNGE once it is started.

Special Cases

Files With the "Permanent" Attribute

The system erases only the contents of any files that have the Permanent attribute, for example your MAIL.TXT file, when you try to expunge them. The file specifications of permanent files remain among your deleted files, and cannot be removed by TOPS-20 commands.

Related Commands

DELETE	for marking files to be later expunged
DIRECTORY-class commands	for obtaining lists of file specifications
INFORMATION DISK-USAGE	for finding out the size of a directory

UNDELETE	for recovering deleted files

Examples

1. Expunge all deleted files from your directory.

```
@EXPUNGE
 PS:<LATTA> [6 pages freed]
```

2. Find out how much of your disk space is in use and how much is occupied by deleted files. Delete some of your backup files, then give the EXPUNGE command to erase all of these.

```
@INFORMATION DISK-USAGE
 PS:<LATTA>
 154 Pages assigned, 101 in use, 53 deleted
 590 Working pages, 590 Permanent pages allowed
 33371 Pages free on PS:
@DELETE *.Q*
 BLUE.QAR.1 [OK]
 REMARK.QXT.1 [OK]
 RIMOUSKI.QXT.1 [OK]
@EXPUNGE
 PS:<LATTA> [56 pages freed]
```

2.40 FDIRECTORY

The FDIRECTORY (Full DIRECTORY) command is equivalent to the DIRECTORY command with the subcommands CRAM, EVERYTHING, and NOHEADING. Use the same format and subcommands with FDIRECTORY as with DIRECTORY. For further information, see the DIRECTORY command description.

When used with magnetic tapes, the FDIRECTORY command is equivalent to DIRECTORY for magnetic tapes.

Examples

1. Get a "Full DIRECTORY" listing, on your terminal, for one of your files.

```
@FDIRECTORY TESTF1.FOR
   MISC:<LATTA>
 TESTF1.FOR.17;P777700;A341   1 162(7) 1 25-Oct-85 11:17:46
25-Oct-85 11:17:46 Never Never LATTA LATTA
```

2. Give the FDIRECTORY command for a file, this time requesting the only piece of information about current files not ordinarily supplied by the command. Ask for a heading also.

```
@FDIRECTORY TESTF1.FOR,
@@HEADING
@@CHECKSUM
@@
   MISC:<LATTA>
                          PGS Bytes(SZ)  Ret   Creation
Write            Read     Tape-write   Creator Writer
Checksum
 TESTF1.FOR.17;P777700;A341   1 162(7) 1 25-Oct-85 11:17:46
25-Oct-85 11:17:46 Never   Never       LATTA LATTA 566101P
```

2.41 FORK

Makes the specified fork your current fork. The current fork is the fork to which TOPS-20 commands are applied.

Format

@**FORK** (IS) **fork**

where:

fork is one of the following: Fork name
Fork number
Default - the fork with the highest fork number

Characteristics

Default Fork

If you do not specify a fork name or number with the FORK command, the fork with the highest fork number (usually the last fork created) becomes your current fork, and the fork name is printed in brackets, [FORK-NAME].

Fork Name and Number

Forks are named after the program they contain and are numbered in the order that they are created. In multiforking class commands, the fork name and number are interchangeable.

Hints

More Information

The FORK command is one of the TOPS-20 multiforking-class commands. For more information about multiforking, see the section named, Running Multiple Programs, in the TOPS-20 User's Guide.

Special Cases

Fork 0

If you are a user with enabled WHEEL privileges you can give the command, FORK 0. This references the command processor

(EXEC) itself.

Related Commands

INFORMATION MEMORY-USAGE	for examining memory of the current process
INFORMATION FORK-STATUS	for finding out the number and status of each fork in your job
INFORMATION PROGRAM-STATUS	for finding what fork attributes have been set with the SET PROGRAM command and the number and status of each fork in your job
CONTINUE, FREEZE, KEEP, RESET, SET NAME, SET PROGRAM, and UNKEEP	other multiforking-class commands

Examples

1. Make the first fork you created your current fork.

```
@FORK 1
```

2. Display the fork status, and make the last fork you created your current fork. Then, redisplay the fork status to check the result. (In the FORK-STATUS display, an arrow (=>) indicates the current fork).

```
@INFORMATION FORK-STATUS
 => EDIT (1): Kept, HALT at 6254, 0:00:12.8
    DUMPER (2): Kept, HALT at 6065, 0:00:30.1
    HOST (3): Kept, HALT at 67543, 0:00:09.3
@FORK
@INFORMATION FORK-STATUS
    EDIT (1): Kept, HALT at 6254, 0:00:12.8
    DUMPER (2): Kept, HALT at 6065, 0:00:30.1
 => HOST (3): Kept, HALT at 67543, 0:00:09.3
```

3. Make the FORK named DUMPER your current fork; then display the fork status.

```
@FORK DUMPER
@INFORMATION FORK-STATUS
    EDIT (1): Kept, HALT at 6254, 0:00:12.8
 => DUMPER (2): Kept, HALT at 6065, 0:00:30.1
    HOST (3): Kept, HALT at 67543, 0:00:09.3
```

4. Find out which forks exist in your job. Look at memory for the first fork, then examine a particular location. Make the second inferior fork current, and do the same thing there.

```
@INFORMATION FORK-STATUS
 => QUILL (1): Kept, HALT at 50340, 0:00:04.5
       Fork 2: HALT at 21010, 0:00:00.4
@INFORMATION MEMORY-USAGE

124. pages, Entry vector loc 4570 len 3

  Section 0     R, W, E,  Private
0-5      Private   R, W, E
6-55     RANDOM:<QUILL>TECPUR.EXE.1120  1-50    R, E
56-77    Private   R, W, E
116-123  Private   R, W, E
620-637  RANDOM:<QUILL>ABBRE.:EJ.614  0-17   R, E
640-643  RANDOM:<QUILL>TYPE.:EJ.27  0-3   R, E
644-645  RANDOM:<QUILL>INIT.:EJ.17  0-1   R, E
646-661  RANDOM:<QUILL>LSTSQ.:EJ.424  0-13   R, E
662-663  RANDOM:<QUILL>SYSTEM.:EJ.1  0-1   R, E
@EXAMINE 6400
6400/ 200040,,4636
@FORK 2
@INFORMATION MEMORY-USAGE

95. pages, Entry vector loc 15710 len 3

  Section 0     R, W, E,  Private
0-11     Private   R, W, E
13-15    Private   R, W, E
16-110   RANDOM:<TOOLS>DEFFNA.3  3-75   R, CW, E
117      Private   R, W, E
166      Private   R, W, E
170      Private   R, W, E
172      Private   R, W, E
174      Private   R, W, E
224      Private   R, W, E
226      Private   R, W, E
231-250  Private   R, W, E
@EXAMINE 2600
2600/ 0
```

2.42 FREEZE

Halts execution of a fork.

Format

@**FREEZE** (FORK) **fork**

where:

fork is one of the following: Fork name
Fork number
Default - the current fork

Characteristics

Same Function as CTRL/C

The FREEZE command stops running background forks while you are at EXEC command level. The effect is the same as if you typed two CTRL/Cs while at the fork's program level.

Hints

More Information

The FREEZE command is one of the TOPS-20 multiforking-class commands. For more information about multiforking, see the section named, Running Multiple Programs, in the TOPS-20 User's Guide.

Related Commands

CONTINUE/BACKGROUND	for continuing a halted fork in the background
INFORMATION FORK-STATUS	for displaying the fork status
FORK, INFORMATION PROGRAM-STATUS KEEP, RESET, SET NAME, SET PROGRAM, and UNKEEP	other multiforking-class commands

Examples

1. Give the FREEZE command to stop the current running background fork. Then, display the fork status. The arrow points to the current fork.

```
@FREEZE
@INFORMATION FORK-STATUS
 => BLISS (1): ^C from RUNNING at 500000, 0:00:13.1
    EDIT (2): Kept, HALT at 6254, 0:00:00.5
```

2. Display the fork status and give the FREEZE command to stop a running background fork. Then, redisplay the fork status to check the result.

```
@INFORMATION FORK-STATUS
    BLISS (1): Background, RUNNING at 500000, 0:00:13.1
 => EDIT (2): Kept, HALT at 6254, 0:00:00.5
@FREEZE BLISS
@INFORMATION FORK-STATUS
    BLISS (1): ^C from RUNNING at 500000, 0:00:13.1
 => EDIT (2): Kept, HALT at 6254, 0:00:00.5
```

2.43 GET

Places an executable program into memory.

Format

@**GET** (PROGRAM) **filespec /switch**

where:

filespec — is the specification of any file containing an executable program.
Default file type - .EXE

/switch — is /USE-SECTION:n
specifies the memory section (from 0 to 37 octal) into which your program is to be loaded. You can use this switch only if your program can be contained in one section.

Effect on Memory

The GET command clears any unkept forks, puts the specified program into memory.

Related Commands

INFORMATION MEMORY-USAGE	for examining the contents of memory
KEEP	for making the specified fork a kept fork
LOAD	for loading a source or object program into memory
MERGE	for putting an executable program into memory without first clearing memory
SAVE	for storing a copy of the program in memory in a file in executable format
START	for starting the program in the current fork

Examples

1. Put an executable program into memory.

```
@GET TESTF1.EXE
```

2. Verify that you have a magnetic tape drive assigned to your job. Get one of your executable programs, save a copy of it on tape, and then start it.

```
@INFORMATION AVAILABLE DEVICES
Devices available to this job:
 DSK, PS, SNARK, MISC, LANG, REL3, DX20, MTA0
 LPT, LPT0, LPT1, CDR, PCDR0, CDP, FE0-15
 PTY23-61, NUL, PLT, PLT0, DCN, SRV
Devices assigned to/opened by this job: MT0, TTY230
@GET TESTF1
@SAVE MT0:
 MT0:TESTF1 Saved
@START

THIS IS A TEST.

END OF EXECUTION
CPU TIME: 0.03 ELAPSED TIME: 0.72
EXIT
```

2.44 HELP

Displays explanatory text for many TOPS-20 system features.

Format

@**HELP** (ON SUBJECT) **name**

where:

name is the name of a topic chosen from the list given in response to the command HELP ?
Default name - HELP

Sample of HELP Command Arguments

ACCT20	ACCTPR	ACTGEN	APL	APLSF	BLIS10	CHECKD	CHKPNT
COBDDT	COBOL	CONV20	CREF	DAEMDB	DBINFO	DBMEND	DIRTST
DLUSER	DUMPER	EDIT	FE	FILCOM	FORDDT	FORDML	FORMAT
FORTRA	HELP	ISAM	LIBARY	LINK	LPTSPL	MACRO	MAIL
MAKLIB	MAKRAM	MAKVFU	OPLEAS	PLEASE	PTYCON	QUEUE	RDMAIL
RERUN	RSXFMT	RUNINP	RUNOFF	SCHEMA	SORT	SYSERR	SYSJOB
TRANSL	ULIST	USAG20	USAH20	WATCH			

Characteristics

Other HELP Command Arguments

Note that some of the HELP command arguments shown here may be omitted if the associated topics are not available at your site. The list may include other texts inserted by your system administrator to describe features special to your system. Also, this list is revised frequently to reflect improvements and additions to standard TOPS-20 programs.

Hints

Printing HELP Files on the Line Printer

The texts displayed by the HELP command are stored in system logical name HLP: under the name of the topic and file type .HLP. Use the PRINT command to request your own copy.

Examples

1. Ask for information about the FILCOM program.

```
@HELP FILCOM
FILCOM V21B(60)

FILCOM compares two files in either ASCII mode
or binary depending upon switches or file name extensions.
All standard binary extensions are recognized as binary by
 default.
Switches are :-
/A compare in ASCII mode
        .
        .
        .
/W compare the Word mode but don't expand files
/X expand files before word mode compare
```

2. Use the DIRECTORY command to search system logical name HLP: for any text involving mail. (Note the use of wildcard characters here.) Then, print a copy of one of the help files.

```
@DIRECTORY HLP:*MAIL*.HLP

   PS:<HELP>
 MAIL.HLP.2
 RDMAIL.HLP.2

 Total of 2 files
@PRINT /AFTER:1700 HLP:MAIL.HLP
[Printer job APLSF queued, request #131, limit 174]
```

2.45 INFORMATION

Displays information about system and job parameters.

Format

@INFORMATION (ABOUT) **argument**

where:

argument is a keyword, chosen from the list below, indicating your choice of INFORMATION command options.

Summary of INFORMATION Command Arguments (defaults in boldface)

ADDRESS-BREAK

ALERTS

ARCHIVE-STATUS filespecs

```
           ---
           | LINES
AVAILABLE  | DEVICES
           ---
```

```
                ---
                | /ALL
BATCH-REQUESTS  | /FAST
                | /PROCESSING-NODE:node name
                | /USER:user name  Default user name - your user
                |                                    name
                ---
```

CLUSTER
COMMAND-LEVEL
DECNET node-name

```
                ---
                | ALL
                | CARDS
                | COMPILE-SWITCHES
                | DECLARE
                | PAPER-TAPE
DEFAULTS        | PLOT
                | PRINT
                | PROGRAM
                | SUBMIT
                | TAKE
                ---

DIRECTORY dev:<directory>,              Default dev:<directory> - your
  @@VERBOSE                                     connected    directory
  @@FAST
  @@NAME-ONLY

DISK-USAGE dev:<directory>              Default dev:<directory> - your
                                              connected    directory

FILE-STATUS  octal JFN                  Default JFN - all JFNs in your
                                                      job
FORK-STATUS
INTERNET STATUS
JOB-STATUS

                ---
                | SYSTEM
LOGICAL-NAMES   | JOB
                | ALL
                | logical name:
                ---

          ---
MAIL      | user name                   Default user name - your user
          | SYSTEM                                          name
          ---

MEMORY-USAGE
MONITOR-STATISTICS

                  ---
                  | /ALL
MOUNT-REQUESTS    | /FAST
                  | /USER:user name   Default user name - your user
                  |                                     name
                  ---
```

```
                  ---
                  | /ALL
                  | /DESTINATION-NODE:node name
OUTPUT-REQUESTS   | /FAST
                  | /USER:user name Default user name - your user
                  |                                     name
                  ---

PROGRAM-STATUS
PSI-STATUS
REMOTE-PRINTING

                     ---
                     | /ALL
RETRIEVAL-REQUESTS   | /FAST
                     | /USER:user name Default user name - your
                     |                              user name
                     ---

SPOOLED-OUTPUT-ACTION
STRUCTURE dev:                    Default dev: - your connected
                                                 structure
SUBSYSTEM-STATISTICS
SUPERIORS
SYSTEM-STATUS
TAPE-PARAMETERS
TERMINAL-MODE number              Default number - your terminal
                                                   line number
VERSION
VOLUMES
```

INFORMATION Command Arguments

ADDRESS-BREAK — gives the location (in numeric or symbolic format - depending upon previous specification of the SET TYPEOUT MODE command) and mode of any address breaks for the program currently in memory. Set with SET ADDRESS-BREAK.

ALERTS — lists the dates and times that the system is to signal you at the terminal. The last line of the display indicates whether alerts are to be sent unconditionally to your terminal (depending upon previous specification of the SET AUTOMATIC command). Set with SET ALERT.

ARCHIVE-STATUS filespecs — prints the archive status of all

specified files for which archival has been requested or for which migration has been prohibited.
Default filespec - *.*.* in your connected directory

```
            ---
            | LINES
AVAILABLE   | DEVICES
            ---
```

lists the devices or terminal lines available to you or already assigned to your job. Use ASSIGN to obtain devices (use MOUNT for structures).
Default - DEVICES

```
               ---
               | /ALL
BATCH-REQUESTS | /FAST
               | /PROCESSING-NODE:node name::
               | /USER:user name
               ---
```

lists the jobs being processed and waiting to be processed by the batch system. The list includes:

- the jobname and request ID number of the request (an asterisk (*) appears before the jobname if the job is currently being processed)
- the scheduled run time of the request
- the name of the user who initiated the request
- the values of the switches /AFTER and /DEPENDENCY-COUNT, if values were given in the original SUBMIT or subsequent MODIFY command

Use SUBMIT, MODIFY, or CANCEL to change this list.

The /ALL switch adds the switches /ASSISTANCE, /PRIORITY, /RESTARTABLE, /SEQUENCE, and /UNIQUE to this list, while /FAST eliminates the display of all switches and column headings; /PROCESSING-NODE specifies the DECnet network node about whose batch jobs you want information; /USER restricts

descriptions to jobs of the user named, and can be given with any of the other three switches.
Default user name - your user name

CLUSTER — displays the names of the systems in a Common File System (CFS) cluster:

- local cluster node (the system you are logged in on).
- accessible CFS nodes (the other systems in the CFS cluster).
- accessible HSC servers (the HSC50 device controllers in the cluster).

COMMAND-LEVEL — prints the status of the LATE-CLEAR-TYPEAHEAD parameter, which prevents you from giving another TOPS-20 command until any error message resulting from a previous command has been printed. Set with SET LATE-CLEAR-TYPEAHEAD.

DECNET node-name — tells whether the specified DECnet network node is accessible to your system. If you do not specify a node name, the system prints the name of your host system, the total number of reachable nodes, and the names of all reachable nodes.
Default node-name - all accessible nodes

DEFAULTS

ALL
CARDS
COMPILE-SWITCHES
PAPER-TAPE
PLOT
PROGRAM
PRINT
SUBMIT
TAKE

displays, in a format suitable for entering them, default arguments established at the current level of TOPS-20 for the specified command. CARDS and PAPER-TAPE refer to the PUNCH CARDS and PUNCH PAPER-TAPE commands, respectively. COMPILE-SWITCHES refers to LOAD-class commands and PROGRAM refers to the SET PROGRAM command. The ALL argument displays the defaults for all these categories. Set with SET

DEFAULT. This argument displays the default for every SET DEFAULT command given, even if duplicate setting are made.

Default - ALL

DIRECTORY dev:<directory>,

@@FAST
@@VERBOSE
@@NAME-ONLY

lists the current parameter values set for the indicated directory (with the exception of the directory password) by the SET DIRECTORY or BUILD commands, or by default. The subcommands call for either a short list of non-default (that is, user-determined) values only (FAST), or a complete list including defaults (VERBOSE), or a listing of directory names only (NAME-ONLY). If you use NAME-ONLY, specify a directory in the form <directory.*>, <*directory*>, or <*>. The categories of information include:

- the name of the directory
- working and permanent storage limits
- capabilities (assigned or withheld)
- whether you can establish DECnet or INTERNET network connections
- whether expired files should be automatically archived
- directory number
- default file protection
- default account for login
- directory protection
- default number of generations maintained for files
- maximum number of subdirectories allowed
- date and time that you started the current terminal session with LOGIN

(for log-in directory only)

- date and time of last interactive login
- date and time of last non-interactive login
- date and time password expires
- number of interactive login failures since last login
- number of non-interactive login failures since last login
- off-line and on-line expiration defaults
- group memberships
- user group numbers assignable to subdirectories
- TOPS-10 project-programmer number

Set with SET DIRECTORY or (for subdirectories) BUILD.

Default dev:<directory> - your connected directory

Default subcommand - FAST

DISK-USAGE dev:<directory>

prints, for the indicated directory, the following:

- the name of the directory
- the number of pages of assigned disk storage, and the number of deleted pages, if any
- working and permanent page limits
- total number of unused pages on the file structure containing the directory

The wildcard characters, * and % can be

included in the <directory> field. For example, type <%directory*>, <directory.*>, or <*> to get information about all matching directories or subdirectories.

Default dev:<directory> - your connected directory

FILE-STATUS octal JFN — gives, for the specified JFN (an internal number identifying each file opening), the following:

- the JFN
- the associated file specification
- the mode of access (Append, Execute, Read, or Write) for which the JFN is open (or was opened last, if NOT OPENED precedes the access mode)
- special access conditions, namely DATA ERROR if an error is made in accessing the file, or EOF if the file pointer is at the end of the file
- if appropriate, byte pointer and byte size, which tell the number of bytes transferred to or from the file, and
- a list of devices currently assigned to or opened by this job But if a file has been opened by another process for its sole use, you see only the message, "Restricted JFN".

Default JFN - all JFNs for your job

FORK-STATUS — gives a summary of the status of each fork belonging to your current copy of the TOPS-20 command processor, including Kept status, RUN status, and total CPU time used so far. An arrow (=>) indicates your current fork.

| INTERNET STATUS — displays, if the system is a member, information about Internet networks, including INTERNET, Milnet and Local Area Networks. The display includes:

		o the name of the local host system followed by its Internet name and its Internet address
		o the status of the network interface
		o whether network interface output is enabled
		o whether network service is enabled
		o the date and time of the last network interface online transition, offline transition, and cycle transition
	JOB-STATUS	prints your
		o host system (Displayed only if your host system is part of a DECnet or INTERNET network.)
		o job number
		o user name
		o connected directory (if not your log-in directory)
		o account; session remark (if any)
		o terminal number
		o terminal access descriptor
		o network node to which your output device, requests are sent. (Displayed only if not your host node.) Set with the command, SET LOCATION.
		You can set some of these parameters with CONNECT, SET ACCOUNT, SET LOCATION, and SET SESSION-REMARK.
	LOGICAL-NAMES	--- | ALL | JOB | SYSTEM | logical name: --- prints the logical names and definitions

which have been established for your job, for the system, or for both; or prints the job-wide and system-wide definitions of the specified logical name. Establish and withdraw logical names with DEFINE.

For the DEFINE and INFORMATION LOGICAL-NAMES commands, a colon following the logical name is optional. However, in INFORMATION LOGICAL-NAMES the logical name SYS: must always be followed by a colon. Otherwise, the system interprets SYS as an abbreviation for the SYSTEM argument.

The wildcard characters, * and % can be included in the logical name. For example, type A* to list all logical names that begin with the letter 'A'. See example 6.

Default - JOB

MAIL | user name
| SYSTEM
---

tells whether there is unread mail for the user, if you have read access to the user's mailbox; otherwise, you see only the message, "Mailbox protected." Also, displays any system messages since your last login when you type SYSTEM instead of user-name. Send mail with one of the two mail programs, MAIL and DECmail/MS. Read mail with the RDMAIL or DECmail/MS program.

Default user name - your user name

MEMORY-USAGE

prints, for the current process of your job, the following:

- the number of pages of memory assigned
- location (in numeric or symbolic format - depending upon previous specification of the SET TYPEOUT MODE command) and length of the current program's entry vector (see with SET ENTRY-VECTOR)

and on each succeeding line

- the page numbers of pages occupied by a file or program
- the file specification if the pages are file pages; the process specification if the pages are mapped from another process; PRIVATE otherwise.
- the page numbers of file pages or process pages. If a page is mapped by indirect pointers, the file specification is printed to which it is mapped; "Fork n" means that these pages are mapped indirectly through another process (process n) of the job; "No page" can mean either of these conditions, when the destination page does not yet exist.
- the permitted accesses to the pages (set with SET PAGE-ACCESS):

  R - Read access

  W - Write access

  CW - Copy-on-Write access

  E - Execute access

See Example 4 at the end of this command description for obtaining information on pages assigned to extended sections of memory.

MONITOR-STATISTICS gives you the following:

- the length of time (in hours, minutes, and seconds) since the monitor was reloaded
- an analysis of monitor overhead time, by percentages
- the number of swap-reads and -writes, and file-reads and -writes
- the number of pages of memory available to user programs

- the number of terminal wake-ups (occasions when a program "wakes up" after waiting for terminal input or output to finish, and of terminal interrupts (occasions when a program is interrupted by a CTRL/C, CTRL/O, or CTRL/T (or other, user-enabled control characters) typed at a user's terminal)

- the average number of processes in the balance set (NBAL, a subset of the run set - these are runnable, and each receives a share of total CPU time) and in the remainder of the run set (NRUN - these are waiting to be run)

- the number of seconds of CPU time given to each of the scheuler queues (where the leftmost listing describes the highest priority queue, for interactive processes, and the rightmost listing is for CPU-bound processes)

- if class scheduling is enabled, the allotted share and actual use of the system (expressed as a percentage of total CPU time) by each class, and the 1-, 5-, and 15-minute load averages of each class

All averages and totals are computed for the time since system start-up.

```
                ---
                | /ALL
MOUNT-REQUESTS  | /FAST
                | /USER:user name
                ---
```

prints a list, at your terminal, of pending structure-mount and tape-mount requests, and of tape-mount requests currently being satisfied. The list includes:

- the volid of the first volume of tape that will be mounted, or the volid of the mounted tape, or the

structure identification of each disk pack that will be mounted

- the status of each volume of tape (either the number of the tape drive, in the form, MTAn, on which it is mounted, or Waiting)
- the type of request (either Disk or Tape)
- the tape density specified in the tape-mount request
- the mode (either Enabled, if the /WRITE-ENABLED switch was specified or assumed in the original MOUNT-TAPE command, or Locked if /READ-ONLY applies) in which each volume of tape is to be mounted
- the request number (i.e., request ID number) of each request
- the number of the job that made the request
- the user name of the owner of the job that made the request

Use the MOUNT, CANCEL (for pending requests), and DISMOUNT (for satisfied requests) commands to change this list.

The /ALL switch adds the following to the display: the /ASSISTANCE, /PRIORITY, /RESTARTABLE/, /NOTE, /SEQUENCE, /UNIQUE, and /REMARK switches, whether a tape mount request is for a labeled tape, and the tape volume-set name. The /FAST switch eliminates column headings and the sum of the number of requests; /USER restricts descriptions to jobs of the user named, and can be given with either of the other two switches.

Default user name - your user name

```
                  ---
                  | /ALL
                  | /DESTINATION-NODE:node name
OUTPUT-REQUESTS   | /FAST
                  | /USER:user name
                  ---
```

prints a listing, at your terminal, of the requests being sent or waiting to be sent to an output device. The list includes:

- the name of the node (for remote line printer requests)
- the name of the queue (card punch, paper tape punch, plotter, or line printer)
- the jobname and request ID number of the request (an asterisk (*) appears before the jobname if the request is currently being processed)
- the output limit, in appropriate units (number of pages, minutes of plotter time, feet of paper tape, or number of cards)
- the name of the user who initiated the request, and
- values of the switches /AFTER, /FORMS, and /UNIT, if given non-default values in the original PRINT, PLOT, PUNCH, or subsequent MODIFY command.

Use PRINT, PLOT, PUNCH, MODIFY, or CANCEL to change this list.

The /ALL switch adds the /NOTE and /SEQUENCE switches to this list, while the /FAST switch eliminates the display of all switches and column headings; /USER restricts descriptions to jobs of the user named, and can be given with either of the other two switches. The /DESTINATION-NODE switch displays the print requests on remote nodes in the same TOPS-20 cluster as the local node.

Default user name - your user name

PROGRAM-STATUS

gives the following information for the current level of the TOPS-20 command processor (EXEC):

- the amount of CPU time you have used, and total elapsed time since you logged in
- the amount of TOPS-20 command processor time used
- SET UUO-SIMULATION (set with SET UUO-SIMULATION) if the TOPS-10 compatibility package is enabled to simulate TOPS-10 monitor calls issued by a program you are running
- SET CONTROL-C-CAPABILITY (set with SET CONTROL-C-CAPABILITY) if your program is allowed to handle CTRL/C interrupts itself
- the settings established with the SET TRAP and SET TYPEOUT commands
- the settings established with the SET DEFAULT PROGRAM command
- the settings established with the SET PROGRAM command
- a summary of the status of each fork belonging to the current copy of the TOPS-20 command processor, including kept status, RUN status, and total CPU time used so far

An arrow (=>) indicates your current fork.

PSI-STATUS

tells you:

- whether the PSI (Programmed Software-Interrupt) system is in use (ON) or not (OFF)
- the memory address of the level table and of the channel table - 0 if none was set

- the numbers of the priority levels for which there are interrupts in progress (1 and/or 2 and/or 3), where 1 is the highest priority

- the numbers of channels enabled (ready) to accept interrupts, and of channels with pending interrupts

For further discussion of the interrupt system see the TOPS-20 Monitor Calls Reference Manual.

REMOTE-PRINTING — displays system definitions and characteristics for remote line printers.

RETRIEVAL-REQUESTS
| /ALL
| /FAST
| /USER:user name

prints a list, at your terminal, of pending retrieval requests. Each file for which you request retrieval constitutes a separate request, even if specified within a single RETRIEVE command. The list includes:

- the name of the request (the first six characters of the filename)

- the request ID number

- the volids of each tape containing the file

- the name of the user who made the request

The /ALL switch includes the complete specification (up to 49 characters of the file, while the /FAST switch eliminates column headings; /USER restricts descriptions to requests of the user named and can be used with either of the other two switches.

Note that the /ALL switch does not

display the complete file specification unless you have Wheel or Operator privileges.

Default user name - your user name

SPOOLED-OUTPUT-ACTION — tells you whether the system processes your spooled output requests immediately, or defers them until you log out. Set with SET SPOOLED-OUTPUT-ACTION.

STRUCTURE dev: — gives, for each structure named, the following:

- information as to whether the system performs checking operations while writing to the data or swapping areas of the structure. The system would perform this checking by immediately reading the data that it has just written. If the system manager has enabled these functions, the following lines appear at the top of the display: "Write verification for data", and "Write verification for swapping".
- the number of users who have mounted the structure, the number of open files on the structure, and the number of disks in the structure
- kind of structure - Public or Private, Domestic or Foreign (see the TOPS-20 User's Guide)
- names of users who have mounted the structure
- names of users who have accessed the structure
- names of users who have connected to the structure
- whether or not the structure is offline

The colon after the structure name is optional. Use an asterisk * for dev: to specify all mounted structures.

Mount and dismount structures with the MOUNT and DISMOUNT commands.
Default dev: - your connected structure

SUBSYSTEM-STATISTICS gives, for each subsystem (any name specified by the SETSN JSYS), the following information:

- its name and total runtime since the system last started - SNAMES, STIMES
- the average number of page faults per second it has caused - SPFLTS
- the number of long-term waits it has caused - SNBLKS
- its average working-set size (the number of pages it occupies in memory)-SSIZE
- the number of times a SETSN JSYS has been executed for it (excluding the EXEC subsystem)

See the TOPS-20 Monitor Calls Reference Manual for more information.

SUPERIORS tells you the number of forks that are superior to the current EXEC level. This number is equal to the number of times you gave the PUSH command without intervening POP commands.

Note that many programs have PUSH commands and that some programs automatically do a PUSH. These PUSHes also change the number of superior forks reported by this command.

SYSTEM-STATUS tells you:

- whether the operator is present
- what kinds of logins are allowed - local, remote, pseudo-terminal, DECnet, Arpanet, or console
- whether accounting (assessing and recording charges for system use) is

being done

- o whether account validation (checking accounts against lists of authorized users) is enabled

- o whether working set preloading is enabled (Working set preloading is discussed in the System Manager's Guide and in the Software Installation Guide.)

- o whether sending of level zero system messages is enabled. System level 0 messages inform users about resource problems, such as:

  - [GIDNEY: Caution, Swapping space low]

  - [CLOYD: Caution, SPT space low]

  - [THEP: Caution, Disk space low on system structure THEP:]

- o whether sending of level one system messages is enabled. System level 1 messages inform users of operationsal type messages, such as:

  - [RONCO: Deleted files will be expunged from system structure RONCO: in 30 seconds]

  - [RONCO: Expunge of structure RONCO: completed]

- o whether sending of operator messages (like BUGCHK, BUGINF, and "RESOURCE LOW") to the CTY (central terminal) is enabled

- o whether tape-drive allocation (automatic assignment of tape drives) is enabled

- o whether automatic file retrieval-waits (the delaying of a command's execution until specified off-line files are [automatically] retrieved) are enabled

- the system's expiration default date for off-line files
- the current setting of the scheduler bias control
- whether class scheduling is enabled, and, if it is enabled, the special class (if any) for batch jobs, and the default class (if any)
- off-line structures timeout interval
- status of cluster information
- status of cluster sendalls
- minimum password length
- number of days for password to expire
- whether the password dictionary is enabled

TAPE-PARAMETERS — gives the default settings of these parameters for magnetic tapes:

- tape density, in bits per inch
- tape parity (ODD or EVEN)
- format (ANSI-ASCII, CORE-DUMP, INDUSTRY-COMPATIBLE, or SYSTEM-DEFAULT), and
- tape record length, in bytes

Set with SET TAPE.

TERMINAL-MODE number — gives the following information about the specified terminal:

- its type (for example, LA36, VT52, or SYSTEM-DEFAULT)
- its speed (baud rate), in bits per second. If the terminal is connected to the system through another node, such as a DECserver-100 or another TOPS-20

system, the terminal speed cannot be determined by this command. This is indicated in the display by the message !Terminal speed indeterminate!.

- whether all output that does not originate from your own job is inhibited.
- whether it is set to receive or refuse links, advice, and system messages
- whether it is set to pause in printing output when you type the pause character, and/or at the end of each full page of output
- the pause and continue characters that you may have set with the TERMINAL PAUSE CHARACTER command (only if TERMINAL PAUSE END-OF-PAGE and TERMINAL PAUSE COMMAND are in effect, and if CTRL/S and CTRL/Q were not the specified characters)
- the length (in number of lines) and width (in number of characters) of its page
- whether it is capable of printing lowercase characters, whether it is set to raise lowercase letters you type to uppercase, and whether it will mark (flag) capital letters with a single quotation mark (')
- whether it has a formfeed mechanism, and whether it is set to only indicate formfeeds or to perform them
- whether it has mechanical tab stops, whether it is set to immediately echo input you type
- whether it is operating in FULLDUPLEX or HALFDUPLEX mode

Set with TERMINAL. The SYSTAT command

displays terminal numbers.

VERSION — tells you:

- the name of the host system
- the TOPS-20 operating system's name and octal version number
- the octal version of the TOPS-20 command processor (EXEC) in use
- the name (and decimal or octal version number, if any) of the program in the current fork for which program data vectors (PDVs) exist and that are associated with the current process. (See the TOPS-20 Monitor Calls Reference Manual and the description of the /PVBLOCK switch in the TOPS-20 LINK Reference Manual for information on PDVs.) See Example 5.
- the decimal version of the UUO simulation package in use (if a TOPS-10 program is in memory)

The format of a version number is:

a.b(c)-d

where: (1) a and b are respectively incremented for major and minor changes in the software (2) c gives a rough indication of the number of times the software component has been edited (3) d, a holdover from earlier versions of TOPS-20 which is now rarely used, identifies the programmer(s) responsible for the software component.

VOLUMES setname: — gives the volids of currently mounted and newly created volumes in the specified tape set. A colon after the tape set name is optional.

Hints

Specifying the Current Fork of TOPS-20

Use the FORK command to specify the fork to be described by the ADDRESS-BREAK, FILE-STATUS, MEMORY-USAGE and VERSION arguments. Find out your current fork with INFORMATION FORK-STATUS.

Restrictions

Using the INFORMATION OUTPUT-REQUESTS /DESTINATION-NODE switch

For non-privileged users, the local node's GALAXY must know if the remote node has printers. If the user is privileged, then the print request queue of the node specified is displayed, regardless of whether the local GALAXY knows if the node has printers or not. The user specified node must be a node in the cluster known to the local GALAXY.

If the node specified by the /DESTINATION-NODE switch is the local node, then the print request queue of the local node is displayed.

You cannot use an * as an argument in the /DESTINATION-NODE switch.

Related Commands

SYSTAT	for printing information about the current state of the system.

Examples

1. Use an INFORMATION command to determine your current terminal settings.

```
@INFORMATION TERMINAL-MODE
 TERMINAL VT100
 TERMINAL SPEED 9600
      .
      .
      .
 TERMINAL NO IMMEDIATE
 TERMINAL FULLDUPLEX
```

2. Mount a structure and access your directory on the structure. Compare the disk space available in this directory and in your connected directory. (Note that there are many more pages free on your connected structure (MISC:) as a whole than on structure SNARK:; this is likely to make your use of the system more efficient if you work only within MISC:.)

```
@MOUNT STRUCTURE SNARK:
Structure SNARK: mounted
@ACCESS SNARK:
@INFORMATION DISK-USAGE SNARK:
 SNARK:<LATTA>
 198 Pages assigned
 400 Working pages, 400 Permanent pages allowed
 2836 Pages free on SNARK:
@INFORMATION DISK-USAGE
 MISC:<LATTA>
 119 Pages assigned
 590 Working pages, 590 Permanent pages allowed
 33172 Pages free on MISC:
```

3. Print a file, ordering several copies and supplying a note to be attached to it. Use an INFORMATION command to verify that your request is in the output queue. Modify the date on which the job will be printed, and use the INFORMATION command again to confirm this action.

```
@PRINT TESTF1.FOR /AFTER:17:00/COPIES:20/FORMS:NARROW/NO -
TE:"T-TH LAB"
[Printer job TESTF1 queued, request-ID 219, Limit 54]

@INFORMATION OUTPUT-REQUESTS /ALL/USER

Printer Queue:
Job Name  Req#  Limit            User
--------  ----  -----  ------------------------
  TESTF1   219     54  LATTA                  /Forms:NARROW
          /After: 8-Nov-85 17:00  /Note:T-TH LAB  /Seq:1791
There is 1 job in the queue (none in progress)

@MODIFY PRINT 219 /AFTER:15-NOV-85 17:00
[1 Job modified]
@INFORMATION OUTPUT-REQUESTS /ALL/USER

Printer Queue:
Job Name  Req#  Limit             User
--------  ----  -----  ------------------------
  TESTF1   219     54  LATTA                   /Forms:NARROW
          /After:15-Nov-85 17:00   /Note:T-TH LAB  /Seq:1791
There is 1 job in the queue (none in progress)
```

4. Request to print a file on a remote node, then use the INFORMATION command to verify that your request is in the remote output queue.

```
@PRINT VENUS.TXT/DESTINATION-NODE:HENSON
[Printer job VENUS queued on node HENSON, request-ID 39,
Limit 12]
```

```
@INFORMATION OUTPUT/DESTINATION-NODE:HENSON

Printer Queue:
Job Name  Req#  Limit             User
--------  ----  -----  ------------------------
  VENUS   200     54   ANDERSON

There is 1 job in the queue (none in progress)
```

5. Place a program in memory section 17. Then give the INFORMATION MEMORY-USAGE command to verify that the program was appropriately placed. The page numbers, beginning at 17000, indicate that section 17 is in use, because a section comprises 1000 (octal) pages. Also, the left half of the entry vector location contains 17.

```
@GET GRADES.EXE.1 /USE-SECTION:17
@INFORMATION MEMORY-USAGE

64.  pages, Entry vector loc 17,,542 len 254000

  Section 0     R, W, E,  Private
  Section 17    R, W, E,  Private
17000-17002      GRADES.EXE.1  1-3   R, CW, E
17374-17425      GRADES.EXE.1  4-35   R, CW, E
17600-17637      GRADES.EXE.1  36-75   R, CW, E
17643-17645      GRADES.EXE.1  76-100   R, CW, E
```

6. Issue the INFORMATION VERSION command for information on programs in your memory area that have program data vectors associated with them. Note that the merging of such programs yields consolidated information.

```
@GET IOLIB
@INFORMATION VERSION
 BOSTON TOPS-20 System, TOPS-20 Monitor 7(163)
 TOPS-20 Command processor 7(10)
 Program is IOLIB
PDVs:  Program name IOPAK, version 1.1(420)
@GET MATHLB
@INFORMATION VERSION
 BOSTON TOPS-20 System, TOPS-20 Monitor 7(163)
 TOPS-20 Command processor 7(10)
 Program is MATHLB
PDVs:  Program name MATHLB, version 3.33(360)
@MERGE IOLIB
@INFORMATION VERSION
 BOSTON TOPS-20 System, TOPS-20 Monitor 7(163))
 TOPS-20 Command processor 7(10)
 Program is MATHLB
PDVs:  Program name MATHLB, version 3.33(360)
```

```
       Program name IOPAK, version 1.1(420)
@MERGE RPTGEN
@INFORMATION VERSION
 BOSTON TOPS-20 System, TOPS-20 Monitor 7(163)
 TOPS-20 Command processor 7(10)
 Program is MATHLB
PDVs:  Program name REPORT, version 3.1(156)
       Program name MATHLB, version 3.33(360)
       Program name IOPAK, version 1.1(420)
```

7. Use the INFORMATION LOGICAL-NAMES command with the * wildcard to list all the job-wide and system-wide logical names that begin with the letter 'P'.

```
@INFORMATION LOGICAL-NAMES P*
Job-wide:

PAS: => PUB:<DBONIN.PASCAL>
PB: => PUB:PHONE.BOOK
PUB: => PUBLIC:<DBONIN>

System-wide:

PCL: => RANDOM:<PCL>
POBOX: = PUBLIC:
POST-OFFICE: => PUBLIC:<OPERATOR>
PS: => GIDNEY:
```

8. Use the INFORMATION INTERNET STATUS command to display the status of INTERNET nodes.

```
@ INFORMATION INTERNET STATUS

Local dec-internet host name is gidney.tops20.dec.com
 Network interface type is IPNI, Internet address is 16.34.0.2
 Network interface is up, output is enabled
 Network service is enabled
 Last network interface up transition: 4-May-90 23:16:04

Local dec-mrnet host name is gidney.mrnet.dec.com
 Network interface type is IPNIA, Internet address is 192.5.5.4
 Network interface is up, output is enabled
 Network service is enabled
 Last network interface up transition: 4-May-90 23:16:02

Local dec-mrrad host name is mrdale.mrrad.dec.com
 Network interface type is IPCI, Internet address is 192.5.6.12
 Network interface is up, output is enabled
 Network service is enabled
 Last network interface up transition: 4-May-90 23:16:02
```

2.46 KEEP

Protects a fork from being cleared from memory.

Format

@KEEP (FORK) **fork**

where:

fork is one of the following: Fork name
Fork number
Default - the current fork

Characteristics

Characteristics of Kept Forks

The KEEP command gives a fork a "kept" status. A kept fork has these characteristics:

- A kept fork is not reset when another program is brought into memory. Normally, when a program is run, it replaces the fork currently in memory. However, if the program in memory is in a kept fork and a new program is run, the new program is placed in a new fork. This allows you to have several programs in memory.

- A kept fork is not cleared from memory with the RESET command unless one of these RESET arguments is specified: the fork name, a period or an asterisk.

- A kept fork can be restarted by typing the kept fork name as an EXEC command. The kept fork name can be abbreviated to the point where it is unique from EXEC commands and other kept fork names. ESCAPE recognition and question mark help also function with kept fork names.

  The KEEP command sets the kept fork to restart at its starting address when the kept fork name is given as a command. For information on changing the fork's restart address see Hints, Setting the Kept Fork's Restart Address.

- A kept fork is named after the program it contains. Forks are numbered in the order in which they were created. In multiforking-class commands the fork name and number are interchangecable.

Inferior Forks

Any inferior forks created by a kept fork are also kept.

Hints

Keeping Forks Automatically

Place SET PROGRAM KEEP commands in your LOGIN.CMD or COMAND.CMD file for programs that you normally place in kept forks. Then when you load the program, the system automatically keeps the fork and notifies you with the message [Keeping FORK-NAME].

The SET PROGRAM command applies only to the current EXEC level. If you want your SET PROGRAM commands to be in effect after a PUSH command, put the commands into your COMAND.CMD file. The COMAND.CMD file is executed automatically after every PUSH command.

Kept Forks Continued Using the Fork Name

The KEEP command sets the fork to be restarted at its starting point when the fork name is given as a command. So, if a program that was kept with the KEEP command is running in a background fork, and you type the kept fork name, execution of the program is canceled and the program returns to its start address, which is usually the program's prompt.

With the SET PROGRAM KEEP command, you to specify the point at which the fork will restart when the fork name is given as a command. The starting point can be the program's continue, reentry, or start address. The command SET PROGRAM KEEP CONTINUE gives the fork name the same function as the CONTINUE /NORMALLY command. So, if a program that was kept with the SET PROGRAM KEEP CONTINUE command is running in a background fork, and you type the kept fork name, execution of the program continues and your terminal is placed at program level.

More Information

The KEEP command is one of the TOPS-20 multiforking-class commands. For more information about multiforking, see the section named, Running Multiple Programs, in the TOPS-20 User's Guide.

Restrictions

Limited Number of Forks

There is a limited number of forks available on a system. When all forks are in use, existing users cannot add forks and new users cannot log in. Therefore, KEEP only necessary forks and return idle forks to the system with the UNKEEP or RESET commands.

Related Commands

INFORMATION FORK-STATUS	for displaying the fork status
RESET	for clearing forks from memory
UNKEEP	for changing a kept fork to an unkept fork
CONTINUE, FORK, FREEZE, INFORMATION PROGRAM-STATUS, SET NAME, and SET PROGRAM	other multiforking-class commands

Examples

1. Display the fork status with the INFORMATION FORK-STATUS command. Then, give the KEEP command to make the current fork a kept fork and redisplay the fork status.

```
@INFORMATION FORK-STATUS
 => EDIT (1): HALT at 6254, 0:00:00.5
    FILCOM (2): ^C from IO wait at 700272, 0:00:00.3
@KEEP
@INFORMATION FORK-STATUS
 => EDIT (1): Kept, HALT at 6254, 0:00:00.5
    FILCOM (2): ^C from IO wait at 700272, 0:00:00.3
```

2. Display the fork status, and KEEP the FILCOM fork. Then verify the new fork status.

```
@INFORMATION FORK-STATUS
 => EDIT (1): Kept, HALT at 6254, 0:00:00.5
    FILCOM (2): ^C from IO wait at 700272, 0:00:00.3
@KEEP FILCOM
@INFORMATION FORK-STATUS
 => EDIT (1): Kept, HALT at 6254, 0:00:00.5
    FILCOM (2): Kept, ^C from IO wait at 776721, 0:00:00.3
```

2.47 LOAD

Loads your program into memory, compiling the source file first if necessary.

Format

@LOAD (FROM) **/switch(es) source/switch(es) object,...**

where:

switches — are keywords chosen from the list below, indicating your choice of LOAD command options. They have different effects depending on their position in the command line: placed before all files in the command, they act as defaults for all; otherwise they affect only the nearest preceding file.
Defaults are shown in the list of switches

source — is the file specification of the source program. The filename must be of 6 or fewer characters, and the file type of 3 or fewer characters; you cannot use a generation number. This argument is not necessary if you supply an object filespec.

object — is the file specification of the object program. The filename must be of 6 or fewer characters, and the file type must be .REL; you cannot use a generation number. This argument is not necessary if you supply a source filespec.
Default (if you give neither source nor object filespecs) - last filespecs and associated switches you gave in a LOAD-class command

Summary of LOAD Command Switches (defaults in boldface)

/10-BLISS
/36-BLISS
/68-COBOL
/74-COBOL
/ABORT
/ALGOL
/BINARY
/COBOL
/COMPILE
/CREF
/CROSS-REFERENCE
/DDT
/DEBUG

/FAIL
/FLAG-NON-STANDARD
/FORTRAN
/LANGUAGE-SWITCHES:"switch(es)"
/LIBRARY
/LIST
/MAC
/MACHINE-CODE
/MACRO
/MAP
/NOBINARY
/NOCOMPILE
/NOCREF
/NOCROSS-REFERENCE
/NODEBUG
/NOFLAG-NON-STANDARD
/NOLIBRARY
/NOLIST
/NOMACHINE-CODE
/NOOPTIMIZE
/NOSEARCH
/NOSTAY
/NOSYMBOLS
/NOWARNINGS
/OPTIMIZE
/PASCAL
/RELOCATABLE
/SAIL
/SEARCH
/SIMULA
/SNOBOL
/STAY
/SYMBOLS
/WARNINGS

LOAD Command Switches

/10-BLISS compiles the file using the BLISS-10 compiler.
Default for files of type .B10 and .BLI

/36-BLISS compiles the file using the BLISS-36 compiler.
Default for files of type .B36

/68-COBOL compiles the file using the COBOL-68 compiler.
Default for files of type .C68 or .68C

/74-COBOL compiles the file using the COBOL-74 compiler.
Default for files of type .C74 or .74C

/ABORT stops a compile if a fatal error is detected

and returns your terminal to TOPS-20 command level.

/ALGOL compiles the file using the ALGOL compiler.
Default for files of type .ALG

/BINARY allows generation of an object (binary) file for each source file given.
Default

/COBOL compiles the file using the COBOL compiler, either COBOL-68 or COBOL-74, that your installation has stored in the file SYS:COBOL.EXE.
Default for files of type .CBL

/COMPILE forces compilation of the source file even if a current object file already exists. Use this switch along with a /LIST or /CREF switch to obtain listings when you have current object files.

/CREF same as /CROSS-REFERENCE.

/CROSS-REFERENCE
creates a file containing cross-reference information for each compilation. The file name is that of the object file; the file type is .CRF. Use the CREF command to obtain a listing of the file. (For COBOL files, the switch automatically produces a cross-reference listing.) See the TOPS-20 User Utilities Guide for more information about the CREF program.

/DDT loads the DDT debugging program along with your object file.

/DEBUG produces an object file containing debugging information beyond that usually provided during a compilation (for use with FORTRAN programs only, and only if you have not given the /OPTIMIZE switch).

/FAIL compiles the file using the FAIL compiler.
Default for files of type .FAI

/FLAG-NON-STANDARD
indicates nonstandard syntax in file.

/FORTRAN compiles the file using the FORTRAN compiler.
Default in the absence of a standard source file type and a language switch

Default for files of type .FOR

/LANGUAGE-SWITCHES:"/switch(es)"
passes the specified switches to the compiler that will process the file(s) to which the switch applies. You must include the switches in double quotation marks (" ").

/LIBRARY same as /SEARCH.

/LIST prints a line printer listing of the program in ASCII format. The name of this listing is the filename of the object file. The /CREF switch overrides /LIST when they both apply to the same file.

/MAC same as /MACRO.

/MACHINE-CODE produces a file containing the generated machine code. The filename is that of the object file; the file type is .LST. For high-level languages.

/MACRO assembles the file using the MACRO assembler.
Default for files of type .MAC

/MAP produces a loader map and stores it in the file object.MAP, where object is the name of the module containing the start address; or (if no start address) nnnLNK.MAP, where nnn is your job number.

/NOBINARY prevents generation of an object (binary) file. Use this switch along with /LIST or /CREF to allow these switches to take effect without producing a new object file.

/NOCOMPILE prevents compilation if the associated object file is current; otherwise it forces compilation. Cancels the /COMPILE or /RELOCATABLE switch.
Default

/NOCREF same as /NOCROSS-REFERENCE.

/NOCROSS-REFERENCE
prevents the creation of a cross-reference file.
Default

/NODEBUG excludes special debugging information from

your object file.
Default

/NOFLAG-NON-STANDARD
prevents the flagging of non-standard syntax in the file.
Default

/NOLIBRARY same as /NOSEARCH.

/NOLIST prevents a line printer listing of the program.
Default

/NOMACHINE-CODE prevents generation of a file containing machine code.
Default

/NOOPTIMIZE prevents the generation of a globally optimized object file (for FORTRAN programs only).
Default

/NOSEARCH requires all modules in the object file library (the file accompanied by this switch in the command line) to be loaded even if they are not called by your program. Cancels the /SEARCH switch.
Default

/NOSTAY stops the compiler from being placed in a background fork. Use when /STAY is set as a default for the compiler.

/NOSYMBOLS prevents a symbol table from being loaded along with the object file.

/NOWARNINGS prevents display of warnings for nonfatal errors.

/OPTIMIZE calls for generation of a globally optimized object file, that is, one that runs as quickly as possible (for FORTRAN programs only, and only if you do not also give the /DEBUG switch).

/PASCAL compiles the file using the PASCAL compiler.
Default for files of type .PAS

/RELOCATABLE identifies the input file as an object file (regardless of its extension) and prevents compilation of the source file, /RELOCATABLE switch forcing use of an existing object file

even if the object file is out of date.
Default for files of type .REL

/SAIL — compiles the file using the SAIL compiler.
Default for files of type .SAI

/SEARCH — requires that the object file library (the file accompanied by this switch in the command line) be searched for modules called by your program or by a program subroutine. Only these modules are loaded, along with modules called from the system libraries, which are always searched.

/SIMULA — compiles the file using the SIMULA compiler.
Default for files of type .SIM

/SNOBOL — compiles the file using the SNOBOL compiler.
Default for files of type .SNO

/STAY — returns your terminal to TOPS-20 command level so that you can perform other work while the system continues to load your program. You immediately receive the TOPS-20 prompt (@ or $), and can then issue any user command. Be careful not to send incorrect data to programs expecting terminal input. (Refer to the CONTINUE command, Restrictions: Programs Competing for Terminal Input. This switch saves you from having to: issue a ^T to make sure loading has begun; give a ^C to halt the job; and issue a CONTINUE /STAY command to remain at command level during loading.

/SYMBOLS — loads a symbol table along with the object file (helpful for debugging a program).
Default

/WARNINGS — displays warnings for nonfatal errors.
Default

Characteristics

Compiling New Sources Only

Before loading programs, the system ordinarily compiles any specified source (and only those sources) whose write date is more recent than that of the object file of the same name. You can override this action with the /COMPILE or /RELOCATABLE switch.

Using Standard File Types

If you specify source files with standard types (.FOR, .MAC, for example) in a LOAD command, the system automatically calls the appropriate compiler when compilation is necessary. If you specify source files by filename only, the system searches your connected directory for a file of this name and a standard type. To load programs from sources that have nonstandard file types, give a switch to indicate the proper compiler (/FORTRAN, /MACRO, /COBOL, or /ALGOL). A switch will take precedence over a standard file type if they indicate different languages. If no compiler is indicated with either a switch or a standard file type, the FORTRAN compiler is used.

Default Switches Not Passed to Compiler

Only switches specified in a LOAD-class command are passed to the compiler; default switches are not passed. Instead, the system assumes that the defaults for the compiler are the same as the defaults for the LOAD-class command.

Hints

Commas Between Filespecs

If you give two or more filespecs separated by commas as arguments to LOAD, the loaded programs exist in memory at the same time and will operate as a single program. You can use this feature to substitute one module for another under varying conditions or for different applications.

Plus Signs Between Filespecs

If you give two or more source filespecs separated by plus signs (+) as arguments to LOAD, they are compiled together as if they were a single file. Their object module is stored under any filename given as the "object" argument of the command, or (if none) under the last filename in the group and file type .REL.

Indirect Files as Arguments

You can store arguments (source and object filespecs, switches) of a LOAD command in an indirect file, and specify them by typing an at sign (@) and its filespec as a LOAD command argument.

Establishing Default Arguments with the SET Command

You can issue the SET DEFAULT COMPILE-SWITCHES command to set up default global arguments to the LOAD command. Insert this SET command in your COMAND.CMD file to change your own defaults permanently.

Running Link Directly

The LOAD command automatically runs LINK, the system's linking loader, but if you require more control of the loading process you can run LINK directly. See the TOPS-20 LINK Reference Manual.

Using GET Instead of LOAD

If you have used the SAVE command to save your programs in executable format, you can use the GET command instead of LOAD to place them in memory. This is a faster and less expensive means of loading programs into memory.

Wildcards Illegal with LOAD

The LOAD command does not accept wildcard characters (* and %) in a file specification.

Warning - Generation Numbers, Long Filespecs

You must not give generation numbers when specifying source or object files; the system automatically uses the highest generation. Also, most compilers require filenames of 6 or fewer characters and file types of 3 or fewer characters.

Related Commands

COMPILE, EXECUTE, and DEBUG	other LOAD-class commands for performing related functions
SAVE	for saving the loaded program in an .EXE file
START	for starting the loaded program
SET DEFAULT COMPILE-SWITCHES	for establishing default switches for LOAD-class commands

INFORMATION DEFAULTS COMPILE-SWITCHES	for examining default switches established for LOAD-class commands

Examples

1. Load an object file into memory.

```
@LOAD LSTSQ.REL
LINK:   LOADING

EXIT
```

2. Load the same program, allowing the system to update the object file if necessary.

```
@LOAD LSTSQ/FORTRAN
FORTRAN: LSTSQ
MAIN.
LINK:   LOADING

EXIT
```

3. Load a MACRO program and request a loader map or memory map. (Notice the filename of this map.)

```
@LOAD TEST2/MAP
MACRO:  FT
LINK:   LOADING

EXIT
@TDIRECTORY
                WRITE


  PS:<LATTA>
 FT.MAP.1        6-APR-85 15:23:17
 TEST2.REL
```

4. Load a COBOL program, forcing a new compilation that includes only the required modules. Request a map.

```
@LOAD /MAP TEST1/COMPILE, COBLIB/SEARCH
COBOL:  DBL     [TEST1.CBL]
LINK:   LOADING

EXIT
```

5. Compile a program. Then load it, requesting a cross-reference listing this time. Finally, save the program

in executable format.

```
@COMPILE TEST1/COBOL
COBOL:  DBL     [TEST1.CBL]
@LOAD /COMPILE/CREF
COBOL:  DBL     [TEST1.CBL]
LINK:   LOADING

EXIT
@SAVE
 TEST1.EXE.1 SAVED
```

6. Combine two FORTRAN sources into an object program under a new name. Start this program.

```
@LOAD LSTSQ+ABRR REGRES
FORTRAN: LSTSQ
MAIN.
MAIN.
LINK:   LOADING

EXIT
@START
```

7. Create an indirect file, and use it to load several modules at once. Request cross-reference files, then give the CREF command to turn these into listings.

```
@CREATE SERVTT.CMD
INPUT: PS:SERVTT.CMD.1
00100   HJRAD/COMPILE, FORLIB/SEARCH
00200   HJVTT/COMPILE, FORLIB/SEARCH
00300   HJINI/RELOCATABLE
00400   $
*E

[SERVTT.CMD.1]
@LOAD /CREF @SERVTT.CMD
FORTRAN:HJRAD
MAIN.
FORTRAN: HJVTT
MAIN.
LINK:   LOADING

EXIT
@CREF
CREF:   HJRAD
CREF:   HJVTT
@
```

2.48 LOGIN

Begins your timesharing job and connects you to your log-in directory.

Format

@LOGIN /FAST (USER) **name** (PASSWORD) **password** (ACCOUNT)**account** - (SESSION REMARK) **remark**

where:

name	is your user name.
pwd	is your secret password (which is not printed on your terminal).
acc	is an account name or number that you are authorized to use.
remark	is an optional remark of up to 39 characters that identifies the terminal session for accounting purposes. Check with INFORMATION JOB-STATUS. Change with SET SESSION-REMARK.
/FAST	is an optional switch that prevents the following: processing of your LOGIN.CMD and COMAND.CMD files and the system's LOGIN.CMD and COMAND.CMD files, printing of system mail, and printing of the notice of new mail. (Your system manager may remove this switch from your system.)

Output

Acknowledgement of Valid Login

The system acknowledges a valid LOGIN command by printing your job number, terminal number, and the current date and time. In addition, it prints the date and time of your last login. You can use this information to determine if another user has learned your password and logged in to your account since the last time you logged out.

Note that a batch job automatically logs in and logs out of your account. The batch login sets the date and time of your last login and should not be confused with illegal access to your account.

Notice of User Mail and System Mail

When you log in, the system notifies you if another user has sent you a message with one of the system mail programs. The system then lists any system mail (mail sent by privileged users to all users) that has accumulated since your last login. Note that this mail appears in the log file if a batch job is run for you between the time the mail was sent and the time you logged in.

Output from Command Files

After a successful LOGIN, the system processes the LOGIN.CMD and COMAND.CMD files in the directory defined by logical name SYSTEM: and the command files in your login directory. The files are processed in this order:

1. SYSTEM:LOGIN.CMD
2. LOGIN.CMD
3. SYSTEM:COMAND.CMD
4. COMAND.CMD

The system displays any output from the commands in these files on your terminal. After execution of each command file, the system displays the message "End of file-name.CMD". If the last command in the command file is a TAKE command with no arguments, this message is not displayed.

Characteristics

Getting the Attention of the System

Before logging in, you may have to press any alphanumeric or special character to display the system herald or greeting and the @ prompt necessary for typing the LOGIN command.

If you are dialing in by telephone to a line declared autobaud by the system manager, this initial character enables the system to determine your terminal's speed setting, as long as the speed is 300, 1200, 1800, 2400, 4800, or 9600. Type a second character if the terminal's speed is 110 or 150. If your initial character(s) fails to get the system identification message, press the BREAK key twice, followed by another character(s).

Rights, Capabilities, and Charges

The LOGIN command gives you ownership rights to your log-in directory, and any group rights established for you on the public structure (usually named PS:). In addition, you are granted whatever capabilities (for example, Maintenance,

Wheel) have been awarded to you, and can be sure that any charges you incur for the use of system resources, such as CPU time or the batch and printing systems, will be recorded to your user name.

Hints

Commands in Files Executed at Log-in Time

For Affecting Entire Session or Current Level Only

Commands that affect your entire job, for example, TERMINAL and DEFINE, belong in LOGIN.CMD. Commands that affect only the current level of TOPS-20, for example, many SET commands, must be put into COMAND.CMD if you want them to be executed automatically after every PUSH command as well as after LOGIN.

For Affecting Batch Jobs

As soon as one of your batch jobs logs in, the system processes the command files in the directory defined by logical name SYSTEM: and the command files in your login directory. The files are processed in this order:

1. SYSTEM:BATCH.CMD
2. BATCH.CMD
3. SYSTEM:COMAND.CMD
4. COMAND.CMD

Note that certain parameters of the batch job, for example, its time limit and the name of its log file, have already been set before these commands are executed. Such parameters are set either to values specified by switches in the SUBMIT command that starts the batch job, or to default values in effect for the job issuing this SUBMIT command. See also Hints - For Affecting Nested Batch Jobs, below.

Avoiding Duplicate Commands in Command Files

After executing a SYSTEM: command file, the system executes the file of the same name in your login directory. The SYSTEM: command files may contain commands that you already have in your own command files. To avoid executing the same commands twice, remove duplicate commands from your command files. To display a SYSTEM: command file, give the command TYPE SYSTEM:file-name.CMD.

For Affecting Nested Batch Jobs

By placing a SET DEFAULT SUBMIT command in your BATCH.CMD file, you cause these defaults to be in effect for a nested batch job, (a batch job started by a SUBMIT command within the control file of another of your batch jobs).

A Final TAKE Command

To suppress the display of the message "End of file-name.CMD" after execution of a command file, make the last command in the file a TAKE command with no arguments.

Simplifying Log-ins

By using the SET DIRECTORY ACCOUNT-DEFAULT command you cause subsequent LOGIN commands to require just your user name and password.

Special Cases

Commands You Can Issue Before Log-in

You can give these commands and arguments before logging in:

Command	Arguments
ATTACH	
BREAK	
DAYTIME	
INFORMATION	AVAILABLE
	COMMAND-LEVEL
	MAIL
	TERMINAL-MODE
	VERSION
LOGOUT	
SET	LATE-CLEAR-TYPEAHEAD
	TIME-LIMIT
SYSTAT	(except with subcommands LPT or OUTPUT)
TERMINAL	
UNATTACH	

Logging in to PTYs

You do not need to give a password when logging in under your own user name to a PTY (pseudo-terminal).

Must Log In Within Five Minutes

If you do not log in within five minutes of your initial CTRL/C, your job will be logged out automatically and you will have to type CTRL/C again.

Logging in to Last Available Job Slot

If you attempt to log in to the last available job slot, the system will not log you in but will send you an error message instead. This job slot is intended for users who wish to attach detached jobs using the ATTACH command. To log in a new job you must wait until a current user logs out.

Related Commands

ATTACH	for joining to your terminal a job that has already been logged in
INFORMATION DIRECTORY	for displaying the date and time that you started the current terminal session with LOGIN.
LOGOUT	for ending your timesharing job
SET ACCOUNT	for changing your account during a terminal session
SET DIRECTORY ACCOUNT-DEFAULT	for specifying a default account for subsequent log-ins
SET SESSION-REMARK	for making or changing your session remark during a terminal session

Examples

1. Log in, using account 341 and automatically executing the system LOGIN.CMD file and your LOGIN.CMD file.

```
@LOGIN C.RYDER ___ 341
 Job 39 on TTY41 GIDNEY:: C.RYDER (CTM) 8-Mar-89 11:04:21,
Last interactive login 7-Mar-89 08:32:15
Last non-interactive login 7-Mar-89 08:32:15
 End of SYSTEM:LOGIN.CMD.1
 End of LOGIN.CMD.1
```

2. Log in using the default account number and the /FAST switch.

```
@LOGIN /FAST C.RIDER ___
 Job 39 on TTY41 LAT1:LAT127(LAT) 8-AUG-88 11:10:34
 Last interactive login 8-Aug-88 11:04:21
 Last non-interactive login 8-Aug-88 11:04:21
```

3. Type a character to get the TOPS-20 herald, then log in, using account 341 and inserting a session remark. Give INFORMATION JOB-STATUS as your first command, to see this session remark.

```
     Unauthorized Access is Prohibited

 BOSTON (KL2871), Development System, TOPS-20 Monitor 7(10)
@LOGIN URQUHART ___ 341 DEBUG ACCOUNT.PAS
 Job 42 on TTY29 LAT64:242(LAT) 8-Mar-90 09:15:15
 Last interactive login 7-Mar-90 09:20:32
 Last non-interactive login Never
@INFORMATION JOB-STATUS
 Host AURORA, Job 42, TTY29 LAT64:242(LAT)
 User URQUHART, FTN:<URQUHART>
 Account 341 Session Remark:DEBUG ACCOUNT.PAS
```

2.49 LOGOUT

Ends a timesharing job.

Format

@LOGOUT /FAST n

where:

/FAST is an optional switch that prevents processing of your LOGOUT.CMD and the system's LOGOUT.CMD files.

n is an optional job number. Specify n only when logging out a job other than your attached job.

Output

System Use Under Current Account

The system acknowledges a valid LOGOUT command by printing your job number, user name, current account, terminal number, and the current date and time. Then it shows the total amount of CPU time you used during the terminal session and the total length of time you were logged in, followed by the account of CPU time used under the current account and the length of time you were logged in under this account.

LOGOUT.CMD Output

Before logging you out, the system processes the commands in your login directory's LOGOUT.CMD file and the system's LOGOUT.CMD file and displays any output from these commands. Then, after execution of each file, the system displays the message "End of LOGOUT.CMD", unless the last command in the file is a TAKE command.

Characteristics

Expunging Your Log-in and Connected Directories

Before logging you out, the system expunges any deleted files from your log-in and connected directories, and prints a message if either directory is still exceeding its assigned permanent disk quota.

Logging Out Other Jobs

By specifying a job number you can log out any other job logged in under the same user name as your attached job. A user with Wheel or Operator capabilities enabled can log out any job on the system. Whenever you log out another job, the system prints the job's user name, terminal number, and current program. You must then confirm the LOGOUT command with an extra RETURN.

When a job is logged out by another job, the logout command files are not processed.

Hints

A Final TAKE Command

To suppress the display of the message "End of LOGOUT.CMD" after execution of your LOGOUT.CMD file, make the last command in the file a TAKE command with no arguments. Be sure the file contains only one RETURN after the TAKE command.

Errors in LOGOUT.CMD

If there is an error in a command in your LOGOUT.CMD file, the system processes the commands up to the one in error and cancels the LOGOUT command. To log out, correct the error in the command file or give the LOGOUT command with the /FAST switch.

Effect on Memory and Terminal

The LOGOUT command clears memory and leaves your terminal in the state before log-in. LOGOUT n does not affect memory and leaves your terminal at TOPS-20 command level.

Related Commands

DETACH	for disengaging a job from your terminal without ending the job
LOGIN	for beginning your timesharing job
UNATTACH	for disengaging a job from another terminal without ending the job

Examples

1. Log out your job.

```
@LOGOUT
 End of SYSTEM:LOGOUT.CMD.2
 End of LOGOUT.CMD.5
Killed Job 18, User C.RYDER, Account 341, TTY 233,
 at 8-Mar-84 16:25:46, Used 0:0:5 in 1:2:16
```

2. Log out your job, receiving a warning message that your directory is over its storage quota.

```
@LOGOUT
 <URQUHART> Over permanent storage allocation by 8 page(s).
Killed Job 39, User URQUHART, Account 341, TTY 41
  at  8-Mar-84 16:33:12,  Used 0:0:1 in 0:1:5
```

3. Check what jobs are logged in under your user name. Log out a detached job and verify that it is gone, then log out your attached job.

```
@SYSTAT WALKER
  18   DET  EXEC    WALKER
  21*   31  SYSTAT  WALKER
@LOGOUT  18
User WALKER, Detached, running EXEC
[Confirm]
@SYS WALKER
  21*   31  SYSTAT  WALKER
@LOGOUT
Killed Job 43, User WALKER, Account 341, TTY 226,
  at  8-Mar-84 16:35:16,  Used 0:0:1 in 0:1:1
```

2.50 MERGE

Places an executable program into the current fork, combining it with whatever program (if any) is already there.

Format

@**MERGE** (PROGRAM) **filespec /switch**

where:

filespec — is the file specification of any executable program.
Default file type - .EXE

/switch — is one or more of the following:

- /OVERLAY — allows pages of the DDT program to be loaded over pages occupied by the existing program in memory.
- /USE-SECTION:n — specifies the memory section (from 0 to 37 octal) into which your program is to be merged. You can use this switch only if your program can be contained in one section.

Characteristics

Executable Files Only

If a program you try to merge is not in executable format, you may get an immediate error message (that is, "?UNEXPECTED END-OF-FILE TRAP...") or a delayed one (that is, "?ENTRY VECTOR LENGTH IS NOT LESS THAN 1000") after the merge. In either case, be sure that you have specified an executable program before investigating further. The MERGE command does not alter the entry vector if the file being merged is in the proper .EXE format.

Existing Pages Not Overlaid

If there is a program already in memory when you give the MERGE command, and pages of the new program overlay it, the new program is not placed into memory, and the system prints the error message, "?Illegal to overlay existing pages." To force the existing pages to be overlaid, reissue the MERGE

command using the /OVERLAY switch.

Effect on Memory

The MERGE command combines the specified program with the program in the current fork. It does not affect the contents of the current fork unless you specify the /OVERLAY switch.

Related Commands

INFORMATION MEMORY-USAGE	for examining the contents of memory
GET	for putting a saved (executable) file into memory
SAVE	for storing a copy of the program in the current fork in a file in executable format
START	for starting the program in memory

Examples

1. Merge an executable program into memory.

   ```
   @MERGE TESTF1.EXE
   ```

2. Place an executable system program in memory, then merge a system debugging program with it. Give INFORMATION MEMORY-USAGE commands to verify that both programs are intact.

   ```
   @GET SYS:DUMPER
   @INFORMATION MEMORY-USAGE

   34. pages, Entry vector loc 4715 len 3

     Section 0     R, W, E,  Private
   0        RANDOM:<NEXT-RELEASE>DUMPER.EXE 4  1   R, CW, E
   4-44     RANDOM:<NEXT-RELEASE>DUMPER.EXE.4  2-42   R, CW, E
   @MERGE SYS:UDDT
   @INFORMATION MEMORY-USAGE

   45. pages, Entry vector loc 4715 len 3

     Section 0     R, W, E,  Private
   0        RANDOM:<NEXT-RELEASE>DUMPER.EXE.4  1   R, CW, E
   ```

```
4-44     RANDOM:<NEXT-RELEASE>DUMPER.EXE.4  2-42   R, CW, E
764-767  RANDOM:<NEXT-RELEASE>UDDT.EXE.1  1-4   R, CW, E
770-776  RANDOM:<NEXT-RELEASE>UDDT.EXE.1  5-13   R, E
```

2.51 MODIFY

Adds or changes switches for a request placed in a batch or output queue.

Format

@MODIFY (REQUEST TYPE) **queue** (ID) **identifier /switch(es)**

where:

queue — is the waiting list in which you placed the original request, chosen from the following list:

BATCH	for requests made using the SUBMIT command
CARDS	for requests made using the PUNCH CARDS command
PAPER-TAPE	for requests made using the PUNCH PAPER-TAPE command
PLOT	for requests made using the PLOT command
PRINT	for requests made using the PRINT command

In the switch summary and descriptions, the word Output in the column headed Applicable Queues means all queues except the batch queue.

identifier — is **one** of the following:

request ID number	the unique identifier assigned by the system to your request. This is the number appearing under the heading "Req#" in the list of requests shown by the INFORMATION BATCH-REQUESTS or INFORMATION OUTPUT-REQUESTS command.
jobname	the jobname of the request, either the first six characters of the first filename in the request, or the argument you supplied to a /JOBNAME switch when making the

original request. This is the name appearing under the heading "Job Name" in the list of requests shown by the INFORMATION BATCH-REQUESTS or INFORMATION OUTPUT-REQUESTS command.

/JOBNAME:jobname switch showing the jobname of the request to modify. You can specify a particular jobname when making the original request. See Special Cases - /JOBNAME Switch, below.

/SEQUENCE:sequence number
switch showing the sequence number of the request to modify. You can specify a particular sequence number when making the original request.

Use an asterisk (*) as identifier to modify all your requests in the specified queue.

/switches are keywords, chosen from the list below, specifying the parameter you want to change (and, where applicable, the new value of this parameter)

Summary of MODIFY Command Switches

Switch	Applicable Queues		
/AFTER:date and/or time			All
/BEGIN:n			All
/CARDS:n		BATCH	
/COPIES:n			Output
/DELETE	PRINT	BATCH	
/DEPENDENCY-COUNT:n		BATCH	
/DESTINATION-NODE:node name::			All
/FEET:n		BATCH	
ASCII	PRINT		
COBOL	PRINT		
/FILE:ELEVEN	PRINT		
FORTRAN	PRINT		
/FORMS:forms name			Output
/GENERIC			Output
/HEADER			Output
/JOBNAME:jobname			All
/LIMIT:n	PRINT		

/LOWERCASE	PRINT	
/MODE:output mode		Output
/NOHEADER		Output
/NOTE:12-character message		Output
/OUTPUT ALWAYS	BATCH	
/OUTPUT ERRORS	BATCH	
/OUTPUT NOLOG	BATCH	
/PAGES:n	BATCH	
/PRIORITY:n		All
/PRESERVE		All
/PROCESSING-NODE:node name	BATCH	
/REMOTE-PRINTER:	PRINT	
/REPORT:title	PRINT	
/RESTARTABLE:NO		
/RESTARTABLE:YES	BATCH	

Switch	Applicable Queues
/SEQUENCE:n	All
/SPACING:SINGLE	PRINT
/SPACING:DOUBLE	PRINT
/SPACING:TRIPLE	PRINT
/TIME:hh:mm:ss	All
/TPLOT:n	BATCH
/TPLOT:0 or NO	BATCH
/UNIQUE:1 or YES	BATCH
/UNIT:octal number	Output
/UPPERCASE	PRINT
/USER:user name	All

MODIFY Command Switches

	Applicable Queues	
/AFTER:date and/or time, **or** day of week (or TODAY) and/or time	All	ensures that the request will not be processed until after he revised date and/or time specified. NOV-12-79, and 18:00 illustrate two arguments to this switch. If you give both date and time, separate them with a space. When given alone, the time may be preceded by a plus sign

			(+), which will delay processing by the indicated length of time from the present. Alternatively, you can give a day of the week (for example, MONDAY) or TODAY as argument; then the job will not be printed until the beginning of the following day. If you follow this argument with a plus sign and a time, the job will be further delayed by this amount.
/BEGIN:n		All	gives the decimal line number of the control file at which processing is to begin (for BATCH), or the decimal page number of the file at which the output is to begin (for CARDS, PAPER-TAPE, PLOT, and PRINT)
/CARDS:n	BATCH		specifies the decimal number of spooled cards the job is allowed to punch
/COPIES:n		Output	tells how many copies of the file to produce
/DELETE		All	deletes the file after processing. Opposite of /PRESERVE.
/DEPENDENCY-COUNT:n	BATCH		sets the request's dependency count to the new value n. This switch can also be followed by a signed value, such as +n or -n, which will increase or decrease the old value by the specified amount.

		A batch request is not processed until its dependency count is 0. See the TOPS-10/20 Batch Reference Manual for more information about dependency counts.
/DESTINATION-NODE:node-name	All	specifies the node on whose line printer the log file of your batch job is to be printed (for BATCH), or the node on whose line printer or other output device your request is to be processed (CARDS, PAPER-TAPE, PLOT, and PRINT). Two colons (::) following the node name are optional.
/FEET:n	BATCH	specifies the decimal number of feet of spooled paper tape the job is allowed to punch.
/FILE:ASCII COBOL ELEVEN FORTRAN	PRINT	specifies that the file consists of ASCII text, or COBOL SIXBIT text; or (ELEVEN) contains four eight-bit bytes in each 36-bit word; or is FORTRAN ASCII text, where column 1 of each line is interpreted as a carriage control character.
/FORMS:forms name	Output	specifies, in six or fewer characters, new forms (determining the size of banner, header, and trailer sections; the paper color, width, and weight; vertical format, carriage control tape, the number of plotter steps per inch,

		etc.) to use with the job
/GENERIC	Output	allows the output to be produced on any available device. Use along with argument PRINT to cancel the /LOWERCASE or /UPPERCASE switch, or with PLOT, PRINT, CARDS, or PAPER-TAPE to cancel the /UNIT switch.
/HEADER	Output	causes a header section containing the jobname to be plotted, printed, or punched before the file itself is produced.
/JOBNAME:jobname	All	does not change the jobname, but specifies which job to modify. Same as jobname in "identifier" argument.
/LIMIT:n	Output	places a new limit of n cards, feet, or pages on the output of the job.
/LOWERCASE	PRINT	specifies that the file is to be produced on a line printer capable of printing lowercase characters.
/MODE:ASCII /MODE:BCD /MODE:BINARY /MODE:IMAGE	CARDS	designates the mode for punching the file onto cards. See the /MODE switch in the PUNCH command description for details.
/MODE:ASCII /MODE:BINARY /MODE:IMAGE /MODE:IMAGE-BINARY	PAPER-TAPE	designates the mode for punching the file onto paper tape. See the /MODE switch in the

			PUNCH command description for details.
/MODE:ASCII /MODE:BINARY /MODE:IMAGE		PLOT	designates the mode for plotting the file. See the /MODE switch in the PLOT command description for details.
/MODE:ARROW /MODE:ASCII /MODE:OCTAL /MODE:SUPPRESS		PRINT	designates the mode for printing the file. See the /MODE switch in the PRINT command description for details.
/NOHEADER		Output	prevents a header section containing the jobname from being produced before the file is produced.
/NOTE:message		Output	labels the header section of output (the section displaying the jobname) with a message or notation of up to 12 characters. The message must be enclosed in double quotation marks if it contains spaces or punctuation characters.
/OUTPUT:ALWAYS /OUTPUT:ERRORS /OUTPUT:NOLOG		BATCH	says whether you want the log file to be printed always, or only in the case of unhandled errors occurring within the job, or never. No matter which option you choose, the log file is always created.
/PAGES:n		BATCH	specifies the decimal number of spooled line printer pages the job is allowed to print.

/PRESERVE	All	saves the file after it is processed. Opposite of /DELETE.
/PRIORITY:n	All	assigns a new number n reflecting the urgency of the request. This n must be from 1 to 63, with larger numbers receiving earlier treatment.
/PROCESSING-NODE:node name::	BATCH	specifies the IBM host system on whose CPU the JCL batch job is to be run. The node name must be of six or fewer characters and must be followed by two colons (::).
/REMOTE-PRINTER: n	PRINT	specifies the name of a remote print queue to print the file.
/REPORT:title	PRINT	scans your files and processes only those lines whose first characters are the title you give. This title can contain up to 12 characters (including the quotation marks that must enclose the title if it contains spaces). The switch is used along with the COBOL report writer.
/RESTARTABLE:NO /RESTARTABLE:YES	BATCH	specifies whether the job should be started again if the system crashes and restarts.
/SEQUENCE:n	All	does not change the sequence number of the job but rather specifies which job to modify. Giving this switch is an

		alternative to supplying a request ID as the request identifier when you have several jobs with the same jobname (if you supply only the jobname to identify the job, the MODIFY command affects all of them).
/SPACING:DOUBLE /SPACING:SINGLE /SPACING:TRIPLE	PRINT	determines the spacing between printed lines.
/TIME:hh:mm:ss	BATCH	revises the limit for the maximum amount of CPU time available to the job; given in hours, minutes, and seconds.
/TPLOT:n	BATCH	limits to n the maximum number of minutes of spooled plotter time allowed for the job.
/UNIQUE:NO or 0 /UNIQUE:YES or 1	BATCH	changes your declaration, if two or more jobs are submitted from the same connected directory, whether they must run at separate times.
/UNIT:octal number	Output	directs your request to the line printer of the specified octal unit number.
/UPPERCASE	PRINT	specifies that the file is to be produced on a line printer that uses uppercase characters only.
/USER:user name	PRINT, BATCH	specifies the user whose request is to be modified; for privileged users only. This switch is required to modify a request from a user

other than yourself.

Characteristics

MODIFY Effective Only Before Processing

The MODIFY command affects a batch or output request only before processing has begun. After processing has begun, you can only cancel the request with the CANCEL command, and then make a new request.

Hints

Using the /DEPENDENCY-COUNT Switch

You can use the /DEPENDENCY-COUNT switch to specify the order in which your batch jobs are processed. Set the dependency count of all but the first job to some positive value when you submit them, and include MODIFY commands in each job's control file to bring the next job's dependency count to 0 at the appropriate time. See Example 4.

Special Cases

/JOBNAME Switch

In the singular case when you want to modify several queue requests of the same jobname using only one command, and that jobname is purely numerical (for example, 5045), you must use the /JOBNAME:jobname switch as second argument to the MODIFY command. Do not also give the request ID or jobname as a command argument if you give the /JOBNAME:jobname switch.

Related Commands

CANCEL	for removing batch and output requests
INFORMATION BATCH-REQUESTS	for examining entries in the batch queue
INFORMATION OUTPUT-REQUESTS	for examining entries in the output queues
PLOT	for placing requests in a plotter output queue

PRINT	for placing requests in a line printer output queue
PUNCH	for placing requests in a card punch or paper tape punch output queue
SUBMIT	for placing requests in the batch input queue

Examples

1. Modify a batch request (of jobname ARTIFI) to make it start more quickly.

   @MODIFY BATCH ARTIFI /PRIORITY:63
   [1 Job modified]

2. Modify a print request (of jobname PHIAL) to include a note on the header page.

   @MODIFY PRINT PHIAL /NOTE:"DUE: 11/4"
   [1 Job modified]

3. Modify one job of several having the same jobname.

   @INFORMATION OUTPUT-REQUESTS /USER

   ```
   Printer Queue:
   Job Name  Req#  Limit            User
   --------  ----  -----  ------------------------
     PRTSK    226   27    LATTA /After: 8-Nov-84 17:00
     PRTSK    236   27    LATTA /After: 8-Nov-84 17:00
     PRTSK    237   27    LATTA /After: 8-Nov-84 17:00
     PRTSK    238   27    LATTA /After: 8-Nov-84 17:00
     TESTF1   219   54    LATTA /Forms:NARROW
               /After:8-Nov-84 17:00
   There are 5 jobs in the queue (none in progress)
   ```

   @MODIFY PRINT 237 /AFTER:18:00
   [1 Job modified]
   /new

4. Use the TYPE command to examine some of your control files. (Notice the use of the MODIFY command within these files to ensure that they are processed in a certain order when submitted together.) Submit these three control files and verify their placement in the batch input queue.

   @TYPE ARVM%.CTL

```
 ARVM1.CTL.2

@RUN TESTF1
@PRINT TESTF1.RSM
@MODIFY BATCH ARVM2 /DEPENDENCY-COUNT:0

 ARVM2.CTL.2

@RUN TESTF2
@PRINT TESTF2.RSM
@MODIFY BATCH ARVM3 /DEPENDENCY-COUNT:0

 ARVM3.CTL.2

@RUN TESTF3
@PRINT TESTF3.RSM
@PRINT SUMJOB.RSM
@SUBMIT /AFTER:17:00 ARVM1
[Batch job ARVM1 queued, request-ID 240, limit 0:05:00]
@SUBMIT /DEPENDENCY-COUNT:1 ARVM2
[Batch job ARVM2 queued, request-ID 241, limit 0:05:00]
@SUBMIT /DEPENDENCY-COUNT:1 ARVM3
[Batch job ARVM3 queued, request-ID 242, limit 0:05:00]
@INFORMATION BATCH-REQUESTS /ALL/USER

Batch Queue:
Job Name  Req#  Run Time            User
--------  ----  --------  ------------------------
  ARVM1    240  00:05:00  LATTA /After: 8-Nov-84 17:00
          /Uniq:Yes  /Restart:No   /Assist:Yes  /Seq:1804
  ARVM2    241  00:05:00  LATTA /Dep:1  /Uniq:Yes  /Restart:No
          /Assist:Yes  /Seq:1805
  ARVM3    242  00:30:00  LATTA /Dep:1  /Uniq:Yes  /Restart:No
          /Assist:Yes  /Seq:1806
There are 3 jobs in the queue (none in progress)
```

2.52 MOUNT

Requests that a specified file structure or magnetic tape set be made available for your job's use.

Format

@MOUNT medium (NAME) **dev: /switch(es)**

where:

medium — is one of the following:

STRUCTURE - for mounting file structures (disk packs)

TAPE - for mounting magnetic tapes

dev: — is either the structure identification (or alias), or the tape setname. The colon after the device name is optional.

/switches — are keywords, chosen from the list below, indicating your choice of MOUNT command options.

Summary of MOUNT Command Switches (defaults in boldface)

/CHECK-SETNAME

/DENSITY:200
556
800
1600
6250
SYSTEM-DEFAULT

/DRIVE-TAPE:7-TRACK
9-TRACK

/LABEL-TYPE:ANSI
BYPASS
EBCDIC
TOPS-20
UNLABELED

/NEW
/NOUNLOAD
/NOWAIT

/OPERATOR	
/PROTECTION:octal protection code	**Default** code - 770000
/READ-ONLY	**Default** - unless /NEW or /SCRATCH specified
/REMARK:119-character remark	
/SCRATCH	
NUMBER number	
/START:VOLID volid	**Default** number - 1
/STRUCTURE-ID:structure identification	
/VOLIDS:list of volids	
/WRITE-ENABLED	**Default** - if /NEW or /SCRATCH specified

The switches /NOWAIT and /REMARK are useful with either the STRUCTURE or TAPE medium, while /STRUCTURE-ID is for STRUCTURE only; the other switches are for TAPE only.

MOUNT Command Switches

/CHECK-SETNAME	ensures that the setname of the mounted tapes matches the setname you specify as the "dev:" argument to the MOUNT command; otherwise an error will be generated. For labeled tapes only.
200 556 800 /DENSITY:1600 6250 SYSTEM-DEFAULT	specifies the density, in bits per inch, at which the tape set is to be read or written. Densities 200 and 556 are for unlabeled tapes only. SYSTEM-DEFAULT, one of the values shown (usually 1600), is established at system start-up time.
7-TRACK /DRIVE-TYPE:9-TRACK	specifies the type of drive on which the tape set is to be mounted. Labeled tapes must be mounted on 9-track drives.
ANSI BYPASS /LABEL-TYPE:EBCDIC TOPS-20	tells the system to read and write the tape set according to the

UNLABELED	specified label standard: ANSI; EBCDIC - IBM TYPES (IN READ-ONLY MODE); TOPS-20 - a superset of ANSI used in TOPS-20 systems; UNLABELED - for unlabeled tapes only. BYPASS (for privileged users only) lets you read and write any tape, labeled or unlabeled, without any label processing.
/NEW	tells the system that you are creating a new file set on an existing tape set, whose setname is then changed to be the name you specify as the dev: argument to the MOUNT command. (If the tape set has more than one volume, remember to specify their volids using the /VOLIDS or /OPERATOR switch.) The /CHECK-SETNAME and /READ-ONLY switches are ignored if present, and /WRITE-ENABLED is assumed. Do not give the /START switch if you give /NEW.
/NOUNLOAD	asks the system not to unload a volume (reel) of tape from its tape drive when the drive is released by a volume switch (change of volumes required by a read or write operation) or DISMOUNT command. Use this switch to facilitate processing when sufficient drives are available.
/NOWAIT	tells the system to return your terminal to TOPS-20 command level as soon as you give the MOUNT command, and to send a message to your terminal when the request has been processed. Otherwise, your terminal waits for the message.
/OPERATOR	asks the operator to specify to the system the volids of the tape set you wish to mount. Do not use if you have given the /VOLIDS switch.

/PROTECTION:code

specifies a 6-digit octal protection code for new volumes of tape written during the current mount request. The owner always has full access to his tapes, so the first two digits are always interpreted as "77"; also, user groups and directory groups have no effect on tape access, so the middle two digits are always interpreted as "00". Therefore, although six digits can be specified, only the last two digits affect the tape's protection code.

(If you specify only two digits, these will be used as the last two digits of the protection code.) These two digits should be the sum of the values corresponding to the modes of access you want to allow, chosen from the following list:

- 40 - read files in the file set
- 10 - overwrite or modify files in the file set
- 04 - append files to the end of the file set

For tapes of label-type TOPS-20 only.

Default code - 770000

/READ-ONLY

ensures that all volumes in the tape set will be mounted without write rings, to prevent accidental erasures.

Default except when /NEW or /SCRATCH is specified

/REMARK:"remark"

sends the specified remark to the operator when he is notified of your mount request. The text of the remark must be enclosed in quotation marks (" ") and can be up to three, 80-character lines long (including the MOUNT command line). Note that while the entire remark is displayed on the operator's terminal, only the first line appears in the INFORMATION MOUNT-REQUESTS display.

For structures, the remark will be sent only if the structure must be put on line or physically mounted to satisfy your mount request.

/SCRATCH

same as /NEW, except that the volumes in the file set you create will be drawn from the pool of scratch tapes (tapes not presently owned by a particular user), rather than from volumes you specify. Use this switch to create a new file set when you are not supplying the volumes of tape to be used.

/START:VOLID volid
NUMBER number

tells the system which volume (reel) of tape to mount first when satisfying your request. (You must also give the /VOLIDS switch, specifying the group of volumes you will be using.) Use the NUMBER argument to give the order of this volume within the group (e.g., 1 for first, 2 for second), or give the VOLID argument to repeat the volid explicitly. You can use this switch to save time and expense when you know which volume you will be using first.

Default - NUMBER 1

/STRUCTURE-ID:structure identification

gives the name of the structure as recorded in the disk(s); used when you gave an alias different from the structure identification as argument "dev:", above. See Hints - Using the /STRUCTURE-ID Switch, below. For privileged users only.

/VOLIDS:volid, volid,...

specifies the volids (volume identifiers) of the volumes (reels) of tape you want to access. These must be consecutive volumes, usually of the tape set specified as the "dev:" argument to the MOUNT command. Although you need not specify every volume in the set, any volume not specified will not

be accessible. Do not use this switch if you have given the /OPERATOR switch. See also Characteristics - Using the /VOLIDS Switch, below.

/WRITE-ENABLED — ensures that all volumes in the tape set will be mounted with write rings.
Default when /NEW or /SCRATCH is specified

Characteristics

Action of MOUNT STRUCTURE Command

If the Structure Has Already Been Mounted

If the structure for which you give the MOUNT command is currently mounted, the system simply increases by 1 the mount count (the number of users who have given the MOUNT but not the DISMOUNT command for the structure), and returns your terminal to TOPS-20 command level. A structure is not ordinarily dismounted until its mount count is 0.

If the Structure Has Not Yet Been Mounted

If the structure for which you give the MOUNT command is not currently mounted, your request stays in the mount request queue until it is acted upon by the operator or until you cancel the request.

Setnames (File Set Identifiers)

The setname, or file set identifier of a set of tapes, is part of the label information written into each volume of the set. It is rewritten every time the /NEW or /SCRATCH switch is included in a MOUNT command. The "dev:" argument of the MOUNT command becomes the setname in this case. If you add volumes to an existing tape set, the system uses the setname of the old volumes as the setname of the new ones.

Using the /CHECK-SETNAME Switch

If you give the MOUNT command to use an existing file set (you do not specify the /NEW or /SCRATCH switch), you can give the /CHECK-SETNAME switch to be sure that the setname written on the tapes matches the setname you specify as the "dev:" argument to the MOUNT

command. However, because more than one set of tapes can have the same setname, the /CHECK-SETNAME switch does not ensure that the correct tape set will be mounted. For information about ensuring that the correct tapes are mounted, see Characteristics - Using the /VOLIDS Switch, below.

Volids (Volume Identifiers)

The volid, or volume identifier of a volume (reel) of labeled tape, is part of the label information written into each volume of tape. It is written only once, by the operator during the tape's initialization procedure, and is not changed during the life of the tape. (You should also affix a paper label displaying the volid onto each reel of tape.) You can get a list of volids for previously specified or newly written volumes in any mounted tape set by giving the INFORMATION VOLUMES command for that set.

Using the /VOLIDS Switch

If you give the MOUNT command to use an existing multi-volume tape set (you do not specify the /SCRATCH switch), you can give the volid of each volume you want to use as an argument to the /VOLIDS switch. The system ensures that the correct volumes of a labeled tape will be mounted for your job as long as you use the /VOLIDS switch to specify them. (If the tape set does not consist of labeled tapes, the system does not ensure that the correct tapes are mounted.)

The volids must represent consecutive volumes and must be specified in the order written (oldest first). Note that in general you cannot rely on any apparent alphanumerical order when specifying the volids but must maintain your own list of the volids in each tape set. (See Hints - Keeping Track of Volids, below.) You need not specify every volid in the tape set, but any volume not specified will not be accessible through the current MOUNT command. See also Characteristics - Using the /OPERATOR Switch, and Special Cases - Single-volume Tape Sets, below.

Using the /OPERATOR Switch

You can use the /OPERATOR switch instead of the /VOLIDS switch when asking the system to mount a multi-volume set of tapes. The /OPERATOR switch sends a message to the operator asking him to specify the volid of each volume himself. You must be sure to supply the operator with a list of the volids you want him to

specify before giving a MOUNT command that contains the /OPERATOR switch.

Hints

Checking Whether Operator is Present

You can give the INFORMATION SYSTEM-STATUS command to find out whether the operator is in attendance and can process your mount request. Even if the operator is not in attendance, your request remains valid until he returns and deals with it in some way.

Using the /STRUCTURE-ID Switch

The /STRUCTURE-ID switch (available only to users with enabled WHEEL or OPERATOR capabilities) gives the name of the structure as recorded in the disk(s) of the structure itself, where it is used by the system for identification. Be sure that the structure identification is also written with a felt-tip marker on the upper surface of each disk pack, and on a gummed label on the pack cover.

Unless you give this switch, the system mounts the structure with its structure identification as alias. (The alias is the name you use when specifying the structure in file specifications and commands; the INFORMATION STRUCTURE and INFORMATION AVAILABLE DEVICES commands list structures by alias only.) The /STRUCTURE-ID switch allows an enabled WHEEL or OPERATOR to mount a structure under a name different from the one recorded in the structure.

Use this switch for mounting a structure whose structure identification is the same as the alias of a currently mounted structure. In such cases give the MOUNT STRUCTURE command with any unique alias as the "dev:" argument, and specify the structure identification with the /STRUCTURE-ID switch. In subsequent file specifications and commands referring to the structure, use the alias only.

Dummy "dev:" Arguments for Mounting Tapes

If you want to use different tape sets on successive runnings of a single program, you can refer to those tape sets as a logical name in the program, and use this logical name as the "dev:" argument of your MOUNT command when mounting tapes. As long as you also specify the volid of each volume of tape with the /VOLIDS switch (or use the /OPERATOR switch to ask the operator to do so), you need not give the actual setname of the tape set as the "dev:"

argument to the MOUNT command. The system considers the "dev:" argument you supply to be a logical name defined as the mounted tape set. Therefore, your program can access the tape set using this logical name.

Keeping Track of Volids

Unless your site has a tape cataloging facility, you must keep your own record of the volids in each of your tape sets. After creating a file set on a new tape set, i.e, one not previously owned by you (by giving the MOUNT command and including the /SCRATCH switch), you should give the INFORMATION VOLUMES command for the set before giving the DISMOUNT command. The system will respond by printing a list at your terminal of the volids of all volumes in the tape set. Similarly, if you mount an old tape set and then perform write operations, you should give INFORMATION VOLUMES before giving DISMOUNT to learn the volids of any volumes added to the set. Keep an ordered list of these volids in a disk file in your directory, for use in subsequent MOUNT commands when you give the /VOLIDS switch.

Special Cases

Single-volume Tape Sets

If the tape set you want to mount consists of a single volume of tape, you need not give the /VOLIDS or /OPERATOR switch to specify its volid. You can give the volid as the "dev:" argument to the MOUNT command.

Structures Unavailable for Mounting

If the operator has given the OPR program command, SET STRUCTURE UNAVAILABLE for a specified structure, the system sends an error message including the phrase, "Structure unavailable for mounting" in response to subsequent MOUNT commands for the structure.

Restrictions

Using SET TAPE Commands

The TOPS-20 SET TAPE DENSITY and SET TAPE PARITY commands are applicable to unlabeled tapes only (but see also Warnings - /DENSITY Switch Has Limited Effect for Unlabeled Tapes, below). The SET TAPE FORMAT and SET TAPE RECORD-LENGTH commands are applicable to both labeled and unlabeled tapes, but to labeled tapes only if they are

mounted using the /LABEL-TYPE:ANSI or /LABEL-TYPE:TOPS-20 switch. In addition, the files that you read from or write to such a labeled tape must be in 36-bit format, and they must not have the ;FORMAT attribute as part of their specification.

Warnings

POP Command Cancels Unsatisfied Mount Requests

If you have given a PUSH command to obtain a new level of TOPS-20 and then give a MOUNT command within that new level, a subsequent POP command will cancel your mount request. However, if the specified structure or tape set has already been mounted, it will remain mounted despite your POP command.

/DENSITY Switch Has Limited Effect for Unlabeled Tapes

The /DENSITY switch, when given in a MOUNT command for an unlabeled tape, ensures only that your tape set will be mounted on a drive that supports the specified density. It does not ensure that the tape set will be read or written at this density. To specify the density at which unlabeled tapes are to be read and written, give the SET TAPE DENSITY command.

Effect on Terminal

The MOUNT command with the /NOWAIT switch, leaves your terminal at TOPS-20 command level. If you have not given the /NOWAIT switch, your terminal waits until the system has processed your request, or to return to TOPS-20 command level. This CTRL/C does not cancel your request.

Related Commands

CANCEL	for withdrawing mount requests before they are processed
DISMOUNT	for giving up access to a particular tape drive or disk drive
INFORMATION AVAILABLE DEVICES	for finding out just the names of structures available for mounting (these are listed after DSK and PS, and before

	the line printers (LPT, LPT0, etc.))
INFORMATION MOUNT-REQUESTS	for finding out information about pending mount requests for structures and tape sets, and about currently mounted tape sets
INFORMATION STRUCTURE	for finding out information about currently mounted structures
INFORMATION VOLUMES	for finding out the volids of all mounted volumes (including newly created volumes) of a tape set
SET TAPE commands	for establishing job-wide defaults for tape density, format, parity, and record length

Examples

1. Mount a structure (it is already physically mounted).

```
@MOUNT STRUCTURE SNARK:
Structure SNARK: mounted
```

2. Mount a structure that is not yet physically mounted. After completing the command, give CTRL/Cs to return to TOPS-20 command level.

```
@MOUNT STRUCTURE PYBL:
[Mount Request PYBL Queued, Request-ID 205]
[MOUNT request remaining in queue]
^C
```

3. Mount a structure, then give CTRL/Cs to return to TOPS-20 command level and cancel the mount request.

```
@MOUNT STRUCTURE PYBL:
[Mount Request PYBL Queued, Request-ID 136]
[MOUNT request remaining in queue]
^C
@CANCEL MOUNT 136
[1 mount request canceled]
```

4. Find out what structures are available for mounting (these are listed after DSK and PS and before the line printers), and mount one of these.

```
@INFORMATION AVAILABLE DEVICES
Devices available to this job:
 DSK, PS, LANG, TYM, MISC, SNARK, REL4, LPT, LPT0
 LPT1, CDR, CDP, PCDP0, FE0, FE4-15, PTY7-10
 PTY23-61, NUL, PLT, PLT0, DCN, SRV
Devices assigned to/opened by this job: TTY220
@MOUNT STRUCTURE REL4: /NOWAIT
Structure REL4: mounted
```

5. Ask that a new tape set be created for you from scratch tapes, and copy some files to it. Before dismounting the tape set, find out the volids of the tape volumes you were assigned.

```
@MOUNT TAPE LAT: /SCRATCH/LABEL-TYPE:TOPS-20
[Mount Request LAT Queued, Request-ID 104
[Tape set LAT, volume LAT mounted]
[LAT: defined as MT3:]
@COPY DN20A-11.SYS LAT:
 DN20A-11A.SYS.1 => MT3:DN20A-11A.SYS.131071 [OK]
 DN20A-11B.SYS.1 => MT3:DN20A-11B.SYS.131071 [OK]
 DN20A-11C.SYS.1 => MT3:DN20A-11C.SYS.131071 [OK]
@INFORMATION VOLUMES LAT:
Volumes of tape set LAT: 01P02,00L16
@DISMOUNT TAPE LAT:
[Tape dismounted, logical name LAT: deleted]
```

6. Find out if any tape drives can be used without giving the MOUNT command (any such drives will be of the form MTAn). Assign one of these and use the PLEASE program to ask the operator to mount your (unlabeled) tape on this drive. Set the necessary tape parameters, position the tape, and copy a file from tape to the line printer. Then give up the resources you have been using.

```
@INFORMATION AVAILABLE DEVICES
Devices available to this job:
 DSK, PS, LANG, TYM, MISC, SNARK, REL4, MTA5, LPT, LPT0, LPT1
 CDR, CDP, PCDP0, FE0, FE4-15, PTY23-61, NUL, PLT
 PLT0, DCN, SRV
Devices assigned to/opened by this job: TTY220
@ASSIGN MTA5:
@PLEASE
Enter text, terminate with CTRL/Z to wait for response,
or ESCape to send message and exit
PLEASE MOUNT MY TAPE NAMED UNLBLD IN READ-ONLY MODE ON MTA5:,
WHICH I HAVE ASSIGNED TO MY JOB.  MTA5: IS A 7-TRACK
```

```
DRIVE THAT SUPPORTS TAPE DENSITIES OF 800 BPI, CORRECT?
[PLSOPN Operator at GIDNEY has been notified at 14:34:26]
@SET TAPE DENSITY 800
@SET TAPE RECORD-LENGTH 128
@REWIND MTA5:
@SKIP MTA5: 4 FILES
@COPY MTA5: LPT:
 MTA5: => LPT: [OK]
@UNLOAD MTA5:
@DEASSIGN MTA5:
```

7. Perform the same task using the same volume of tape as in the previous example by using the MOUNT command. Note that you still cannot specify a particular tape file by name when the tape is an unlabeled tape.

```
@MOUNT TAPE UNLBLD: /LABEL-TYPE:UNLABELED/DENSITY:800/DR -
I'VE-TYPE:7-TRACK
[Mount Request UNLBLD Queued, Request-ID 128]
[Tape set UNLBLD, volume UNLBLD mounted]
[UNLBLD: defined as MT3:]
@INFORMATION MOUNT-REQUESTS/USER

Tape/Disk Mount Queue:
Volume    Status  Type   Write   Req Name  Req#  Job#  User
-------  -------  ----  -------  --------  ----  ----  -----
UNLBLD     MTA4  Tape   Locked   UNLBLD    128    55   LATTA
There is 1 Request in the Queue
@REWIND UNLBLD:
@SKIP UNLBLD: 4 FILES
@COPY UNLBLD: LPT:
 MT3:..4 => LPT: [OK]
@DISMOUNT TAPE UNLBLD:
[Tape dismounted, logical name UNLBLD: deleted]
```

8. Mount a labeled tape containing the same files as in the previous two examples, and perform the same task. Note that you need not specify tape parameters in this MOUNT command, as this information is present in the tape labels and is read automatically. Also, the system ensures that the correct volume of tape is used. Finally, you can specify the tape file by name when using labeled tapes.

```
@MOUNT TAPE LBLD: /LABEL-TYPE:ANSI/VOLIDS:00115
[Mount Request LBLD Queued, Request-ID 133]
[Tape set LBLD, volume 00115 mounted]
[LBLD: defined as MT3:]
@INFORMATION MOUNT-REQUESTS/USER
```

```
Tape/Disk Mount Queue:
Volume    Status  Type   Write   Req Name  Req#  Job#  User
-------  -------  ----  -------  --------  ----  ----  -----
00115     MTA0    Tape   Locked  LBLD       133    55  LATTA
There is 1 Request in the Queue
@REWIND LBLD:
@COPY LBLD:COMPR.BRN LPT:
 MT3:COMPR.BRN.13107 => LPT:COMPR [OK]
@DISMOUNT TAPE LBLD:
[Tape dismounted, logical name LBLD: deleted]
```

9. Mount a two-volume tape set (using the NOUNLOAD switch to simplify volume changes) and overwrite any existing files with new files. Then rewind the tape set. Give various INFORMATION commands as you proceed, to verify logical name and device assignments and to determine whether new volumes have been written.

```
@MOUNT TAPE LAT: /WRITE-ENABLED/NOUNLOAD/NOWAIT/VOL-
IDS:DBL01, DBL02
[Mount Request LAT Queued, Request-ID 19]
[Tape set LAT, volume DBL01 mounted]
[LAT: defined as MT0:]
@INFORMATION MOUNT-REQUESTS /USER

Tape/Disk Mount Queue:
Volume    Status   Type   Write   Req Name  Req#  Job#  User
-------  --------  ----  -------  --------  ----  ----  -----
DBL01    MTA1      Tape  Enabled  LAT         19     7  LATTA
There is 1 Request in the Queue
@INFORMATION AVAILABLE DEVICES
Devices available to this job:
 DSK, PS, LANG, SNARK, NTWORK, REL4, MISC, PACKAG, MTA5
 MT0, LPT, LPT0, LPT1, CDP, PCDP0, FE0, FE2-15, PTY13-61
 NUL, PLT, PLT0, DCN, SRV
Devices assigned to/opened by this job: MT0, TTY217
@INFORMATION LOGICAL-NAMES JOB
LAT: => MT0:
SYS: => DSK:,SYS:
@REWIND LAT: /ENTIRE-VOLUME-SET
@COPY HLP:.HLP LAT:
 PS:<HELP>ACCT20.HLP.1 => MT0:ACCT20.HLP.131071 [OK]
 PS:<HELP>ACCTPR.HLP.1 => MT0:ACCTPR.HLP.131071 [OK]
      .
      .
      .
 PS:<HELP>WAIT.HLP.2 => MT0:WAIT.HLP.131071 [OK]
 PS:<HELP>WATCH.HLP.6 => MT0:WATCH.HLP.131071 [OK]
@INFORMATION MOUNT-REQUESTS /USER
```

```
Tape/Disk Mount Queue:
Volume    Status   Type   Write   Req Name  Req#  Job#  User
-------  --------  ----  -------  --------  ----  ----  -----
DBL02    MTA3      Tape  Enabled  LAT         19     7  LATTA
There is 1 Request in the Queue

@INFORMATION VOLUMES LAT:
Volumes of tape set LAT: DBL01,DBL02
@REWIND LAT: /ENTIRE-VOLUME-SET
@INFORMATION MOUNT-REQUESTS /USER

Tape/Disk Mount Queue:
Volume    Status   Type   Write   Req Name  Req#  Job#  User
-------  --------  ----  -------  --------  ----  ----  -----
DBL01    MTA1      Tape  Enabled  LAT         19     7  LATTA
There is 1 Request in the Queue
```

10. (For privileged users only.) Enable your capabilities and call the OPR program, then give the PUSH command to OPR. (This action puts you at TOPS-20 command level, but allows you also to see the OPR messages caused by your TOPS-20 commands.) Repeat the write operations of the previous example, then dismount the tape set and exit from the OPR program. Notice that, because of the /NOUNLOAD switch in your original MOUNT command, both volumes of your tape set remain mounted on their drives. Therefore, you can mount the tapes again without operator assistance.

```
@ENABLE
$OPR
OPR>PUSH

 TOPS-20 Command processor 7(7)
@REWIND LAT: /ENTIRE-VOLUME-SET
@COPY HLP:. LAT:
 PS:<HELP>ACCT20.HLP.1 => MT0:ACCT20.HLP.131071 [OK]
 PS:<HELP>ACCTPR.HLP.1 => MT0:ACCTPR.HLP.131071 [OK]
     .
     .
     .
 PS:<HELP>CHKPNT.HLP.1 => MT0:CHKPNT.HLP.131071
15:11:55          --Tape Drive Released By User--
                MTA1: Volume DBL01 Remaining mounted on drive

15:11:57          --MTA3: Given to Request 19--
                  Volume DBL02 now in use by
                  User LATTA, Job 7, Terminal 217
 [OK]
 PS:<HELP>COBDDT.HLP.6 => MT0:COBDDT.HLP.131071 [OK]
     .
     .
```

```
      .
 PS:<HELP>WAIT.HLP.2 => MT0:WAIT.HLP.131071 [OK]
 PS:<HELP>WATCH.HLP.6 => MT0:WATCH.HLP.131071 [OK]
@INFORMATION VOLUMES LAT:
Volumes of tape set LAT: DBL01,DBL02
@INFORMATION MOUNT-REQUESTS /USER

Tape/Disk Mount Queue:
Volume    Status   Type   Write   Req Name  Req#  Job#  User
-------  --------  ----  -------  --------  ----  ----  -----
DBL02    MTA3      Tape  Enabled  LAT         19     7  LATTA
There is 1 Request in the Queue

@REWIND LAT: /ENTIRE-VOLUME-SET

15:14:51          --Tape Drive Released By User--
                 MTA3: Volume DBL02 Remaining mounted on drive

15:14:51          --MTA1: Given To Request 19--
                  Volume DBL01 now in use by
                  User LATTA, Job 7, Terminal 217

@DISMOUNT TAPE LAT:
[Tape dismounted, logical name LAT: deleted]
15:15:21          --Tape Drive Released By User--
                 MTA1: Volume DBL01 Remaining mounted on drive
@POP
OPR>EXIT
$DISABLE
```

11. Mount the same tape set as in the previous examples, but ask the operator to specify the volids. Use the PLEASE program to help you.

```
@MOUNT TAPE LAT: /WRITE-ENABLED/NOUNLOAD/NOWAIT/OPERATOR
[Mount Request LAT Queued, Request-ID 197]
@PLEASE
Enter text, terminate with CTRL/Z to wait for response,
or ESCape to send message and exit
PLEASE ENTER THE VOLIDS OF MY TAPE SET LAT: FOR
REQUEST 197.  THEY ARE RECORDED IN YOUR TAPE
LIBRARY CATALOG.  THANKS.
[PLSOPN Operator at GIDNEY has been notified at 14:34:26]
@INFORMATION MOUNT-REQUESTS /USER

Tape/Disk Mount Queue:
Volume    Status   Type   Write   Req Name  Req#  Job#  User
-------  --------  ----  -------  --------  ----  ----  -----
DBL01    Waiting   Tape  Enabled  LAT        197    65  LATTA
There is 1 Request in the Queue
```

2.53 PERUSE

Allows you to read a file using read-only editor commands.

Format

@**PERUSE** (FILE) **/switch(es) filespec**

where:

/switch(es) are keywords that apply to the EDIT editor. For descriptions of these switches, see the /CREATE or EDIT commands.

filespec is the name of the file you want to read.
Default - the last file specification and associated switches you gave in a CREATE, EDIT, or PERUSE command during the current terminal session

Characteristics

The PERUSE command runs the EDIT system program in read-only mode. (However, see Special Cases - Using an Editor Other than EDIT, below.) PERUSE is actually the same as the EDIT /READONLY command. In EDIT /READONLY mode you can use only EDIT program switches and commands that do not modify the file.

Hints

Avoid Accidental File Modification

Use PERUSE when it is important to avoid the risk of accidentally modifying a file.

PERUSE Line in SWITCH.INI

Add a line to your SWITCH.INI file for use with the PERUSE command. (See Example 3.)

Special Cases

Using an Editor Other than EDIT

The CREATE, EDIT and PERUSE commands in this manual assume that these commands call on the program EDIT. If your job uses another editing program, for example, EDT, the switches

and examples shown here will not be applicable.

The Editor used by the CREATE, EDIT and PERUSE commands is specified by the logical name EDITOR:. You can find out the name of this program by giving the command, INFORMATION LOGICAL-NAMES EDITOR:. The job-wide definition (if any) will be given first, followed by the system-wide definition; the job-wide definition prevails if both exist. If the definition of EDITOR: is SYS:EDIT.EXE, the CREATE, EDIT and PERUSE commands will function as described in this manual. Otherwise, you must consult the appropriate manual (for example, the EDT-20 Reference Manual) for information.

Restrictions

Listing Available EDIT Read-Only Switches and Commands

Many of the EDIT switches and commands that are displayed in response to a ? modify the file and do not function with PERUSE. If you give a file-modifying switch, the switch is ignored but no error message is displayed. If you give a file-modifying EDIT command, you receive the message %ILLEGAL COMMAND.

For further information on these switches and commands see the EDIT Reference Manual.

Effect on Memory

The PERUSE command clears any unkept forks from memory, then loads your edit program.

Related Commands

CREATE	for creating new files
DIRECTORY-class commands	for getting lists of existing files
EDIT	for modifying files
TYPE	for printing files on your terminal

Examples

1. PERUSE a file

```
@PERUSE FILEX.FOR
Read: FILEX.FOR.1
*P
00100  !THIS IS FILEX.FOR
```

2. PERUSE a file, ten lines at a time, and begin at line 100.

```
@PERUSE /PLINES:10 FIND.BAS
Read: FIND.BAS
*P 100
00100   !       ADD ROUTINE
00200   FOR X = 0 TO 400
00300   IF X$(X) = "XXX" THEN GOTO 2050
00400   NEXT X
00500   PRINT STRING$(10,10);"FILE FULL"\SLEEP 4\GOTO 199
00600   PRINT E$ \LINPUT"ENTER THE NAME ";M$
00700   IF M$ <> "" THEN LET X$(X) = M$ ELSE GOTO 199
00800   PRINT E$ \LINPUT"ENTER THE NUMBER ";O$
00900   IF O$ <> "" THEN LET Y$(X) = O$ ELSE GOTO 199
01000   PRINT E$\INPUT"MORE NAMES TO ENTER ";D$
*E
```

2.54 PLOT

Places requests in a plotter output queue.

Format

@PLOT (FILES) **/switch(es) filespec/switch(es),...**

where:

switches — are keywords, chosen from the list below, indicating your choice of PLOT command options. These switches are of two kinds: job switches and file switches.

Job switches apply to all files specified in the command, no matter where you give the switches.

File switches have different effects depending on their positions in the command line: placed before all files in the command, they act as defaults for all; otherwise they affect only the nearest preceding file.

Defaults are shown in the list of switches

filespec — is the specification of a file you wish to plot. You can use wildcard characters (% and *) to specify more than one file.

Summary of PLOT Command Switches (defaults in boldface)

Job Switches
(affecting the entire command)

/ACCOUNT:account	**Default** account - your current account
/AFTER:date and/or time	
/DESTINATION-NODE:node name	
/FORMS:forms name	**Default** forms name - NORMAL
/GENERIC	
/JOBNAME:6-character name	**Default** name - first six characters of first filename in request
/LIMIT:n	**Default** n - calculated from length of files
/NOTE:12-character message	
YES	

/NOTIFY:NO
/**PRIORITY**:n **Default** n - 10
/SEQUENCE:n
/UNIT:octal number
/USER:user name

File Switches
(affecting only the nearest preceding file,
unless placed before all filespecs)

/**COPIES**:n **Default** n - 1
/DELETE **Default** for files of type .LST
/**HEADER**
ASCII
/MODE:BINARY
IMAGE
/NOHEADER
/**PRESERVE** **Default** for all files except those of type .LST

PLOT Command Switches

Job Switches
(affecting the entire command)

/ACCOUNT:account — specifies the account of 39 or fewer characters to charge for your plotting request. This account must be valid for your user name. **Default** account - your current account. Check with INFORMATION JOB-STATUS.

/AFTER:date and/or time, **or** day of week (or TODAY) and/or time — ensures that the job will not be plotted until after the date and/or time specified. NOV-12-79 and 18:00 illustrate two arguments to this switch. If you give both date and time, separate them with a space. When given alone, the time may be preceded by a plus sign (+), which will delay processing by the indicated length of time from the

present.

Alternatively, you can give a day of the week (such as MONDAY) or TODAY as argument; then the job will not be plotted until the beginning of the following day. If you follow this argument with a plus sign and a time, the job will be further delayed by this amount.

/DESTINATION-NODE:node-name — specifies the remote node on whose plotter your request is to be satisfied. Two colons (::) following the node name are optional.

/FORMS:forms name — specifies, in six or fewer characters, the forms (determining the size of banner, header, and trailer sections; the paper color, width, and weight; the number of plotter steps per inch, location of the origin for plotted data, and so on.) to use for the plotting job. Using this switch may delay processing until the operator can mount the proper forms. Note that your installation may provide a different default argument to this switch.
Default forms name - NORMAL

/GENERIC — allows any plotter to be used for filling the request; use this switch to override a previous /UNIT switch.
Default

/JOBNAME:name — assigns a name (of six or fewer characters) to the plotting job.
Default name - first six characters of first filename in the request

/LIMIT:n — places a limit of n minutes of plotter time on the output of the plotting job.
Default limits, usually adequate, are computed

	from the size of the files you want plotted
/NOTE:message	labels the header section of output (the section displaying the jobname) with a message or notation of up to 12 characters. The message Must be enclosed in double quotation marks if it contains spaces or non-alphanumeric characters.
YES /NOTIFY:NO	tells the system whether to send a message to your terminal when the request has been satisfied. **Default** argument - NO **Default** argument (if switch is given) - YES
/PRIORITY:n	assigns a number n, reflecting the urgency of the plot request. This n must be from 1 to 63, with larger numbers receiving earlier treatment. Note that for non-privileged users the maximum priority that can be specified is lower (usually 20), and that your installation may provide a different value both for this maximum and for the default priority. **Default** n - 10
/SEQUENCE:n	specifies sequence number n for the printing request, which you can use when modifying or canceling the request.
/UNIT:octal number	directs your request to the plotter of the specified octal unit number.
/USER:user name	specifies the user who is to be the owner of the plot request. For privileged users only.

File Switches
(affecting only the nearest preceding file,
unless placed before all file specifications)

/BEGIN:n	starts the plotting at page n of the file. **Default** n - 0
/COPIES:n	requests that n copies of the file be plotted; n must be less than or equal to 62. **Default** n - 1
/DELETE	deletes the file after plotting. **Default** for files of type .LST
/HEADER	causes a header section containing the jobname to be produced before the file itself is plotted. **Default**
/MODE:ASCII /MODE:BINARY /MODE:IMAGE	designates the mode for plotting the file. ASCII treats each word of a disk file as five seven-bit bytes, and truncates each byte to six bits before plotting it. BINARY treats each word as six six-bit bytes, each of which is plotted without modification. IMAGE is the same as BINARY.
/NOHEADER	prevents the production of a header section before the file.
/PRESERVE	saves the file after plotting. **Default** for all files except those of type .LST

Output

Jobname, Request ID, Limit, Number of Input Files

As soon as you complete a valid PLOT command, the system responds by printing, on your terminal, the jobname, request ID number, the limit in minutes of plotter time assigned to the request, and the number of input files in the request.

Characteristics

Ordinary Operation - No Switches

For most purposes you can use the PLOT command with just a series of filespecs for arguments.

Switch Defaults Set by System Manager

The defaults shown in the list of switches are correct for most user sites. However, your system manager can change some of those default settings. The switches most commonly affected are: /FORMS, /HEADER and /NOHEADER, /LIMIT, and /PRIORITY.

Hints

Using SET DEFAULT PLOT

If there are switches that you always or usually supply when using PLOT, give the SET DEFAULT PLOT command to establish them as defaults (at the current TOPS-20 command level) for the remainder of your terminal session. The switches will then behave as if you had typed them directly after the command name. You can supersede any of these default switches by actually supplying the switch, with another value, when you give the PLOT command. Put SET DEFAULT PLOT into a file of specification COMAND.CMD in your log-in directory if you want these default switches to be in effect for all levels of future terminal sessions as well.

Special Cases

/SPOOLED-OUTPUT Switch

You can give the special switch, /SPOOLED-OUTPUT, as sole argument to the PLOT command. This causes any spooled output accumulated so far during your terminal session to be placed in a plotter queue immediately, rather than at log-out time. The/SPOOLED-OUTPUT switch is useful only if the SET SPOOLED-OUTPUT DEFERRED command is in effect. Programs that you run (especially FORTRAN programs) may create spooled output for the plotter, or you can create it directly by giving the command, COPY filespec PLT:.

Related Commands

CANCEL for withdrawing PLOT requests

INFORMATION OUTPUT-REQUESTS	for examining requests in the output queues
MODIFY	for changing PLOT requests before processing has begun
SET DEFAULT PLOT	for establishing default switches for subsequent PLOT commands.

Examples

1. Plot a file.

```
@PLOT CNTR.MED
[Plotter job CNTR queued, request-ID 91, limit 2]
```

2. Send all files having a four-character file type ending in "CTH" to the plotter. Assign a jobname to the request, and ensure they are not plotted until tomorrow. Check for the request in the output queues, and then cancel it.

```
@PLOT *.CTH /JOBNAME:HATCH/AFTER:TODAY
[Job HATCH Queued, Request-ID 94, Limit 3, 3 Files]
@INFORMATION OUTPUT-REQUESTS

Plotter Queue:
Job Name  Req#  Limit  User
--------  ----  -----  -----------
  CNTR      91      2  LAUDERDALE
  HATCH     94      3  ASHLEY /After:21-Jul-79 00:00
There are 2 jobs in the queue (none in progress)

@CANCEL PLOT 94
[1 Job canceled]
```

2.55 POP

Terminates the current level of TOPS-20 and returns you to its superior process.

Format

@**POP** (COMMAND LEVEL)

Characteristics

POP the Opposite of PUSH

You can do one and only one POP command for every previous PUSH command. Giving too many POP commands will cause an error message to be printed on your terminal.

Job Parameters Affected by POP

As soon as you complete a valid POP command at some level of TOPS-20, you give up the copy of memory for that level of TOPS-20 and any program you were running. Any defaults established at that level (such as default filespecs for LOAD-class and EDIT-class commands, defaults specified by SET DEFAULT commands) are cancelled as well. If POP returns you to a higher level of TOPS-20, all these parameters revert to any values established at that higher level.

Special Cases

Returning to Other Programs With POP

The POP command usually returns you to the level of TOPS-20 from which you gave a previous PUSH command. But a few system programs such as PTYCON and OPR, also allow you to give PUSH to get a new level of TOPS-20. Giving the POP command to this level of TOPS-20 returns you to that program.

Effect on Memory and Terminal

The POP command clears memory, terminates the current level of TOPS-20, and returns your terminal to the previous TOPS-20 command level (but see Special Cases, above). Memory for the previous TOPS-20 command level is not affected by this action.

Related Commands

CONTINUE	for resuming execution of a program in memory
INFORMATION SUPERIORS	for information on the number of forks that are superior to the current EXEC level
PUSH	for obtaining a new level of TOPS-20

Examples

1. Give the POP command to return to a higher level of the TOPS-20 command processor (EXEC).

   ```
   @POP
   ```

2. Run a program and halt it with CTRL/Cs. Give a CONTINUE STAY command to resume its execution, and then the PUSH command for a new level of TOPS-20. Run another program at this lower level, then use the POP command to return to the first level; in this case you return before receiving the final message of the first program.

   ```
   @RUN DMN

   ^C
   @CONTINUE /STAY
   @PUSH

    TOPS-20 Command processor 7(7)
   @RUN TESTF1

   THIS IS A TEST.

   CPU time: 0.03  Elapsed time: 0:72
   EXIT
   @POP
   EXIT
   ```

2.56 PRINT

Places requests in a line printer output queue.

Format

@PRINT (FILES) **/switch(es) filespec/switch(es),...**

where:

switches — are keywords, chosen from the list below, indicating your choice of PRINT command options. These switches are of two kinds: job switches and file switches.

Job switches apply to all files specified in the command, no matter where you give the switches.

File switches have different effects depending on their positions in the command file: placed before all files in the command, they act as defaults for all; otherwise they affect only the nearest preceding file.

Defaults are shown in the list of switches

filespec — is the specification of a file you wish to print. You can use wildcard characters (% and *) to specify more than one file.

Summary of PRINT Command Switches (defaults in boldface)

Job Switches
(affecting the entire command)

/ACCOUNT:account	**Default** account - your current account
/AFTER:date and/or time	
/CHARACTERISTIC:characteristic value	
/DESTINATION-NODE:node name	
/FORMS:forms name	**Default** forms name - NORMAL
/GENERIC	
/JOBNAME:6-character name	**Default** name - first six characters of first filename in request
/LIMIT:n	**Default** n - calculated from length of files

/LOWERCASE
/NOTE:12-character message
/NOTIFY:NO
/**PRIORITY**:n — **Default** n - 10
/REMOTE-PRINTER:type
/SEQUENCE:n
/UNIT:octal number
/UPPERCASE
/USER:user name

File Switches
(affecting only the nearest preceding file,
unless placed before all filespecs)

/**BEGIN**:n — **Default** n - 0
/**COPIES**:n — **Default** n - 1
/DELETE — **Default** for files of type .LST

ASCII
COBOL
/FILE:ELEVEN
FORTRAN — **Default** for files of type .DAT only

/**HEADER**

ARROW
ASCII
/MODE:OCTAL
SUPPRESS

/NOHEADER
/**PRESERVE** — **Default** for all files except those of type .LST

/REPORT:12-character title

SINGLE
/**SPACING**:DOUBLE
TRIPLE

PRINT Command Switches

Job Switches
(affecting the entire command)

/ACCOUNT:account
: specifies the account of 39 or fewer characters to charge for your printing request. This account must be valid for your user name.
  Default account – your current account (check with INFORMATION JOB-STATUS)

/AFTER:date and/or time, **or**

day of week (or TODAY)
and/or time
: ensures that the job will not be printed until after the date and/or time specified. NOV-12-79 and 18:00 illustrate two arguments to this switch. If you give both date and time, separate them with a space. When given alone, the time may be preceded by a plus sign (+), which will delay processing by the indicated length of time from the present.

  Alternatively, you can give a day of the week (such as MONDAY) or TODAY as argument; then the job will not be printed until the beginning of the following day. If you follow this argument with a plus sign and a time, the job will be further delayed by this amount.

/CHARACTERISTIC:characteristic value
: specifies an alphanumeric string that communicates print features, such as layout or lettering type, to the remote system for DQS remote print requests. The user can specify one or more characteristic keywords; multiple values are separated by commas. You can also specify numeric bit values, which must be separated by commas and enclosed in parentheses. The keywords to the /CHARACTERISTIC switch are defined by the SET REMOTE-PRINTING CHARACTERISTICS command. The maximum length of each characteristic keyword is 14 characters, and the name must

begin with an alphabetic character.

You must use the /CHARACTERISTIC switch in conjunction with the /REMOTE-PRINTER switch not with the /UNIT switch, which applies only to local and cluster printers.

/DESTINATION-NODE:node-name

specifies the remote node on whose line printer your request is to be printed. The remote node can be either an IBM remote station, a node in a TOPS-20 cluster, a DQS server node, or LATserver. Two colons (::) following the node name are optional. You can use this switch either to send output to a remote node or to redirect it from a remote node.

/FORMS:forms name

specifies, in six or fewer characters, the forms (determining the number of banner, header, and trailer pages; the paper color, width, and weight; vertical format, carriage control tape, and so on) to use for the printing job. Using this switch may delay processing until the operator can mount the proper forms. Note that your installation may provide a different default argument to this switch.

Default forms name - NORMAL

/GENERIC

allows any printer, either upper or lowercase, and of any unit number, to be used for satisfying the request. Use this switch to override a previous /UPPERCASE, /LOWERCASE, or /UNIT switch.

Default

/JOBNAME:name

assigns a name (of six or fewer characters) to the printing job.

Default name - first six characters of first filename in the request

/LIMIT:n

places a limit of n pages on the output of the printing job.

Default limits, usually adequate, are computed from the size of the files you want printed

/LOWERCASE — directs the job to a line printer that can print both uppercase and lowercase characters.

/NOTE:message — labels the header page of output (the page displaying the jobname) with a message or notation of up to 12 characters. The message must be enclosed in double quotation marks if it contains spaces or non-alphanumeric characters.

YES
/NOTIFY:NO — tells the system whether to send a message to your terminal when the request has been satisfied.

Default argument - NO

Default argument (if switch is given) - YES

/PRIORITY:n — assigns a number n, reflecting the urgency of the print request. This n must be from 1 to 63, with larger numbers receiving earlier treatment. Note that for non-privileged users the maximum priority that can be specified is lower (usually 20), and that your installation may provide a different value both for this maximum and for the default priority.

The system acknowledges this switch by displaying the message [Priority has been modified].

Default n - 10

/REMOTE-PRINTER:type — directs a print request to a remote destination. The destination is either a remote printer queue for DQS printers or a LATserver PORT or SERVICE for LAT printers. The type refers to an actual remote queue name or an alias of a remote printer queue name for DQS printers, or to an application terminal on a reachable

LATserver. Aliases are set up with the SET REMOTE-PRINTER command.

You can use the /REMOTE-PRINTER switch with the /DESTINATION-NODE switch, but not with the /UNIT switch.

/SEQUENCE:n — specifies sequence number n for the printing request, which you can use when modifying or canceling the request.

/UNIT:octal number — directs your request to the line printer with the specified octal unit number. This switch only applies to local or TOPS-20 cluster print requests.

/UPPERCASE — directs the job to a line printer that uses only uppercase characters.

/USER:user name — specifies the user who is to be the owner of the print request. For privileged users only.

File Switches
(affecting only the nearest preceding file,
unless placed before all file specifications)

/BEGIN:n — starts the printing at page n of the file.
Default n - 0

/COPIES:n — requests that n copies of the file be printed; n must be less than or equal to 62.
Default n - 1

/DELETE — deletes the file after printing.
Default for files of type .LST

/FILE:ASCII
/FILE:COBOL
/FILE:ELEVEN
/FILE:FORTRAN — specifies that the file consists of ASCII text, or COBOL SIXBIT text; or (ELEVEN) contains four eight-bit bytes in each 36-bit word - for emulating paper tape punch only; or is FORTRAN ASCII text, where column 1 of each line is interpreted as a

carriage control character.
Default - ASCII (except for files of type .DAT, for which the default is FORTRAN)

/HEADER — causes header pages containing the jobname to be printed before the file itself.
Default

/MODE:ARROW
/MODE:ASCII
/MODE:OCTAL
/MODE:SUPPRESS — designates the mode for printing the file. ARROW prints the file literally; but denotes each control character by an up-arrow (^) and the character, except for the following, which are reproduced literally (that is, the control characters perform their normal actions): carriage return, line feed, horizontal tab, vertical tab, form feed, ^P, ^Q, ^R, ^S, and ^T. ASCII prints the file literally, without omissions or substitutions, except for the escape character (^[) which is represented as a dollar sign.

OCTAL prints each word in the file as unsigned octal integers; 3 groups of 128 words (8 rows of 16 columns each) appear on a standard line printer page. SUPPRESS prints the file without any blank lines, causing all vertical format characters (CTRL/K, CTRL/L, CTRL/Q, CTRL/R, CTRL/S, and CTRL/T) to be converted to CRLFs (carriage return/linefeeds), and then interpreting multiple occurrences of CRLFs as a single CRLF.
Default - ARROW

/NOHEADER — prevents the printing of header pages before the file.

/PRESERVE — saves the file after printing.
Default for all files except those of type .LST

/REPORT:title — scans your files and prints only those lines whose first characters are the title you give. This title

may contain up to 12 characters (including the quotation marks that must enclose the title if it contains spaces). The switch is used along with the COBOL report writer.

SINGLE
/SPACING:DOUBLE
TRIPLE

determines the spacing between lines in the printout.
Default - SINGLE

Output

Jobname, Request ID, Limit, Number of Files

As soon as you complete a valid PRINT command, the system responds by printing, on your terminal, the jobname, the node name (if printed on remote node), request ID number, the limit in pages of output assigned to the request, and the number of files in the request.

Characteristics

Ordinary Operation - No Switches

For most purposes you can use the PRINT command with just a series of filespecs for arguments.

Switch Defaults Set by System Manager

The defaults shown in the list of switches are correct for most user sites. However, your system manager can change some of those default settings. The switches most commonly affected are: /FORMS, /HEADER and /NOHEADER, /LIMIT, and /PRIORITY.

Restrictions

Using /UNIT, /REMOTE-PRINTER, and /CHARACTERISTICS

You must use the /CHARACTERISTICS switch in conjunction with the /REMOTE-PRINTER switch and not with the /UNIT switch, which applies only to local and cluster printers. If you use the /UNIT switch in conjunction with the /REMOTE-PRINTER switch, it is ignored. The /CHARACTERISTICS switch is ignored if used in a local or cluster print job.

NOTE

The /CHARACTERISTICS switch does not apply to LATprinters.

Hints

Using SET DEFAULT PRINT

If there are switches that you always or usually supply when using PRINT, give the SET DEFAULT PRINT command to establish them as defaults (at the current TOPS-20 command level) for the remainder of your terminal session. The switches will then behave as if you had typed them directly after the command name. You can supersede any of these default switches by actually supplying the switch, with another value, when you give the PRINT command. Put SET DEFAULT PRINT into a file of specification COMAND.CMD in your log-in directory if you want these default switches to be in effect for all levels of future terminal sessions as well.

Special Cases

/SPOOLED-OUTPUT Switch

You can give the special switch, /SPOOLED-OUTPUT, as sole argument to the PRINT command. This causes any spooled output accumulated so far during your terminal session to be placed in a line printer queue immediately, rather than at log-out time. The /SPOOLED-OUTPUT switch is useful only if the SET SPOOLED-OUTPUT DEFERRED command is in effect. Programs that you run (especially FORTRAN programs) may create spooled output for the printer, or you can create it directly by writing to device LPT: (by giving the command, COPY filespec LPT:), or giving a CREF command.

Related Commands

CANCEL	for withdrawing PRINT requests
INFORMATION OUTPUT-REQUESTS	for examining requests in the output queues
MODIFY	for changing PRINT requests before processing has begun
SET DEFAULT PRINT	for establishing default

	switches for subsequent PRINT commands
SET REMOTE-PRINTING CHARACTERISTICS	for setting up a string to be used as input to the /CHARACTERISTICS switch
SET REMOTE-PRINTING PRINTER	for setting up a string to be used as input to the /REMOTE-PRINTER switch
SET REMOTE-PRINTING SYSTEM-DEFINITIONS	for setting up remote printing information for a job based on input in SYSTEM:REMOTE-PRINTING.CMD

Examples

1. Print two of your files.

```
@PRINT 4-UPED.TXT, CMPTN.TXT
[Printer job 4-UPED queued, request  302, limit 200, 2 files]
```

2. Print three files, assigning a jobname and a note for the header page; postpone the printing. Make 4 copies of one of the files, and double-space another one.

```
@PRINT /JOBNAME:COMFIL/NOTE:CONFIDENTIAL/AFTER:12-DEC-85 -
FOO.CTL, HOLMAX.CTL/COPIES:4, INSIDE.RNO/SPACING:DOUBLE
[Printer job COMFIL queued, request #306, limit 27, 3 files]
@INFORMATION OUTPUT-REQUESTS /USER/ALL

Printer Queue:
Job Name  Req#  Limit             User
--------  ----  -----  ------------------------------
  COMFIL   306     27  LATTA  /After:12-Dec-85  0:00
          /Note:CONFIDENTIAL   /Seq:1865
There is 1 job in the queue (none in progress)
```

3. Print a job in a hurry, by assigning a high priority and skipping the header and first five pages. Print 10 copies of the first file, and 18 of the second.

```
@PRINT /JOBNAME:RUSH /PRIORITY:60/NOHEADER/BEGIN:5/COPIES:1 -
0 RFM.CTL, HOLMAX.CTL/COPIES:18
[Printer job RUSH queued, request #312, limit 27, 2 files]
[Priority has been modified]
@INFORMATION OUTPUT-REQUESTS /USER/ALL

Printer Queue:
```

```
Job Name  Req#  Limit             User
--------  ----  -----  ------------------------------
* RUSH    312      27  LATTA    On Unit:0  /Prio:20
         /Seq:1870
   Started at 15:00:11, printed 10 of 27 pages
 COMFIL   306      27  LATTA   /After:12-Dec-79  0:00
         /Note:CONFIDENTIAL  /Seq:1865
There are 2 jobs in the queue (1 in progress)
```

4. Print a job with a P90 characteristic on a XEROX 8700 printer on a VMS system.

```
@PRINT FILE4.MEM/REMOTE-PRINTER:XEROX/CHARACTERISTIC:P90
[Printer job FILE4 queued, request #33, limit 1 files
```

5. Print a job on LATserver printer in one of four ways. The TOPS-20 host has access to a printer service named LASER, which is attached to port LC14 on a LATserver named LAT97.

   o SERVICE name only

```
@PRINT FILE.DAT/REMOTE-PRINTER:LASER
[Printer job FILE queued, request #43, limit 1 files]
```

   o SERVICE name and SERVER name

```
@PRINT FILE.DAT/REMOTE-PRINTER:LASER/DESTINATION-NODE:LAT97
[Printer job FILE queued, request #45, limit 1 files]
```

   o PORT name and SERVER name

```
@PRINT FILE.DAT/REMOTE-PRINTER:LC14/DESTINATION-NODE:LAT97
[Printer job FILE queued, request #53, limit 1 files]
```

   o SERVER name

```
@PRINT FILE.DAT/DESTINATION-NODE:LAT97
[Printer job FILE queued, request #58, limit 1 files]
```

2.57 PUNCH

Places requests in a card punch or paper tape punch output queue.

Format

@PUNCH (ONTO) **medium** (FILES) **/switch(es) filespec/switch(es),...**

where:

medium — is the name of the medium on which you want to punch your file(s). It can be either

CARDS
or
PAPER-TAPE

switches — are keywords, chosen from the list below, indicating your choice of PUNCH command options. These switches are of two kinds: job switches and file switches.

Job switches apply to all files specified in the command, no matter where you give the switches.

File switches have different effects depending on their positions in the command line: placed before all files in the command, they act as defaults for all; otherwise they affect only the nearest preceding file.

Defaults are shown in the list of switches

filespec — is the specification of a file you wish to punch. You can use wildcard characters (% and *) to specify more than one file.

Summary of PUNCH Command Switches (defaults in boldface)

Job Switches
(affecting the entire command)

/ACCOUNT:account	**Default** account - your current account
/AFTER:date and/or time	
/DESTINATION-NODE:node-name	
/FORMS:forms name	**Default** forms name - NORMAL

/GENERIC	
/JOBNAME:6-character name	**Default** - first six characters of first filename in request
/LIMIT:n	**Default** n - calculated from length of files
/METERS:n (PAPER-TAPE only)	**Default** n - calculated from length of files
/NOTE:12-character message	
YES	
/NOTIFY:NO	
/PRIORITY:n	**Default** n - 10
/SEQUENCE:n	
/UNIT:octal number	
/USER:user name	

File Switches
(affecting only the nearest preceding file,
unless placed before all filespecs)

/COPIES:n	**Default** n - 1
/DELETE	**Default** for files of type .LST
/HEADER	
ASCII	
BCD	
/MODE:BINARY (CARDS only)	
IMAGE	
ASCII	
BINARY	
/MODE:IMAGE (PAPER-TAPE only)	
IMAGE-BINARY	
/NOHEADER	
/PRESERVE	**Default** for all files except those of type .LST

PUNCH Command Switches

Job Switches
(affecting the entire job)

/ACCOUNT:account — specifies the account of 39 or fewer characters to charge for your punching request. This account must be valid for your user name.
Default account - your current account (check

with INFORMATION
JOB-STATUS)

/AFTER:date and/or time **or**

day of week (or TODAY)
and/or time — ensures that the job will not be punched until after the date and/or time specified. NOV-12-79 and 18:00 illustrate two arguments to this switch. If you give both date and time, separate them with a space. When given alone, the time may be preceded by a plus sign (+), which will delay processing by the indicated length of time from the present.

Alternatively, you can give a day of the week (such as MONDAY) or TODAY as argument; then the job will not be punched until the beginning of the following day. If you follow this argument with a plus sign and a time, the job will be further delayed by this amount.

/DESTINATION-NODE:node-name
specifies the IBM remote node on whose card punch or paper tape punch your request is to be satisfied. Two colons (::) following the node name are optional.

/FORMS:forms name
specifies, in six or fewer characters, the forms (determining the weight and color of card or paper stock, the size of banner, header, and trailer sections, etc.) to use for the punching job. Using this switch may delay processing until the operator can mount the proper forms. Note that your installation may provide a different default argument to this switch.
Default forms name - NORMAL

/GENERIC
allows any card punch or paper tape punch to be used for satisfying the request; use this switch to override a previous /UNIT switch.
Default

/JOBNAME:name

assigns a name (of six or fewer characters) to the punching job.

Default name - first six characters of first filename in the request

/LIMIT:n

places a limit of n cards (or n feet of paper tape) on the output of the punching job.

Default limits, usually adequate, are calculated from the size of the files you want punched

/METERS:n

places a limit of n meters on the output of the punching job (PAPER-TAPE only).

/NOTE:message

labels the header section of output (the section displaying the jobname) with a message or notation of up to 12 characters. The message must be enclosed in double quotation marks if it contains spaces or non-alphanumeric characters.

YES
/NOTIFY:NO

tells the system whether to send a message to your terminal when the request has been satisfied.

Default argument - NO

Default argument (if switch is given) - YES

/PRIORITY:n

assigns a number n, reflecting the urgency of the punch request. This n must from 1 to 63, with larger numbers receiving earlier treatment. Note that for non-privileged users the maximum priority that can be specified is lower (usually 20), and that your installation may provide a different value both for this maximum and for the default priority.

Default n - 10

/SEQUENCE:n

specifies sequence number n for the punch request, which you can use when modifying or canceling the request.

/UNIT:octal number

directs your request to the card punch

or paper tape punch of the specified octal unit number.

/USER:user name — specifies the user who is to be the owner of the punch request. For privileged users only.

File Switches
(affecting only the nearest preceding file, unless placed before all file specifications)

/COPIES:n — requests that n copies of the file be punched; n must be less than or equal to 62.
Default n - 1

/DELETE — deletes the file after punching. Opposite of /PRESERVE.
Default for files of type .LST

/HEADER — causes a header section containing the jobname to be punched before the file itself is produced.
Default

/MODE:ASCII
/MODE:BCD
/MODE:BINARY (CARDS only)
/MODE:IMAGE — designates the mode for punching the file onto cards. One of the following:

- ASCII treats each word of a disk file as five seven-bit bytes and punches each byte into one column of the card, using the ASCII translation table for conversion into Hollerith code.
- BCD is the same as ASCII, except that it uses the 026 translation table.
- BINARY treats each group of 26 words as 78 12-bit bytes and punches each byte into one column of the card, from column 3 through column 80; column 1 contains the octal word count in rows 12 through 3 and rows 7 and 9 punched, while column 2

contains a 12-bit folded checksum.

- o IMAGE treats each group of 27 words as 81 12-bit bytes and punches each byte into one column of the card, ignoring the eighty-first byte.

ASCII
BINARY
/MODE:IMAGE (PAPER-TAPE only)
IMAGE-BINARY designates the mode for punching the file onto paper tape. One of the following:

- o ASCII treats each word of a disk file as five seven-bit bytes plus an even parity bit for each byte, and punches each byte into one frame of paper tape; if a vertical or horizontal tab is punched, it is followed by a rubout character, and if a formfeed is punched, it is followed by 16 null characters.

- o BINARY treats each group of 33 words as 1 control word followed by 32 words of data, where each word (both control and data) consists of six 6-bit bytes, and punches each byte into one frame of paper tape after adding 200 (octal) to the byte; the control word consists of a folded checksum in the left half and the data word count in the right half.

- o IMAGE treats each word of a disk file as one 8-bit byte followed by 28 zeroes, and punches each byte into one frame of paper tape.

- o IMAGE BINARY treats each word as six 6-bit bytes, and punches each byte into one frame of paper tape after adding 200 (octal) to each byte.

/NOHEADER prevents the punching of a header section before the file

/PRESERVE saves the file after punching. Opposite of /DELETE.

Default for all files

Output

Jobname, Request ID, Limit, Number of Input Files

As soon as you complete a valid PUNCH command, the system responds by printing, on your terminal, the jobname, request ID number, the output limit in number of cards or feet of paper tape assigned to the request, and the number of input files in the request.

Characteristics

Ordinary Operation - No Switches

For most purposes you can use the PUNCH command with just the medium and a series of filespecs for arguments.

Switch Defaults Set by System Managers

The defaults shown in the list of switches are correct for most user sites. However, your system manager can change some of those default settings. The switches most commonly affected are: /FORMS, /HEADER and /NOHEADER, /LIMIT, and /PRIORITY.

Hints

Using the SET DEFAULT Commands

If there are switches that you always or usually supply when using PUNCH, give the SET DEFAULT CARDS or SET DEFAULT PAPER-TAPE command to establish them as defaults (at the current TOPS-20 command level) for the remainder of your terminal session. The switches will then behave as if you had typed them directly after the command name. You can supersede any of these default switches by actually supplying the switch, with another value, when you give the PUNCH command. Put SET DEFAULT commands into a file of specification COMAND.CMD in your log-in directory if you want these default switches to be in effect for all levels of future terminal sessions as well.

Special Cases

/SPOOLED-OUTPUT Switch

You can give the special switch, /SPOOLED-OUTPUT, as sole argument to the PUNCH CARDS or PUNCH PAPER-TAPE command. This causes any spooled output accumulated so far during your terminal session to be placed in a card punch or paper tape punch queue immediately, rather than at log-out time. The /SPOOLED-OUTPUT switch is useful only if the SET SPOOLED-OUTPUT DEFERRED command is in effect. Programs that you run (especially FORTRAN programs) may create spooled paper tape punch or card punch output. Or you can create it directly by giving the command, COPY filespec PTP:, or COPY filespec CDP:, respectively, or by giving a CREF command.

Related Commands

CANCEL	CARDS PAPER-TAPE	for withdrawing PUNCH requests
INFORMATION	OUTPUT-REQUESTS	for examining requests in the output queues
MODIFY	CARDS PAPER-TAPE	for changing PUNCH requests before processing has begun
SET DEFAULT	CARDS PAPER-TAPE	for establishing default switches for subsequent PUNCH commands

Examples

1. Punch a file onto cards.

   ```
   @PUNCH CARDS ESTMT.DAT
   [Card-Punch job ESTMT queued, request-ID 146, limit 30]
   ```

2. Punch a file onto paper tape.

   ```
   @PUNCH PAPER-TAPE REAUMUR.LNS
   [Papertape job REAUMU queued, request-ID 12, limit 55]
   ```

3. Punch three files onto paper tape, specifying a particular paper tape punch for two of them and allowing the third to be punched on any available device.

   ```
   @PUNCH PAPER-TAPE /UNIT:2 INDX.LTG, PON.LG4/GENERIC, -
   BENNETT.TXT
   [Papertape job INDX queued, request-ID 149, limit 110, 3 files]
   ```

4. Punch a file onto paper tape, specifying that the job not begin for an hour. Check for your requests in the output queues, then cancel both of your paper tape requests.

```
@PUNCH PAPER-TAPE FORUM.APR /AFTER:+1:00
[Papertape job FORUM queued, request-ID 150, limit 10]
@INFORMATION OUTPUT-REQUESTS/USER

Papertape Queue:
Job Name  Req#  Limit             User
--------  ----  -----  --------------------------------
    INDX   149    110  SCARNY
   FORUM   150     10  SCARNY /After:20-Jul-79 16:20
There are 2 jobs in the queue (none in progress)

Card-Punch Queue:
Job Name  Req#  Limit             User
--------  ----  -----  --------------------------------
  ESTMT    146     30  SCARNY
There is 1 job in the queue (none in progress)

@CANCEL PAPER-TAPE *
[2 Jobs canceled]
```

2.58 PUSH

Creates a new level of TOPS-20 inferior to the one from which you give the PUSH command.

Format

@**PUSH** (COMMAND LEVEL)

Characteristics

A New Level of TOPS-20

The PUSH command creates an inferior level of the TOPS-20 command processor (EXEC). The system's SYSTEM:COMAND.CMD file and your login directory's COMAND.CMD file are executed again, you have a fresh copy of memory and can begin giving commands as if you had just logged in. However, job-wide parameters (for example, connected and accessed directories, logical name definitions, most parameters altered by SET commands) are unaffected by the PUSH command and retain their values.

Hints

Creating a Different Copy of an EXEC

You can use the PUSH command to create an inferior level of an EXEC of your choice. Normally, PUSH creates the EXEC defined by the system logical name, DEFAULT-EXEC:. Use the DEFINE command to define a job logical name, DEFAULT-EXEC:, with the name of the EXEC you want to create each time you PUSH.

Note that many TOPS-20 programs have their own PUSH commands. However, only the EXEC and OPR PUSH commands refer to the job's definition of DEFAULT-EXEC:.

Using CONTINUE STAY With PUSH

You can use the PUSH command to run two programs at once or to do other work that requires more than one copy of memory. Simply use the CONTINUE /STAY or CONTINUE /BACKGROUND command to continue execution of your current program before using PUSH. After PUSH you can run another program or otherwise alter memory without affecting memory for the first program. See Example 2. But see also Warning, below.

Use of Multiforking Instead of PUSH

The PUSH command allows you to run several programs at once by running the programs at different EXEC levels. The EXEC's multiforking feature allows you to run multiple programs at the same EXEC level. Working from a single EXEC makes multiprogramming easier to monitor and manage. For information on multiforking see the KEEP command or the TOPS-20 User's Guide.

Restrictions

Number of Successive PUSH commands

You can give as many pairs of PUSH and POP commands as necessary to complete your task. Although there is a limit to the number of times you can give PUSH without giving intervening POP commands, this limit is large enough (approximately 24, although smaller for a heavily loaded system) not to interfere with most applications. There is a smaller limit (usually 5) on the number of EXECs that can give Queue-class commands.

Use the INFORMATION SUPERIORS command to learn how many superior EXEC levels you have created.

Invalid Definition of DEFAULT-EXEC:

If you define DEFAULT-EXEC: with the name of a nonexistent EXEC, (if, for example, you make a spelling error in your DEFINE command), the PUSH command ignores the job's definition of DEFAULT-EXEC: and creates the EXEC defined by the system definition.

Withheld Log-out Capability

You can usually log out from a lower level of TOPS-20 than the one to which you logged in. By doing so, you simultaneously conclude all processes of your job. However, if a program (such as, PTYCON) has initialized a level of the TOPS-20 command processor but has withheld log-out capability from it, you must use the POP command, followed, if necessary, by a program command to exit from the program and return to a higher level of TOPS-20, before you can log out.

Warning

Competition Between Processes

If you have two programs running at once after using CONTINUE /STAY or CONTINUE /BACKGROUND and PUSH commands (see Hints, above) they may try to access the same files at the same time. Or, TOPS-20 commands given at the lower level may be intercepted by a program running at the higher level. For a discussion of these possibilities, see the Restrictions section of the CONTINUE command description.

Effect on Memory and Terminal

The PUSH command preserves your present memory, gives you a fresh copy of memory, and leaves your terminal at a new TOPS-20 command level.

Related Commands

CONTINUE /STAY	for beginning execution of a program before giving the PUSH command
INFORMATION SUPERIORS	for displaying the number of superior EXEC levels.
POP	for returning to a previous level of TOPS-20

Examples

1. Give the PUSH command.

```
@PUSH

 TOPS-20 Command processor 7(28)
```

2. Run a program, and give a CTRL/C to return to TOPS-20 command level. Give a CONTINUE /STAY command to resume this program's execution, and then a PUSH command for a new copy of the TOPS-20 command language. Repeat this process twice; now you have three programs running at once. In the lowest (fourth) level of your job, begin editing a file. (Note: when running more than one program in this way, be sure that they do not use the same compiler or the same data base; otherwise, competition among them could cause unpredictable situations to develop.)

```
@RUN TESTF1
^C
@CONTINUE /STAY
@PUSH
```

```
 TOPS-20 Command processor 7(28)
@RUN DMN
^C
@CONTINUE /STAY
@PUSH
 TOPS-20 Command processor 7(28)
@RUN PRODUK
^C
@CONTINUE /STAY
@PUSH

 TOPS-20 Command processor 7(28)
@EDIT ARTIFI.CTL
```

3. Define logical name DEFAULT-EXEC: with the name of a specialized EXEC. Then, run that EXEC with the PUSH command.

```
@DEFINE DEFAULT-EXEC: SYSTEM:EXTENDED-EXEC.EXE.7
@PUSH

 TOPS-20 Command processor 7(6530)
```

2.59 R

Places an executable system program in memory and starts it.

Format

@R (PROGRAM) **filespec** /switch

where:

filespec — is the file specification of any executable program.
Default dev:<directory> - SYS:

Default .typ - .EXE

/switch — is /USE-SECTION:n
specifies the memory section (from 0 to 37 octal) in which your program is to run. You can use this switch only if your program can be contained in one section.

Characteristics

Need for R Command

Although in most cases you can run system programs by simply typing the program name in place of an EXEC command, the R command is necessary for running a program whose name is the same as an EXEC command or an abbreviation for an EXEC command. For example, if your site has a system program named CONNECT, it must be run with the R command in order to distinguish it from the EXEC's CONNECT command. If you have a system program named GE, it must be run with the R command to distinguish it from the GE abbreviation for the GET command.

Cancels the Ephemeral Attribute

If a system program has been set ephemeral by the system manager or, you have given a SET PROGRAM EPHEMERAL command, you can cancel the ephemeral attribute by running the program with the R command instead of typing the program name as an EXEC command.

For more information on the ephemeral attribute, see the ERUN command.

Hints

Defining SYS:

If you redefine logical name SYS: to be different from the system-wide definition, you should include SYS: in the search list if you want to use the R command to run system programs. For further information, see the section entitled, Redefining System Logical Names, in the DEFINE command description.

Effect on Memory

The R command clears any unkept forks, places in memory and starts the specified program.

Related Commands

INFORMATION LOGICAL-NAMES	for examining the definition of SYS:
RUN	for running executable user programs

Examples

1. Run the FILCOM system program.

```
@R FILCOM
*
```

2. Find out what APL programs are available in logical name SYS:. Run one of them.

```
@DIRECTORY SYS:*APL*.EXE

   PS:<FIELD-IMAGE>
 APL.EXE.1
 APLSF.EXE.1
 MAPLFL.EXE.1

 Total of 3 files
@R APL.EXE
```

2.60 RECEIVE

Notifies the system that you are willing to accept communication links, advice, user messages, and system messages.

Format

@RECEIVE argument

where:

argument — is a keyword, chosen from the list below, naming the kind of communication you are willing to accept.

RECEIVE Command Arguments

ADVICE — allows both assistance and communication links initiated by another user's ADVISE or TALK command.

LINK — allows communication links established by another user's TALK command.

SYSTEM-MESSAGES — allows notices of new mail and messages of interest to all users sent by the operator or other privileged users.

USER-MESSAGES — allows messages sent by another user's SEND command.

Default - LINKS, SYSTEM-MESSAGES, and USER-MESSAGES

Hint

Typing RECEIVE During Attempted TALK

If your terminal has been set to refuse links and another user tries to talk to you by using the TALK command, both terminals will give a series of CTRL/G signals (ringing bells or beeps) indicating the refused attempt. If you give the RECEIVE LINKS command before these signals are finished, the TALK command will succeed.

Related Commands

ADVISE	for sending commands to another user's job
INFORMATION TERMINAL-MODE	for examining your current terminal settings
REFUSE	for refusing communication links, advice, and system and user messages
SEND	for sending a message to another user's terminal
TALK	for linking your terminal to another user's terminal
TERMINAL INHIBIT	for refusing all types of terminal communication including links, advice, system messages, user messages, and notices of new mail.

Examples

1. Give the RECEIVE command to accept communication links from other users.

   ```
   @RECEIVE LINKS
   ```

2. Set your terminal to receive links, at the request (sent via the MAIL program, not shown here) of another user. Begin a communication session with this user, during which you give the RECEIVE ADVICE command also, to allow a demonstration of the UDP program. Afterwards, set your terminal again to refuse advice.

   ```
   @RECEIVE
   @
   LINK FROM RENQUIST, TTY 127
   @;THANKS, BUT IF YOU LET ME DO AN "ADVISE" I CAN SHOW YOU
   @;HOW TO RUN THE PROGRAM BY ACTUALLY DOING IT.  OKAY?
   @;SURE, I'LL FIX MY SETTING.
   @RECEIVE ADVICE
   @ADVISE LATTA
    Escape character is <CTRL>E, type <CTRL>^? for help
    LATTA, MISC:<LATTA> Job 33 EXEC
    [Advising]
   UDP
   UDP>LIST/DOCUMENTATION:/CREATED-SINCE:1-1-78 0:0
   ```

```
UDP>EXIT
@;YOU'LL GET A PRINTED LISTING TOMORROW.
@;DO YOU SEE HOW I DID IT?
@;YES, THANKS.  GOODBYE.
@
 [Advice terminated]

@REFUSE ADVICE
```

2.61 REENTER

Starts the current fork at its alternate entry point.

Format

@**REENTER** (PROGRAM)

Characteristics

Using REENTER

The REENTER command starts your program at the address specified by the second word in the program's entry vector. For most programs this address is contained in location 124. Usually the REENTER and START commands start the program at the same point, but another re-entry point can be provided to avoid initialization procedures, perform error recovery, or to use the program in a different way.

Hints

Further Information

For more information about entry vectors, see the TOPS-20 Monitor Calls Reference Manual.

Related Commands

GET	for placing an executable program in memory
LOAD	for loading a source or object program into memory
START	for entering a program at its normal entry point

Examples

1. Give the REENTER command for your current program.

   @REENTER

2. Begin running a program, then give a CTRL/C to leave it and obtain a file. Resume execution of the program at the alternate entry point.

```
@R DUMPER
DUMPER>^C
@ACCESS PS:<P.SPECCINI>
Password:___
@COPY PS:<P.SPECCINI>USR.FIL
 PS:<P.SPECCINI>USR.FIL.1 => USR.FIL.1 [OK]
@END-ACCESS PS:<P.SPECCINI>
@REENTER
DUMPER>
```

2.62 REFUSE

Notifies the system that you are not willing to accept communication links, advice, user messages, and system messages.

Format

@REFUSE argument

where:

argument	is a keyword, chosen from the list below, naming the kind of communication you are not willing to accept.

REFUSE Command Arguments

ADVICE	prevents assistance initiated by another user's ADVISE command.
LINKS	prevents both assistance and communication links from being established by another user's ADVICE or TALK command.
SYSTEM-MESSAGES	prevents notices of new mail and messages of general interest sent to all users by the monitor or by the operator or other privileged users.
USER-MESSAGES	prevents messages sent by another user's SEND command.

Default - ADVICE and LINKS

Hints

Refusing All Communication

Users with Wheel or Operator capabilities enabled can ADVISE, SEND, and TALK to terminals that have refused advice, user-messages, and links. To refuse messages and links from privileged users, use the TERMINAL INHIBIT command.

Typing RECEIVE During Attempted TALK

If your terminal has been set to refuse links and another user tries to talk to you by using the TALK command, both

terminals will give a series of CTRL/G signals (ringing bells or beeps) indicating the refused attempt. If you give the RECEIVE LINKS command before these signals are finished, the TALK command will succeed.

Safeguarding Terminal Output

If you want your terminal to print a long file, without interference, use the TERMINAL INHIBIT command to prevent all classes of message from being received. Be sure to use the TERMINAL NO INHIBIT command afterwards to restore the previous condition of your terminal.

Special Cases

Implicit Refusal of Advice

If you give the REFUSE LINKS command, your terminal will be set to refuse advice also. However, the INFORMATION TERMINAL-MODE command may not display this setting unless you give an explicit REFUSE ADVICE command as well.

Privileged Disregard of REFUSE

A user with enabled Wheel or Operator capabilities can give the TALK or ADVISE command for any job.

Related Commands

ADVISE	for sending commands to another user's job
INFORMATION TERMINAL-MODE	for examining your current terminal settings
RECEIVE	for receiving communication links, advice, and system and user messages
SEND	for sending a message to another user's terminal
TALK	for linking your terminal to another user's terminal
TERMINAL INHIBIT	for refusing all types of terminal communication including links, advice, system messages, user messages, and notices of new mail.

Examples

1. Use the REFUSE command to prevent other users from advising your job.

   @REFUSE ADVICE

2. Receive a communication link formed by another user's TALK command. Confer with him briefly, then set your terminal to refuse all classes of message over which you have control.

   ```
   LINK FROM RENQUIST, TTY 127
   @!HELLO DAVID. CAN YOU HELP ME WITH EDIT?
   @!SORRY, PLEASE BREAK.  I'M EXPECTING PRINTOUT AND THIS
   @!WILL INTERFERE.  WILL GET IN TOUCH LATER.
   @!OKAY
   @BREAK
   @REFUSE LINKS
   @REFUSE SYSTEM-MESSAGES
   @REFUSE USER-MESSAGES
   ```

3. As an alternative to giving three REFUSE commands as in the previous example, give the TERMINAL INHIBIT command. This command refuses all types of messages.

   @TERMINAL INHIBIT

2.63 REMARK

Tells the system to regard the terminal input that follows as comment only.

Format

@REMARK (MODE)
Type remark. End with CTRL/Z.

Characteristics

Ending Remarks

Until you give a CTRL/Z, the system merely displays what you type, instead of trying to interpret it as commands.

Hints

Useful During TALK or ADVISE Session

If you have already established contact with another user by a TALK or ADVISE command before giving REMARK, his terminal will also display what you type. Give the REMARK command before sending lengthy comments or demonstrating commands that you don't want to take effect.

Related Commands

ADVISE for sending commands to another user's job

TALK for sending comments to another user

Examples

1. Give the REMARK command.

   @REMARK
   Type remark. End with CTRL/Z.

2. Receive a communication link from another user. Give the REMARK command to speak with him. Give a CTRL/Z afterwards to end the remarks.

   LINK FROM P.SPECCINI, TTY 127
   @;WHERE ARE THE NOTES FROM THE LAB DEMO THIS A.M.?

```
@REMARK
Type remark. End with CTRL/Z.
HI, PAUL.  THEY'RE IN THE LAB'S LIBRARY AREA.
THAT'S CHEM:<P-CHEM.20.NOTES>.  I DON'T KNOW
THE TITLE BUT LOOK AT THE DATES WITH A
TDIRECTORY COMMAND.  OKAY?
;YES, THANKS.  BYE
@BREAK
^Z
```

2.64 RENAME

Changes the name of a file.

Format

@RENAME (EXISTING FILE) **old filespec(s)** (TO BE) **new filespec**

where:

old filespec(s) — is a single file specification, or a series of them separated by commas and/or indicated by wildcard characters (% and *).

new filespec — is the new specification under which you want to store the file(s); the new specification must be on the same structure; you may include an asterisk (*) if you gave more than one old filespec.

Default new filespec - old filespec, but with a generation number higher by 1 than the highest existing generation number

Output

Status of Files

If you use recognition on the new file specification, the system prints !Old Generation!, !New Generation!, or !New File!, to describe its status.

Confirmation of Action

As each file is renamed, the system prints its old and new specification, and the word [Superseding] if it is replacing previous contents, and finally the word [OK]. The delay before you see this [OK] indicates how long it took to rename the file.

Hints

Specifying a New Account and Protection Number

You can specify the new file's protection number and the account to which its storage fees will be charged. Follow the new filespec with a semicolon (;) and the letter P before giving a new 6-digit protection number, and with a

semicolon and the letter A before giving a new account. Ordinarily these values are set to the default file protection and current account. However, non-default protection numbers will be maintained for higher generations of existing files, unless you specify otherwise in the RENAME command that creates that higher generation.

RENAME Faster Than COPY for Transferring Files

For moving a set of files from one directory to another on the same structure, the RENAME command is a faster and more efficient means than COPY. This is because RENAME only changes the file specifications; it does not copy the contents of the files. Also, a file transfer with the RENAME command leaves only one set of files, while a transfer with the COPY command leaves two sets: the original copies and the destination copies. The original copies are often unnecessary and must be deleted.

Restrictions

Renaming Between Structures

You cannot rename a file from one structure to another, but must use the COPY command to reproduce its contents on the new structure, then the DELETE command to remove it from the old structure.

Renaming Open or Mapped Files

You cannot rename a file that is open or mapped into memory. First give the RESET command, or POP followed by RESET, if this is the case.

Renaming Archived Files

You can rename an archived file by specifying it as the first (or old) argument of a RENAME command. It will then have the second (or new) argument as its specification and will remain an archived file. However, you cannot give the specification of an archived file as the second argument of a RENAME command, as this would replace the file's contents. If you attempt to do so, the file you specify as the first argument will be renamed to a generation higher by 1 than the highest existing generation of the archived file, leaving the archived file intact.

Warning

Replacing Previous Contents of Files

If you rename a file into a specification (including generation number) that already exists, the previous contents of the new file are replaced and cannot be recovered. But see Restrictions - Renaming Archived Files, above.

Related Commands

COPY for making copies of files

Examples

1. Rename a file.

```
@RENAME ATM-50.SPC ATM-50.PRL
 ATM-50.SPC.1 => ATM-50.PRL.1 [OK]
```

2. Use a wildcard character to rename all files of a given name.

```
@RENAME ATM-50.* 1-ATM-50.*
 ATM-50.BAK.1 => 1-ATM-50.BAK.1 [OK]
 ATM-50.PRL.1 => 1-ATM-50.PRL.1 [OK]
```

3. Access another user directory and transfer to it the files renamed in Example 2.

```
@ACCESS <ORBEN>
Password:___
@RENAME 1-ATM-50.* <ORBEN>
 1-ATM-50.BAK.1 => <ORBEN>1-ATM-50.BAK.1 [OK]
 1-ATM-50.PRL.1 => <ORBEN>1-ATM-50.PRL.1 [OK]
@END-ACCESS <ORBEN>
```

2.65 RESET

Clears memory of the specified forks.

Format

@**RESET** (FORK) argument

where:

argument is one of the following:	Fork name Fork number * for all forks . (period) for the current fork **Default** - all unkept forks

Characteristics

Action of RESET

In addition to clearing memory for the specified forks, the RESET command closes all files, mapped and unmapped, opened by the specified forks and their inferior forks. RESET also simultaneously terminates the specified fork's inferior forks.

Effect on Memory

The RESET command clears the specified forks from memory.

Related Commands

INFORMATION FILE-STATUS	for determining which files are currently open
INFORMATION MEMORY-USAGE	for determining contents of memory
CONTINUE, FORK, FREEZE INFORMATION, FORK-STATUS, INFORMATION PROGRAM-STATUS, KEEP, RESET, SET NAME, SET PROGRAM, and UNKEEP	other multiforking-class commands

Examples

1. Give the RESET command to clear all unkept forks from memory.

   ```
   @RESET
   ```

2. Clear all forks, including kept forks, from memory.

   ```
   @RESET *
   ```

3. Display the fork status with INFORMATION FORK-STATUS. Then, clear the SORTER fork from memory. Redisplay the fork status to check the result.

   ```
   @INFORMATION FORK-STATUS
       EDT (1): Kept, ^C from IO wait at 413773, 0:00:00.0
    => PASCAL (2): Kept, Background, Running at 324004, 0:00:00.8
       SORTER (3): HALT at 400370, 0:00:00.6
   @RESET 3
   @INFORMATION FORK-STATUS
       EDT (1): Kept, ^C from IO wait at 413773, 0:00:00.0
       PASCAL (2): Kept, Background, Running 453004, 0:00:01.3
   ```

2.66 RETRIEVE

Returns an off-line file (magnetic tape copy of a file) to disk.

Format

@**RETRIEVE** (FILES) **filespec,...**

where:

filespec	is the specification of any off-line file (archived or not, visible or invisible) to which you have access; you may include wildcard characters (% and *).

Output

Acknowledgment of Request

As soon as you complete a valid RETRIEVE command, the system responds by printing, on your terminal, the specification of each off-line file for which you requested retrieval, followed by [OK].

Notice of Retrieval Sent to Requestor

Depending on the procedures at your site, when the files for which you have requested retrieval have been restored to their directory on disk, you may receive a mail message that contains the names of each retrieved file. Remember that, depending on how frequently your site processes retrieval requests, this message may not be sent until one or more days after your request.

Characteristics

Invisibility of Retrieved Files

If you retrieve invisible files, they will remain invisible (whether archived or not) when restored to disk. Use the SET FILE VISIBLE command to make invisible files visible. Until you do so, they will be inaccessible to most TOPS-20 commands.

Hints

Using Retrieved Archived Files

As long as a retrieved archived file is visible, you can inspect it using the TYPE or PRINT command, or list its specifications using DIRECTORY-class commands. However, you cannot add to it or change it (for example, by using APPEND). To make changes to a copy of a retrieved archived file, first use the COPY command to copy it to a new specification. If you wish, you can then request archival for this new file (using the ARCHIVE command) and delete the old one (using the DELETE command with the ARCHIVED subcommand). You can return an (unchanged) on-line archived file to off-line status by using the DELETE command with the CONTENTS-ONLY subcommand, or withdraw archive status from the file (make it an ordinary disk file) by using the DISCARD command.

Using Retrieved Non-archived Files

As long as a retrieved non-archived file is visible, you can use TOPS-20 commands with it as with any other disk file. The only difference is that after any command that has changed the file, the tape copy of the file is no longer valid. This means that you cannot give the DELETE command with the CONTENTS-ONLY subcommand to return the file to off-line status.

Special Cases

Implied Retrieval Requests

If your system has enabled the "automatic retrieval-wait" feature (give the INFORMATION SYSTEM-STATUS command to find out whether it has), and the SET RETRIEVAL-WAIT command is in effect for your job, any command that attempts to use an off-line file will create an automatic retrieval request for that file. Under these conditions, commands such as TYPE or COPY for which you specify off-line files will not be executed until those files are retrieved. Implied retrieval requests are most useful in batch jobs.

Related Commands

ARCHIVE	for requesting archival of specified files
CANCEL RETRIEVE	for canceling retrieval

	requests before they are filled
DELETE (with CONTENTS-ONLY subcommand)	for deleting the disk contents only of retrieved (on-line) files
DIRECTORY (with OFFLINE subcommand)	for listing the specifications of visible off-line files
DIRECTORY (with OFFLINE and INVISIBLE subcommands)	for listing the specifications of invisible off-line files
DIRECTORY (with TIMES TAPE-WRITE subcommand)	for finding out the write date of the tape copy of files
DISCARD	for giving up the tape copy of retrieved files
INFORMATION RETRIEVAL-REQUESTS	for finding out the status of retrieval requests

Examples

1. Retrieve an off-line file.

```
@RETRIEVE BRCHIVE.TXT
 BRCHIVE.TXT.1 [OK]
```

2. Attempt to use a file. Upon discovering that it is off-line, retrieve the file. When it has been restored to your directory, discard the tape copy of the file, and then have it printed on your terminal.

```
@TYPE FILBRK.HLP
?File is off-line: FILBRK.HLP.1
@RETRIEVE FILBRK.HLP
 FILBRK.HLP.1 [OK]
     .
     .
     .
@DISCARD FILBRK.HLP
 FILBRK.HLP.1 [OK]
```

```
@TYPE FILBRK.HLP
!THIS IS JUST A TEXT FILE TESTER.
```

3. Get a listing of your archived files. Retrieve one that is off line, examine it, and return it to off-line status.

```
@DIRECTORY,
@@ARCHIVE
_@@

   MISC:<GOLDEN>
 ARCHEK.FIL.1
 ARCHIVE.ALSO.1;OFFLINE
   .NOT.1;OFFLINE
   .TOO.1;OFFLINE
 MOOBE.TXT.1;OFFLINE
 TESTY.BBN.1,2

 Total of 6 files
@RETRIEVE BRCHIVE.TXT
 BRCHIVE.TXT.1 [OK]
     .
     .
     .
@TYPE BRCHIVE.TXT
!A TEXT FILE TESTER
@DELETE BRCHIVE.TXT,
@@CONTENTS-ONLY
@@
MISC:<GOLDEN>BRCHIVE.TXT.1 [OK]
MISC:<GOLDEN> [1 page freed]
```

4. Get an inclusive listing of your off-line files, including the date the tape copy was written. Retrieve three of them, and check the requests in the retrieval queue. Cancel one of the requests.

```
@DIRECTORY,
@@OFFLINE

@@TIMES TAPE-WRITE
@@


   MISC:<GOLDEN>
                Tape-write

 ARCHIVE.ALSO.1;OFFLINE  8-Jun-85 07:59:08
   .NOT.1;OFFLINE  8-Jun-85 07:59:09
 DUMPER.MAC.1;OFFLINE  7-Mar-85 05:19:10
```

```
 PRODUK.EXE.4;OFFLINE  7-Mar-85 05:19:13
 SQUARE.EXE.1;OFFLINE  7-Mar-85 05:19:14

 Total of 5 files
@DIRECTORY,
@@OFFLINE
@@INVISIBLE
@@TIMES TAPE-WRITE
@@
   MISC:<GOLDEN>
                Tape-write

 ARCHIVE.TOO.1;OFFLINE  8-Jun-79 07:59:10
 BRCHIVE.TXT.1;OFFLINE 27-Jun-79 04:04:58
 ERCHIVE.TXT.1;OFFLINE  8-Jun-79 07:59:11
 FRCHIVE.TXT.1;OFFLINE 12-Jul-79 03:23:03
 MOOBE.TXT.1;OFFLINE  8-Jun-79 07:59:12

 Total of 5 files
@RETRIEVE PRODUK.EXE, FRCHIVE.TXT, MOOBE.TXT
 PRODUK.EXE.4 [OK]
 FRCHIVE.TXT.1 [OK]
 MOOBE.TXT.1 [OK]
@INFORMATION RETRIEVAL-REQUESTS

Retrieval Queue:
 Name   Req#    Tape 1  Tape 2  User
------  ----    ------  ------  -------
MOOBE    507      5329    5520  GOLDEN
PRODUK   505      5538    5583  GOLDEN
FOOBAR   407      5845    5856  TOMCZAK
EE155    442      6279    5883  WRIGHT
BRCHIV   504      5543    7138  GOLDEN
FRCHIV   506      7138    7559  GOLDEN
There are 6 jobs in the queue (none in progress)

@CANCEL RETRIEVE 507
[1 Job canceled]
```

2.67 REWIND

Returns a magnetic tape to its load point (logical beginning, the beginning of the first file).

Format

@REWIND (DEVICE) **dev:** /switch

where:

dev: is the name of the tape set or magnetic tape drive that you want to rewind. The colon after the device name is optional.

/switch is one of the following:

/CURRENT-VOLUME-ONLY
rewinds tape set to beginning of currently mounted volume

/ENTIRE-VOLUME-SET
rewinds tape set to beginning of first volume
Default - ENTIRE-VOLUME-SET

Note: these switches can be used only for devices of the form MTn:, not MTAn:.

Restrictions

REWIND with Open Files

If you have given a CTRL/C to exit from a program that has opened a magnetic tape set and you then give the REWIND command for that tape set, the system will first ask if you want to close the associated file. You must do so for REWIND to succeed, but will probably be unable to continue the program from that point because the file will now be closed.

Related Commands

BACKSPACE for moving a magnetic tape backward a specified number of files or records

DIRECTORY (when used with a magnetic tape device) for

rewinding a tape set, printing a directory of its files, and again rewinding the tape set

SKIP — for moving a magnetic tape forward a specified number of files or records

UNLOAD — for rewinding a magnetic tape completely onto the source reel

Examples

1. Rewind your magnetic tape.

```
@REWIND DAY:
```

2. Mount a tape, and prepare to copy files onto it. (Use the REWIND command to be sure you are at the beginning.) After copying the files, rewind the tape and (using the COPY command) read the first one. Then give TOPS-20 commands to free the resources you have been using.

```
@MOUNT TAPE DAY:
[Mount Request DAY Queued, Request-ID 183]
[Tape set DAY, volume DAY mounted]
[DAY: defined as MT0:]
@MOUNT STRUCTURE SNARK:
Structure SNARK: mounted
@ACCESS SNARK:
@REWIND DAY:
@COPY SNARK:FIL-1.TAP DAY:
 SNARK:FIL-1.TAP.1 => MT0:FIL-1 [OK]
@COPY SNARK:FIL-2.TAP DAY:
 SNARK:FIL-2.TAP.1 => MT0:FIL-2 [OK]
@COPY SNARK:FIL-3.TAP DAY:
 SNARK:FIL-3.TAP.1 => MT0:FIL-3 [OK]
@REWIND DAY:
@COPY DAY: TTY:
 MT0: => TTY:

!THIS IS THE FIRST FILE.!
@DISMOUNT TAPE DAY:
[Tape dismounted, logical name DAY: deleted]
@END-ACCESS SNARK:
@DISMOUNT STRUCTURE SNARK:
Structure SNARK: dismounted
@
```

2.68 RUN

Places an executable program in memory and starts it.

Format

@RUN (PROGRAM) **filespec /switch**

where:

filespec — is the file specification of any executable program.
Default dev:<dir> - DSK:

Default .typ - .EXE

/switch — is /USE-SECTION:n
specifies the memory section (from 0 to 37 octal) in which your program is to run. You can use this switch only if your program can be contained in one section.

Characteristics

Efficiency of RUN

The RUN command does the work of the pair of commands GET and START. It is a faster and less expensive means of executing programs than EXECUTE, or than LOAD and START. Therefore you should store frequently-run programs in .EXE files and run them with this command.

Hints

Alternative to RUN command

When you type only a program name, the system looks for a matching system program. When you precede the program name with the RUN command, the system looks for the program in your connected directory. The RUN command can be eliminated by typing the directory name with the program name. These two commands for example, each run SORTER located in the connected directory:

```
@RUN SORTER.EXE

@<SMITH>SORTER.EXE
```

Effect on Memory and Terminal

The RUN command clears any unkept forks, places the specified program in memory, starts it and leaves your terminal at command level in the program (if any), or at TOPS-20 command level.

Related Commands

ERUN	for running a system program without disturbing the program already in memory
EXECUTE	for running source or object programs
GET	for placing an executable program in memory
R	for running executable programs stored on SYS:
SAVE	for saving a program in executable (.EXE) format
START	for starting the program currently in memory

Examples

1. Run one of your executable programs.

   ```
   @RUN TESTF1.EXE
   ```

2. Mount a structure and access a user's directory on the structure. Run one of his programs.

   ```
   @MOUNT STRUCTURE SNARK:
   Structure SNARK: mounted
   @ACCESS SNARK:<ELDRIDGE>
   Password:
   @RUN SNARK:<ELDRIDGE>FT.EXE
   ```

2.69 SAVE

Stores a copy of memory in an executable file.

Format

@SAVE (ON FILE) **filespec** (PAGES FROM) **loc1** (TO) **loc2, loc3 loc4, ...**

where:

filespec	is the file specification under which you want to store the program. **Default** filespec - program name.EXE
loc1 loc2, loc3 loc4, ...	are pairs of octal numbers or symbolic expressions that specify the span(s) of memory pages you want to save. **Default** loc1 loc2 - all **assigned** pages of memory from 0 to the highest page number of the highest existing section

Output

Status of Files

If you use recognition of the file specification, the system prints !Old Generation!, !New Generation!, or !New File!, to indicate its status on disk, or !OK! if saved on a non-disk device.

Hints

Saving Programs Before Running Them

When you load a source or object program using the LOAD command, save it using SAVE before running it. Then you can run it in the future using RUN, without first loading it using a LOAD-class command. This is also true if you save the program after running it, but it will then be in a post-run state.

More Information

For more information about saved files, see the TOPS-20 Monitor Calls Reference Manual.

Restriction

Saving an Execute-only Compiler

It is illegal to use the SAVE command after using the LOAD command for an execute-only compiler. An alternative is to use LINK with the LOAD command and the /SAVE switch.

Related Commands

GET	for putting a saved file into memory
LOAD	for putting a source or object file into memory
RUN	for running a saved program
START	for starting the program in memory

Examples

1. Save the program currently in memory.

```
@SAVE
 TESTF1.EXE.6 Saved
```

2. Mount a magnetic tape in write-enabled mode. Use the GET command to put an executable program into memory, then save it (specifying a new filename) on tape and on disk. Finally, start the program, which is still in memory.

```
@MOUNT TAPE LAT:/WRITE-ENABLED
[Mount request LAT: queued, request-ID 415]
[Tape set LAT, volume LAT mounted]
[LAT: defined as MT2:]
@GET TESTF1
@SAVE LAT:TAP.EXE
 MT2:TAP.EXE Saved
@SAVE
 TAP.EXE.1 Saved
@START

THIS IS A TEST.

CPU time: 0.04  Elapsed time: 0.17
```

2.70 SEND

Sends a message immediately to another user's terminal.

Format

@SEND (TO) **/switch argument message-text**

where:

argument — is one of the following: a user name, a terminal line number, an asterisk (*) [for all terminals]

(The asterisk argument requires WHEEL or OPERATOR privileges.)

message — is a message of up to six lines of text followed by a carriage return.

switch — is /NODE:node-name which specifies a node in the TOPS-20 cluster to send the message to.

Note that the privileged ^ESEND command can also send a message to all terminals on all nodes of the argument (see the TOPS-20 Operator's Command Language Reference Manual).

Characteristics

Multiple-line Messages

SEND allows you to send multiple-line messages with up to six 80-character lines of text. Type the message past the end of the first line and onto the next line without typing RETURN. SEND will reorganize your message on the receiver's terminal so that words broken across two lines appear on the same line.

Refused SEND

You cannot contact a user with SEND if his terminal is set to refuse messages with the REFUSE USER-MESSAGES command or the TERMINAL INHIBIT command. Normally, if you attempt to SEND to a user who has refused user messages, the system prints the message ?User is refusing messages and/or links.

However, a user with Wheel or Operator capabilities enabled, can SEND messages to users who have given a REFUSE USER-MESSAGES command, but not the TERMINAL INHIBIT command.

Hints

Finding the Receiver's Line Number

To find the terminal line number for the receiver of a SEND message, give the SYSTAT command with the receiver's user name as an argument.

Sending Terminal Bells

To get the attention of the user at the receiving terminal, type a few CTRL/Gs in your message. This will ring the terminal bell on the receiving terminal.

SEND in a Batch Job

You can place SEND commands in a batch control file to send messages to your terminal on the condition of a running batch job. Since your username is associated with your batch job and your timesharing job, use the line number argument.

SEND as an Alternative to Mail

The SEND command can be used as an alternative to sending a message with one of the mail programs. For short messages, SEND can be more convenient and faster to use than a mail program. For urgent messages, a SEND message is read immediately by the receiver, unlike a mail message, which can be read at the user's leisure.

Special Cases

User Has More Than One Job

If you attempt to SEND a message to a user who is logged-in on more than one terminal, the system responds with a list of the user's terminal line numbers and the programs being run at each terminal. Type your choice of terminal line number (if available, one running the EXEC) after the TTY: prompt.

Related Commands

ADVISE	for sending commands to another user's job
RECEIVE USER-MESSAGES	for receiving another user's SEND message
REFUSE USER-MESSAGES	for refusing another user's SEND message
REMARK	for telling the system to regard your terminal input as comment only
TALK	for linking your terminal to another user's terminal
TERMINAL INHIBIT	for refusing all types of terminal communication including links, advice, system messages, user messages, alerts, and mail notices

Examples

1. Send a message with the SEND command.

   ```
   @SEND 141 PAUL, DO YOU HAVE THE TCO TAPE?
   ```

2. Send a multiple-line message. Type the message past the end of the line and onto the next. Press RETURN only at the end of the message.

   ```
   @SEND ASMITH AL, I CAN'T MAKE THE NORTH PROJECT MEETING;
   HERMAN JUST CALLED A STAFF MEETING. I'LL WRITE MY ENGINE
   ERING STATUS AND SEND IT TO YOU BEFORE LUNCH. - BILL
   ```

3. Enable your Wheel or Operator privileges and send a message to all users.

   ```
   @ENABLE
   $SEND * THE LETTER QUALITY PRINTER IS UP
   From NELSON on line 127:
   [THE LETTER QUALITY PRINTER IS UP]
   $DISABLE
   @
   ```

4. Place SEND commands in your batch control file to monitor the progress of your batch job.

   ```
   @IF (ERROR) @SEND 122 Error in PROJEC batch run
        .
        .
        .
   ```

```
14015 @SEND 122 PROJEC batch job almost done
```

5. Send a message to a user who has two jobs on two different terminals. Send the message to the terminal that is running the EXEC.

```
@SEND JOHNSON Ready for lunch?
 TTY20, running EXEC
 TTY4, running PASCAL
TTY: 20
```

6. Send a message to a user on remote node VENUS.

```
@SEND /NODE:VENUS ANDERSON Don't forget the meeting!
```

2.71 SET

Sets or modifies various characteristics of your job, a directory, a file, a device, or some other entity.

Format

@SET argument(s) setting(s)

where:

argument — is a keyword, chosen from the list below, indicating your choice of SET command options.

setting — is a word or number, required to complete the meaning of most SET commands.

Summary of SET Command Arguments (defaults in boldface)

ACCOUNT account remark
ADDRESS-BREAK octal or symbolic memory address,
@@**AFTER** n **Default** n - 1
@@**ALL**
@@EXECUTE
@@NONE
@@READ
@@WRITE
ALERT date/time message
AUTOMATIC

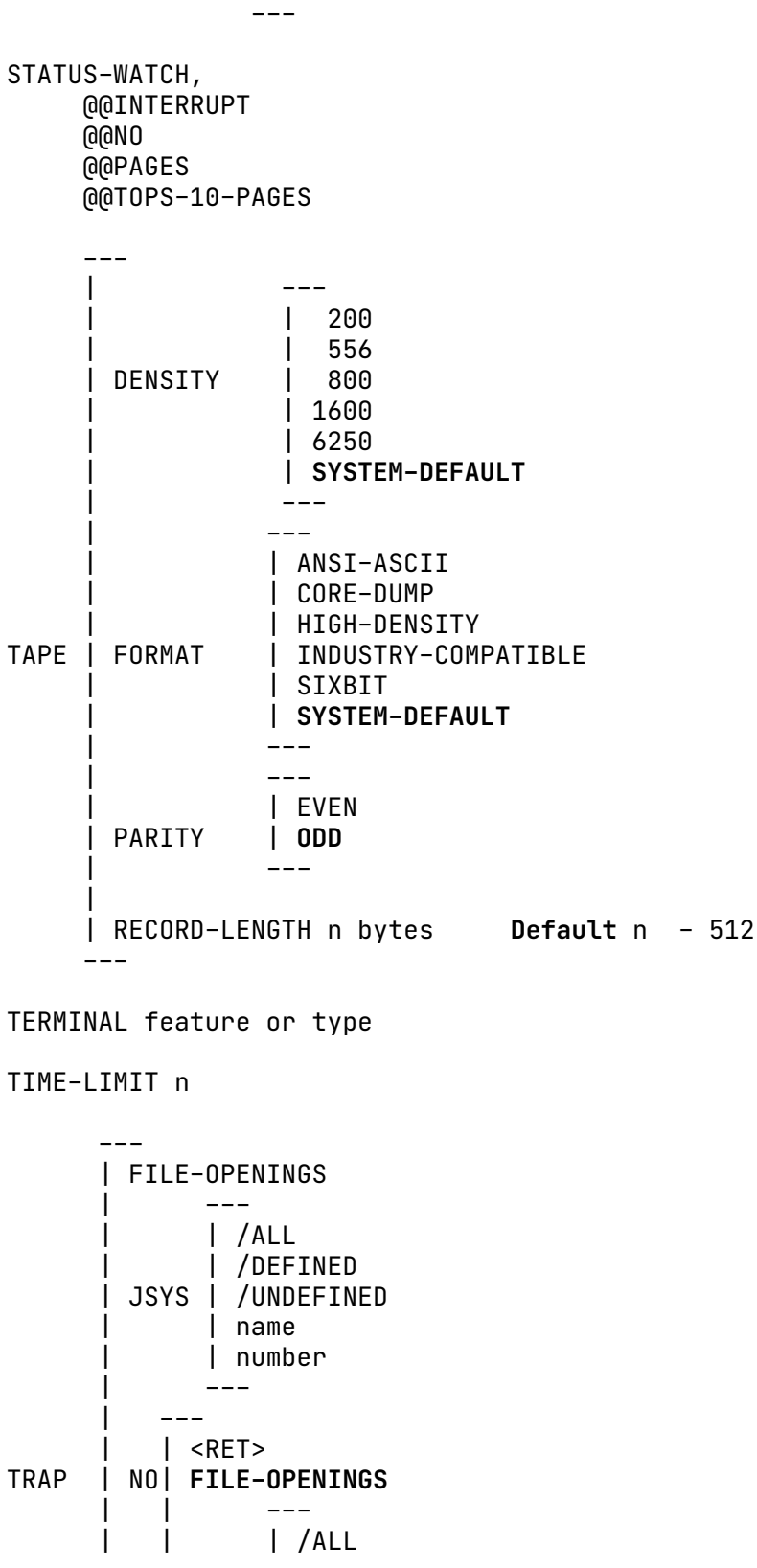
CARD-READER-INPUT-SET name of input set n
CONTROL-C-CAPABILITY

```
         ---
         | CARDS  /switch(es)
         | COMPILE-SWITCHES file type  /switch(es)
         | PAPER-TAPE  /switch(es)
DEFAULT  | PLOT  /switch(es)
         | PRINT  /switch(es)
         | SUBMIT  /switch(es)
         |     ---
         |     | ALLOW
         |TAKE | DISALLOW
         ---   |
               | ECHO
               | NO ECHO
               ---
```

```
                  ---
                  | ACCOUNT-DEFAULT dev:<directory> account
                  | password
                  |
                  | ARCHIVE-ONLINE-EXPIRED-FILES
                  | FILE-PROTECTION-DEFAULT dev:<directory> octal code
                  | password
                  |      Default code - 777700
                  | GENERATION-RETENTION-COUNT-DEFAULT
                  | dev:<directory> n password
                  |      Default n - 1
        DIRECTORY | NO ARCHIVE-ONLINE-EXPIRED-FILES
|                 | NO SECURE
                  | OFFLINE-EXPIRATION-DEFAULT dev:<directory>date or +n
                  |      Default n - 90
                  | ONLINE-EXPIRATION-DEFAULT dev:<directory>date or +n
                  |      Default n - 60
                  | PASSWORD dev:<directory>
                  | old password
                  | new password
                  | new password
                  |
                  | PROTECTION dev:<directory> octal protection code
                  | password
                  |      Default code - 777700
|                 | SECURE str:<directory>
                  ---


                    ---               ---
        ENTRY-VECTOR| octal or symbolic | octal or symbolic length
                    | memory location   | between 1 and 777
                    ---               ---
                           Default length - 1

                 ---
                 | ACCOUNT  filespecs account
                 | EPHEMERAL filespecs
                 | EXPIRED filespecs
                 | GENERATION-RETENTION-COUNT filespecs n  Default n - 1
                 | INVISIBLE
                 |    EPHEMERAL filespecs
|                |    PERMANENT filespecs
                 | NO PROHIBIT filespecs
        FILE     |    RESIST filespecs
|                |    SAVE-BY-BACKUP-SYSTEM filespecs
|                |    SECURE filespecs
|                |    PERMANENT filespecs
|                |    TEMPORARY filespecs
|                |    UNDELETABLE filespecs
```

```
                | OFFLINE-EXPIRATION filespecs date or +n
                | ONLINE-EXPIRATION filespecs  date or +n
|               | PERMANENT
                | PROHIBIT filespecs
                | PROTECTION filespecs   octal protection code
                |           Default code - 777700
                | RESIST filespecs
|               | SAVE-BY-BACKUP-SYSTEM
|               | SECURE filespecs
|               | TEMPORARY
|               | UNDELETABLE
                | VISIBLE
                ---

                      ---
        HOST node-name::  | /CTERM
                          | /NRT
                          ---
        LATE-CLEAR-TYPEAHEAD
        LOCATION node-name  Default node-name - your host node
        MAIL-WATCH user-name message-count
        NAME fork-name

             ---
             | ADDRESS-BREAK
             | ALERT
             | AUTOMATIC
             | CONTROL-C-CAPABILITY
             |          ---
             |          | CARDS
             |          | COMPILE-SWITCHES  file type or *
             |          | PAPER-TAPE
             | DEFAULT  | PLOT
|            |          | PROGRAM
             |          | PRINT
        NO   |          | SUBMIT
             |          ---
             |
             | LATE-CLEAR-TYPEAHEAD
             | MAIL-WATCH user-name
             | RETRIEVAL-WAIT
|            | STATUS-WATCH
             | TIME-LIMIT
             |      ---
             |      | <RET>
             | TRAP | FILE-OPENINGS
             |      |     ---
             |      |     | /ALL
             |      |     | /DEFINED
             |      | JSYS| /UNDEFINED
             |      ---   | name
```

```
        |            | number
        |            ---
        | UUO-SIMULATION
        ---

                                          ---
                                          | COPY-ON-WRITE
                                          | EXECUTE
                                          |
                                          |    ---
                                          |    | COPY-ON-WRITE
PAGE-ACCESS  octal page numbers           | NO | WRITE
                                          |    ---
                                          |
                                          | NONEXISTENT
                                          | READ
                                          | WRITE
                                          ---

|       PASSWORD dev:<directory>
|       old password
|       new password
|       new password
|
                          ---
                          | EPHEMERAL
                          |      ---
                          |      | CONTINUE
PROGRAM fork-name         | KEEP | REENTER
                          |      | START
                          |      ---
                          | NO-EPHEMERAL
                          | NONE
                          ---

                          CHARACTERISTICS name value
                                       ---              --- ---
REMOTE-PRINTING           PRINTER name | remote queue     | | DQS node
                                       | LATserver port   | | LAT server
                                       | LATserver service| ---
                                       | alias            |
                                       ---              ---
                          SYSTEM-DEFINITIONS

RETRIEVAL-WAIT

SESSION-REMARK remark of up to 39 characters

                       ---
                       | IMMEDIATE
SPOOLED-OUTPUT         | DEFERRED
```


STATUS-WATCH,
@@INTERRUPT
@@NO
@@PAGES
@@TOPS-10-PAGES
DENSITY
200
556
800
1600
6250
SYSTEM-DEFAULT
TAPE
FORMAT
ANSI-ASCII
CORE-DUMP
HIGH-DENSITY
INDUSTRY-COMPATIBLE
SIXBIT
SYSTEM-DEFAULT
PARITY
EVEN
ODD
RECORD-LENGTH n bytes
Default n - 512
TERMINAL feature or type
TIME-LIMIT n
TRAP
NO
FILE-OPENINGS
JSYS
/ALL
/DEFINED
/UNDEFINED
name
number
<RET>
FILE-OPENINGS
/ALL

```
      |   |      | /DEFINED
      |   | JSYS | /UNDEFINED
      |   |      | name
      |   |      | number
      |   |      ---
      |   | PROCEED
      |   ---
      | PROCEED
      ----

                ---
TYPEOUT MODE    | NUMERIC
                | SYMBOLIC
                ---
UUO SIMULATION
```

SET Command Arguments

ACCOUNT account remark
: begins charging the specified account for the remainder of your current terminal session or until you use the command again. You must supply an alphanumeric account name of 39 or fewer characters valid for your user name. Then you can type an optional session remark, also of 39 or fewer characters, to be inserted in system accounting data for your current terminal session. Check your current account and session remark with INFORMATION JOB-STATUS.

ADDRESS-BREAK octal or symbolic memory location,
@@AFTER n
@@ALL
@@EXECUTE
@@NONE
@@READ
@@WRITE
: causes the program in memory to be suspended and a message to be printed on your terminal when the memory location you specify is referenced for the indicated operation - execute, read, write, or any of these (ALL). With the AFTER subcommand you determine how many times it must be referenced before the address break occurs; with NONE you cancel address breaks for the specified location, just as with the SET NO ADDRESS-BREAK command. Each SET ADDRESS-BREAK command cancels any previous address break. Check your current address break with INFORMATION ADDRESS-BREAK.

Default subcommands - ALL, and

AFTER 1

```
        ---                        ---
        | date and hh:mm             |
        | hh:mm                      |
ALERT   | +hh:mm                     |            message
        | day-of-week and +hh:mm     |
        | TODAY and +hh:mm           |
        ---                        ---
```

causes the system to ring your terminal bell andtype a line at the specified date and time. This line contains the time of day and your message. The sign (+), used with the day-of-week and TODAY arguments, adds the time you specify to the beginning of the day (00:00:00 or midnight). For example, the command SET ALERT THURSDAY +10:00 sets an alert for Thursday at 10:00 A.M. If you omit the plus sign after a day-of-week or TODAY argument, the time is interpreted as part of the message. When a time argument is used without a day-of-week or TODAY argument, the plus sign adds the specified time to the current time. For example, the command SET ALERT +1:00 sets an alert for one hour from the time the command is given.

If the SET AUTOMATIC command is in effect, this message is sent no matter what you are doing at your terminal. Otherwise, you are alerted only when your terminal is about to type a TOPS-20 prompt ($ or @). Alert settings are erased when you log out. Therefore, you should enter this command in your COMAND.CMD file if you want to be alerted in the distant future or on a regular basis. Check the setting of this command with INFORMATION ALERTS. See Example 8.

AUTOMATIC — allows you to be notified by the system (as a result of a SET ALERT or SET MAIL-WATCH command) whether or not your job is at TOPS-20 command level. Every five minutes, the system checks to see if you should be notified. It is recommended that you enter this command

in the LOGIN.CMD file to ensure coverage from the time you log in. Check with INFORMATION ALERTS or INFORMATION COMMAND-LEVEL.

CARD-READER-INPUT-SET name of input set n
is used by the batch system to associate the indicated set of punch cards, beginning with deck n, with system device CDR:

CONTROL-C-CAPABILITY allows any program executed at the current command level to handle CTRL/C interrupts itself. You cannot use this command in a batch job. Check the current setting with INFORMATION PROGRAM-STATUS.

Default

```
         ---
         | CARDS /switch(es)
         | COMPILE-SWITCHES file-type /switch(es)
         | PAPER-TAPE /switch(es)
         | PLOT /switch(es)
         | PRINT /switch(es)
         |        ---
         |        | EPHEMERAL
DEFAULT  |PROGRAM | KEEP argument
         |        | NO-EPHEMERAL
         |        |NONE
         |        ---
         | SUBMIT /switch(es)
         |      ---
         |      | ALLOW
         |      | DISALLOW
         | TAKE | ECHO
         |      | NO ECHO
         ---    ---
```

sets up, as default global arguments to the command selected, the arguments you specify. CARDS refers to the PUNCH CARDS command, COMPILE- SWITCHES to all the LOAD-class commands, and PAPER-TAPE to the PUNCH PAPER TAPE command. These arguments are any switch or keyword valid for the given command.

For COMPILE-SWITCHES, you must specify the type of file you want the switches applied to by preceding the switches with one of the following: a file type (excluding the period), a period for file specifications with a null

extension, or an asterisk (*) for all file types. Check current settings with INFORMATION DEFAULTS.

DIRECTORY ACCOUNT-DEFAULT dev:<directory> default account PASSWORD:password

sets the account of 39 or fewer characters to charge for your terminal session whenever you log in to this directory without specifying an account. Check with INFORMATION DIRECTORY.

DIRECTORY ARCHIVE-ONLINE-EXPIRED-FILES dev:<directory>

causes on-line files that have expired to be automatically archived. Check with INFORMATION DIRECTORY.

DIRECTORY FILE-PROTECTION-DEFAULT dev:<directory> octal code PASSWORD:password

sets a default protection code governing access to files subsequently created in the directory. See description of FILE PROTECTION argument for a list of valid protection codes. Check with INFORMATION DIRECTORY.

Default code - 777700

DIRECTORY GENERATION-RETENTION-COUNT-DEFAULT dev:<directory> n PASSWORD:password

prescribes for the directory a default value for the number of generations of subsequently-created files to save. Check with INFORMATION DIRECTORY.

Default n - 1

DIRECTORY NO ARCHIVE-ONLINE-EXPIRED-FILES

prevents on-line files that have expired from being automatically archived

Default

| DIRECTORY NO SECURE str:<directory>

| Specifies that files created in the directory are not secure. The Access Control Job is not used to verify user access to new files in this directory.

| **Default**

DIRECTORY OFFLINE-EXPIRATION-DEFAULT dev:<directory> date or +n

sets the tape expiration date for files that are to go off line because of archiving or migration. If you specify

"+n", the expiration date is n days from the date the files were moved off line.
Default n - 90

DIRECTORY ONLINE-EXPIRATION-DEFAULT dev:<directory> date or +n
sets the disk expiration date for files that are to be created in the directory. If you specify "+n", the expiration date is n days from the creation date.
Default n - 60

DIRECTORY PASSWORD dev:<directory>
Old password:old password
New password:new password
Retype new password:new password
allows you to change the password of the directory named. The password can consist of up to 39 alphanumeric characters, including hyphens.

DIRECTORY PROTECTION dev:<directory> octal protection code
Password:password
establishes for the directory a protection code constructed (by addition) from the values shown below. Check with INFORMATION DIRECTORY.

77	full access to the directory
40	access to files in the directory (including expunging individual files), consistent with the file protection of the files
10	connect to the directory without giving a password, undelete files, expunge the entire directory, and change times, dates and accounting information for files. All other access is governed by the file protection of each file.
04	create files in the directory
00	no access to the directory

Default code - 777700

See the TOPS-20 User's Guide for more information about protection codes.

| DIRECTORY SECURE str:<directory>
| indicates that any new files created in the specified directory be made secure by default. When a file is secure, the Access Control Job checks to see if the user has access to that file before the user can read, write, append, rename, delete, set secure, or set unsecure that file.

ENTRY-VECTOR	octal or symbolic memory location	octal or symbolic length from 1 to 777

lets you change the entry vector of the program in memory. Check the current setting with INFORMATION MEMORY-USAGE.
Default length - 1

FILE ACCOUNT filespecs account
specifies the account to charge for storage of the files named. Check with the DIRECTORY command and the ACCOUNT subcommand.

FILE EPHEMERAL filespec
assigns an ephemeral attribute to a system program. The program is placed in an ephemeral fork only when you run it by typing just the program name as an EXEC command. Running an ephemeral system program with the R command cancels the ephemeral attribute. For a description of an ephemeral fork, see the ERUN command, Characteristics of an Ephemeral Fork.

Wheel or Operator privileges are required to set a system program ephemeral. See Example 11 for setting a system program ephemeral. See Example 12 for running your own programs by typing only the program name.

FILE EXPIRED filespecs
establishes today as the expiration date for the specified on-line files. Check with the DIRECTORY command and the DATES ONLINE-EXPIRATION subcommand.

FILE GENERATION-RETENTION-COUNT filespecs n
tells the system how many generations of the specified files to save. Check with the DIRECTORY command and the GENERATION-RETENTION-COUNT subcommand.
Default n - 1

FILE INVISIBLE filespecs
makes the specified file inaccessible to most programs and TOPS-20 commands. Check with the DIRECTORY command and the INVISIBLE subcommand.

FILE NO EPHEMERAL filespec
removes the ephemeral attribute from a

system program. Wheel or Operator privileges are required to alter a system program.

You can cancel the ephemeral attribute for your copy of a system program with the SET PROGRAM NO-EPHEMERAL command or, by running the program with the R command instead of simply typing the system program name as a command.

FILE NO PERMANENT filespecs — specifies that the file does not exist after it is deleted and expunged.
Default

FILE NO PROHIBIT filespecs — allows the system to migrate the specified file to off-line storage if disk space becomes low. For privileged users only. Check with the DIRECTORY command and the PROHIBIT-MIGRATION subcommand.
Default

FILE NO RESIST filespecs — cancels the effect of the SET FILE RESIST command. This switch allows the system to move the specified files to off-line storage without hesitating. Check with the DIRECTORY command and the RESIST-MIGRATION subcommand.
Default

FILE NO SAVE-BY-BACKUP SYSTEM filespecs
indicates not to save the specified file as indicated by a DUMPER save command. For example, a SYSTEM:DUMP.EXE file.

FILE NO SECURE filespecs — indicates that specified files are not secure. The Access Control Job is not used to verify user access to this file(s).

FILE NO TEMPORARY filespecs — indicates the specified file is not a temporary file.
Default

FILE NO UNDELETABLE filespecs
specifies that the file can be deleted.
Default

FILE OFFLINE-EXPIRATION filespecs date or +n
specifies when the contents of an

off-line file can be expunged from off-line storage. If you specify "+n", the expiration date is n days from the date it was moved off line. Check with the DIRECTORY command and the DATES OFFLINE-EXPIRATION subcommand.

FILE ONLINE-EXPIRATION filespecs date or +n
establishes the date on which the disk contents of the specified files will expire. If you specify "+n", the expiration date is n days from the current date. Check with the DIRECTORY command the DATES ONLINE-EXPIRATION subcommand.

FILE PERMANENT filespecs specifies that after a file is deleted and expunged, the file name still exists. For example, a MAIL.TXT file.

FILE PROHIBIT filespecs tells the system never to migrate the specified file to off-line storage. For privileged users only. (Nonprivileged users should refer to the description of the SET FILE RESIST command. See also Hints - Alternative to SET FILE PROHIBIT for Non-privileged Users, below.) Check with the DIRECTORY command and the PROHIBIT-MIGRATION subcommand.

FILE PROTECTION filespecs octal protection code
sets, for the specified files, a protection code constructed (by addition) from the octal values shown below. Check with the DIRECTORY command and the PROTECTION subcommand.

77	full access to the file
40	read the file
20	write and delete the file
10	execute the program contained in the file
04	append to the file
02	access the file using wildcarded file specifications
00	no access to the file

Default code - 777700

See the TOPS-20 User's Guide for more information about protection codes.

FILE RESIST filespecs offers nonprivileged users limited protection against migration. The

specified files will be forced off-line only when absolutely necessary. Check with the DIRECTORY command and the RESIST-MIGRATION subcommand.

| FILE SAVE-BY-BACKUP-SYSTEM filespecs
| indicates that the specified file is saved as required by a incremental or full DUMPER save command.
| **Default**
|
| FILE SECURE filespecs
| indicates that the specified files are secure. When a file is secure, the Access Control Job checks to see if the user has access to that file before the user can read, write, append, rename, delete, set secure, or set unsecure that file.
|
| FILE TEMPORARY filespecs
| indicates the specified file is temporary.
|
| FILE UNDELETABLE filespecs
| indicates the specified file cannot be deleted.

FILE VISIBLE filespecs
makes the specified file accessible to all programs and TOPS-20 commands. Check with the DIRECTORY command and the INVISIBLE subcommand.
Default

HOST
Due to the number of options in the SET HOST command, it is described separately from the SET command. See the SET HOST command description following the SET command.

LATE-CLEAR-TYPEAHEAD
instructs the system to disregard all terminal input made after a line that causes an error and before the next prompt. Check the setting for your current level of TOPS-20 with INFORMATION COMMAND-LEVEL.

LOCATION node-name::
causes all output device request to be sent to the specified IBM remote station. Two colons (::) following the node name are optional. Check available nodes with INFORMATION DECNET, and check your current setting (if different from your host node [log-in node]) with

INFORMATION JOB-STATUS.
Default node-name - your host node

MAIL-WATCH user-name message-count

checks the MAIL file for the specified user immediately and every five minutes thereafter whenever your terminal is about to type a TOPS-20 prompt (@ or $), and sends a message notifying you that the user has new mail if this file contains unread mail. If you specify your own user-name, you receive the message [You have mail from USER-NAME at 00:00:00]. If you specify a user-name other than your own, you receive the message [RECEIVER-USER-NAME has mail from SENDER-USER-NAME at 00:00:00]. You must have read access to the specified user's mail file.

The message count argument sets the number of times you are notified of unread mail. If the SET AUTOMATIC command is in effect, this message is sent no matter what you are doing at your terminal. The maximum number of users that you can MAIL-WATCH is five. See Example 10.
Default user-name - your user-name
Default message-count - 1000

NAME fork-name

renames the current fork with the specified alphanumeric name. Select the current fork with the FORK command. Check with INFORMATION FORK-STATUS.

If you give a fork the name of a program specified in a SET PROGRAM command, the fork will receive the attributes assigned in the SET PROGRAM command. For example, suppose you have given these two commands:

```
SET PROGRAM COMPUTE KEEP CONTINUE
SET PROGRAM SQUARE EPHEMERAL
```

If only COMPUTE is in memory and you name it SQUARE, the fork will assume the attributes defined in the SET PROGRAM SQUARE command and become an ephemeral fork. The system indicates this with

the message [Assuming attributes of SQUARE].

A fork must have a unique name. If you attempt to name a fork with the same name as another fork, the system appends a digit to the new name. For example, if you attempted to name two forks EDIT, the second fork would be named EDIT0.

NO ALERT date/time

removes settings that were established with the SET ALERT command. You can specify date and time in the same formats as with SET ALERT. Additionally, you can enter BEFORE or AFTER the date and time to indicate a time period in which alerts are to be suppressed. If you specify no date or time argument, all alert settings are erased. Alerts are valid only for the current terminal session and are erased automatically when you log out.

Default

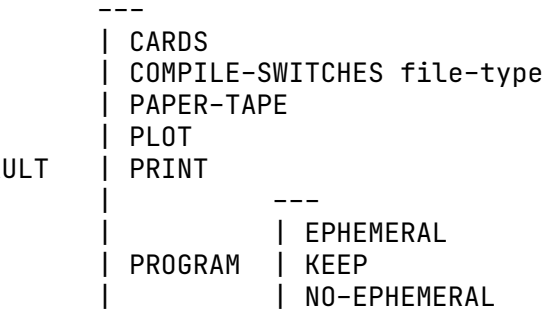
NO AUTOMATIC

causes you to be alerted by the system (as a result of a SET ALERT or SET MAIL-WATCH command) only when your terminal is about to type a TOPS-20 prompt (@ or $). Check with INFORMATION ALERTS or INFORMATION COMMAND-LEVEL.

Default

NO CONTROL-C-CAPABILITY

removes the ability of programs at the current level of TOPS-20 to prevent your terminal from returning to the TOPS-20 command processor whenever you type a CTRL/C; ensures that CTRL/C will return you to TOPS-20. Check the setting for your current level of TOPS-20 with INFORMATION PROGRAM-STATUS.



NONE	
SUBMIT	

nullifies all default arguments (established with a previous SET DEFAULT command) for the indicated command. For COMPILE-SWITCHES you must specify the type of file for which you want to clear the switches with one of the following: a file type (excluding the period), a period for file specifications with a null extension, or an * to clear the switches for all file types. Check with INFORMATION DEFAULTS.

NO LATE-CLEAR-TYPEAHEAD

instructs the system to accept terminal input made after an error message is sent to your terminal and before the next prompt. Check the setting for your current level of TOPS-20 with INFORMATION COMMAND-LEVEL.

Default

NO MAIL-WATCH user-name

disables periodic checking of the MAIL file associated with the specified user. The notice of new mail is still displayed at log-in time and when you receive mail, unless you have given the REFUSE SYSTEM-MESSAGES **or** REFUSE LINKS command. You can always check the status of your MAIL file at any time by giving the INFORMATION MAIL command.

Default user-name - your user-name

Default

NO RETRIEVAL-WAIT

tells the system to send an error message if your job attempts to use off-line files.

Default

NO STATUS-WATCH

cancels the effect of the SET STATUS-WATCH command, disabling the interrupt character or character sequence that displays the status of all open, mapped pages.

NO TIME-LIMIT

removes any time limit set by a previous SET TIME-LIMIT command. You cannot use this command in a batch job.

NO TRAP

prevents any trapping that would have occurred as the result of a SET TRAP

command.
Default

NO TRAP FILE-OPENINGS — nullifies the effects of the SET TRAP FILE-OPENINGS command, disabling the TOPS-20 feature that causes you to be notified when a program tries to open a file.
Default

NO TRAP JSYS [/ALL | /DEFINED | /UNDEFINED | name | number] — nullifies the effects of the SET TRAP JSYS command, disabling the TOPS-20 feature that causes traps to occur when a JSYS is executed.
Default

NO TRAP PROCEED — same as TRAP NO PROCEED.

NO UUO-SIMULATION — disables the feature of the TOPS-20 monitor that makes it possible to use programs originally written for the TOPS-10 operating system. Check the current setting with INFORMATION PROGRAM-STATUS.

PAGE-ACCESS range of octal page numbers type of access
Sets the type of access allowed to programs for the specified pages existing in memory.

COPY-ON-WRITE
provides programs with private copies of the specified pages (13:17, 21 specifies pages 13 through 17 and page 21, 6 pages in all) of current memory whenever they try to change (write to) them

EXECUTE allows programs accessing these pages to execute the instructions they may contain

NO [COPY-ON-WRITE | WRITE]
prevents programs from performing the indicated operation on the specified

pages

NONEXISTENT
removes the indicated pages from memory

READ permits programs to examine the indicated pages of memory

WRITE permits programs to change as well as examine the indicated pages

Check the status of current memory pages with INFORMATION MEMORY-USAGE.

| PASSWORD
| Old password:old password
| New password:new password
| Retype new password:new password
| allows you to change the password of the login directory PS:<username>. The password can consist of up to 39 alphanumeric characters, including hyphens. This command is identical to the SET DIRECTORY PASSWORD command, except that PS:<username> is the default directory for the SET PASSWORD command.

PROGRAM fork-name EPHEMERAL tells the system to make the specified fork an ephemeral fork when it is loaded. For a description of an ephemeral fork, see the ERUN command, Characteristics of an Ephemeral Fork.

It is recommended that you enter this command in your COMAND.CMD file for programs that you commonly place in ephemeral forks. Check with INFORMATION PROGRAM-STATUS.

PROGRAM fork-name KEEP

| CONTINUE
| REENTER
| START
---

tells the system to make the specified fork a kept fork automatically when the fork is loaded, or immediately if the fork is already loaded. A kept fork is not reset when another fork is loaded and is not reset by the RESET command unless the kept fork is explicitly named

or the asterisk (*), or period (.) argument is specified.

The required KEEP attribute establishes where the program restarts when you type the fork name as an EXEC command. CONTINUE begins the program at the point where it was interrupted. REENTER begins the program at its reentry address (for most programs the reentry address is the same as the start address). START begins the program at its start address. The system informs you when the fork is "kept" with the message [Keeping fork-name]. When you type the kept fork name, the system responds with [CONTINUING], [REENTERING], or [STARTING].

This command automatically keeps forks that are loaded by typing the system program name or one of the following commands: CSAVE, GET, R, RUN, and SAVE.

It is recommended that you enter this command in your COMAND.CMD file for programs that you commonly place in kept forks. Check with INFORMATION PROGRAM-STATUS. See Example 9. For a restriction on the CONTINUE argument, see the CONTINUE command description under Restrictions: Continued Programs Do Not Prompt for Input.

PROGRAM fork-name NO-EPHEMERAL

disables the ephemeral attribute for your copy of a system program. Note that you can also cancel a program's ephemeral attribute by running the program with the R command instead of typing just the program name. It is recommended that you enter this command in your COMAND.CMD file. See Example 13.

PROGRAM fork-name NONE

cancels the setting established for the specified fork with the SET PROGRAM command. If the program is in a kept fork, the address used when the fork name is given as a command is changed to the start address. Check with

```
                    INFORMATION PROGRAM-STATUS.
               CHARACTERISTIC name value
                              ---              ---  ---
REMOTE-PRINTING  PRINTER name | remote queue     |  | DQS node
                              | LATserver port   |  | LAT server
                              | LATserver service|  ---
                              | alias            |
                              ---              ---
               SYSTEM-DEFINITIONS
```

Lets you create a way to directly specify a queue and a characteristic parameter when submitting a remote print request. SET REMOTE-PRINTING commands can be invoked at command level or within a command file.

CHARACTERISTIC sets up a string to be used as input to the PRINT command /CHARACTERISTIC switch. The system file SYSTEM:REMOTE-PRINTING.CMD uses this command to establish the initial system setting, which equate each characteristic string to an integer value. You can rename a system-wide characteristic setting by re-issuing the command with a new name and value. You can undefine a characteristic by issuing the SET REMOTE-PRINTING CHARACTERISTIC command with a null value. Multiple characteristics are separated by commas.

The name argument can be a maximum of 14 characters per characteristic and must begin with an alphabetic character. The name can consist of any combination of the following:

- the letters of the alphabet
- the digits 0 through 9
- the symbols _(underscore) and $ (dollar sign)

To get information about the system characteristics settings, use the INFORMATION REMOTE-PRINTING command or read the SYSTEM:REMOTE-PRINTING.CMD file.

PRINTER sets up a string to be used as

input to the PRINT command /REMOTE-PRINTER switch. You can create keywords (aliases) that designate the names of remote printers and print queues or LATprinter ports and services. This allows you to use simple names for remote print designations when using the /REMOTE-PRINTER switch. You can define an alias for the following:

- an actual remote printer queue (such as XEROX defining SI$8700 on VAXNOD)
- a LATprinter port or service (such as LN03 defining LBBNA129 on LAT990)
- another alias (such as FAST defining XEROX).

The form of the command that includes the node or server name is used to define the actual printer or queue. After this is done, you can use the other forms of the command to apply aliases to the defined printer name.

SYSTEM-DEFINITIONS sets up remote printing information for a job based on the settings in the system-wide file SYSTEM:REMOTE-PRINTING.CMD. Invoking this command provides the complete set of printers and characteristics available for remote printing as defined by the system manager.

RETRIEVAL-WAIT — tells the system that your job is willing to wait for retrieval of off-line files. Retrieval is then requested implicitly whenever you or a program you run attempts to access off-line files. Use INFORMATION SYSTEM-STATUS to be sure that automatic retrieval waits are enabled for the system before giving this command.

SESSION-REMARK remark — lets you insert a note or reminder of up to 39 characters into system accounting data. Check with INFORMATION JOB-STATUS.

| IMMEDIATE SPOOLED-OUTPUT | DEFERRED ---	directs the system either to begin processing your spooled output requests as soon as you make them, or to defer them until log-out. You make spooled output requests not with the PLOT, PRINT, or PUNCH commands (these are always processed immediately), but with a command or program that writes files to a spooled output device (for example, a line printer - LPT:, plotter - PLT:, or card punch - CDP:). The COPY commands, the /LIST switch for LOAD class commands, the CREF command, and the LPT and OUTPUT subcommands for DIRECTORY-class and SYSTAT commands may make spooled output requests. Check with INFORMATION SPOOLED-OUTPUT-ACTION. **Default** - IMMEDIATE
STATUS-WATCH, @@INTERRUPT @@NO @@PAGES @@TOPS-10-PAGES	sets an interrupt character that displays the status of all open, non-execute files mapped to the current fork. The display includes: o the job file number. The JFN identifies a file to the job. The user program uses the JFN in all references to the file. o the file specification. o file pages mapped to a process. o file position and byte size displayed in the form "Byte nn(mm)"" where nn is the byte pointer and (mm) is the byte size. Not displayed if file position is zero. o file size displayed in the form "Page nn of mm" where nn is the page being read and mm is the total number of pages in the file. If a single number is displayed, as in "Page nn", nn is the total number of pages in the file. For example, "Page 5 of 9" represents a 9-page input file where page 5 is currently being read. "Page 11" represents an

output file where 11 pages have been created. Note that some programs do not read the input file sequentially.

- The mode of access (read, write and append) for which the file has been opened.

Although much of the above information is also provided by the INFORMATION FILE-STATUS and INFORMATION MEMORY commands, these commands can only display information while your terminal is at EXEC command level. The STATUS-WATCH interrupt character functions from EXEC or program level. In addition, the interrupt character displays the status of an executing EXEC command (for example, COPY).

To check the status of pages mapped to a program, you must specify one of these subcommands: PAGES, or TOPS-10-PAGES. The interrupt character always displays the pages opened by the EXEC, for example the pages opened by a COPY command. When an EXEC command is processing, only the EXEC's pages are checked. If no command is in progress, the current fork's address space is checked. If there is no current fork, no checking takes place.

For efficiency and to reduce the overhead of this command, a maximum of 512 pages (not including EXEC pages) are checked by the interrupt character. These pages do not have to be contiguous or in the same section.

To simplify your typing, SET STATUS-WATCH accepts subcommands as arguments on the command line.

INTERRUPT "^x" is a required subcommand that sets a control character or character sequence that, when typed during program or EXEC command execution, displays the status of all files opened by the current fork. Use

the FORK command to select a different current fork.

The interrupt character can be a single control character or a two-character sequence enclosed in double quotes. For example, "^B", "DF", "^QW" are valid interrupt characters. Control characters that are already used by the system, such as CTRL/C and CTRL/T, cannot be redefined by SET STATUS-WATCH. See Appendix D for a list of defined system control characters.

A two-character interrupt sequence is job-wide and can be issued from any EXEC level. If another program in your job defines a two-character interrupt sequence (for example the SET HOST command with the CTERM-SERVER), this sequence supersedes the SET STATUS-WATCH interrupt sequence. A single-character interrupt applies only to the current EXEC level. Also, the interrupt character is not echoed on the terminal.

NO INTERRUPT disables the interrupt characters.

PAGES nn adds the specified octal pages (nn) or range of pages (n:m) to the pages checked by the interrupt character. Multiple pages and ranges of pages can be specified by separating the pages with commas. For example, PAGE 100:40, 350 specifies pages 100 through 140 and page 350.

NO PAGES disables checking for all pages except pages used by the EXEC.

The total number of mapped pages checked by the interrupt character (not including EXEC pages) cannot exceed 512. Generally, programs map pages within the range of 0:777.

TOPS-10-PAGES adds the pages used by PA1050 (for TOPS-10 compatibility) to the address space checked by the interrupt character.

It is recommended that you place the SET STATUS command in your COMAND.CMD file.

See Example 14.

```
              ---
              |  200
              |  556
TAPE DENSITY  |  800
              | 1600
              | 6250
              | SYSTEM-DEFAULT
              ---
```

instructs the system to read and write magnetic tapes for your job at the indicated density (in bits per inch). SYSTEM-DEFAULT, one of these values (usually 1600), is established by your system manager. The value set by this command can be superseded by commands within a program. Check with INFORMATION TAPE-PARAMETERS.

Default - SYSTEM-DEFAULT

```
              ---
              | ANSI-ASCII
              | CORE-DUMP
              | HIGH-DENSITY
TAPE FORMAT   | INDUSTRY-COMPATIBLE
              | SIXBIT
              | SYSTEM-DEFAULT
              ---
```

advises the system that the format to use in processing tapes is either ANSI-ASCII, which stores each word of data as five 7-bit bytes in five frames of a 9-track type; or CORE-DUMP, which stores each word of data as a single 36-bit byte in five frames of a 9-track tape, partially using the fifth frame; or HIGH-DENSITY, which stores each two words of data as nine 8-bit bytes in nine frames of a 9-track tape; or INDUSTRY-COMPATIBLE, which stores each word of data as four 8-bit bytes in four frames of a 9-track tape; or SIXBIT, which stores each word of data as six 6-bit bytes in six frames of a 7-track tape. SYSTEM-DEFAULT, one of these (usually CORE-DUMP), is chosen by your

system manager. See also Restrictions - Using SET TAPE Commands, in the MOUNT command description in this manual. See the TOPS-20 Monitor Calls Reference Manual for more information about hardware data modes for magnetic tapes. Check with INFORMATION TAPE-PARAMETERS.

Default - SYSTEM-DEFAULT

TAPE PARITY --- | EVEN | ODD ---	tells the system which parity to assume when verifying the accuracy of tape records. Check with INFORMATION TAPE-PARAMETERS. **Default** - ODD
TAPE RECORD-LENGTH n	sets the size, in bytes, for each physical record on a tape. Check with INFORMATION TAPE-PARAMETERS. Not applicable with labeled tapes. **Default** n - 512
TERMINAL feature or type	same as TERMINAL command.
TIME-LIMIT n	tells the system to stop any program or terminal printout when the given amount of additional CPU time (in seconds) has been used, and to inform you with a fatal error message. This command is used by the batch system to limit the runtime of batch jobs. Display the time limit set for your job with the SYSTAT command and the LIMIT subcommand. Display the CPU time used by your job with CTRL/T or INFORMATION PROGRAM-STATUS.
TRAP FILE-OPENINGS	displays a message when any program attempts to open a file. Check with INFORMATION PROGRAM-STATUS. See Example 5.
TRAP JSYS --- | /ALL | /DEFINED | /UNDEFINED | name | number ---	displays a message when any program calls a TOPS-20 JSYS. You can cause trapping to occur for all JSYSs, for

defined JSYSs only (JSYSs known to the Monitor), for undefined JSYSs only (JSYSs not known to the Monitor), or for the JSYS(s) specified by name or number. You can specify multiple JSYSs separated by commas. Check with INFORMATION PROGRAM-STATUS. See Example 6.

Default - /DEFINED

NOTE

The SET TRAP command is ineffective for execute-only programs (those with a protection code that prohibits reading and writing the file). Attempts to run such programs after a SET TRAP command has been specified will result in error messages.

TRAP NO — same as SET NO TRAP.

TRAP NO FILE-OPENINGS — same as SET NO TRAP FILE-OPENINGS.

TRAP NO JSYS | /ALL | /DEFINED | /UNDEFINED | name | number — same as SET NO TRAP JSYS.

TRAP NO PROCEED — directs the system to terminate the program after a trap has occurred as a result of a SET TRAP command. Check with INFORMATION PROGRAM-STATUS. See Example 7.

TRAP PROCEED — directs the system to continue a program after a trap has occurred as a result of a SET TRAP command. Check with INFORMATION PROGRAM-STATUS.

Default

TYPEOUT MODE | NUMERIC | SYMBOLIC — establishes the mode in which memory addresses and contents are to be typed on your terminal in response, for example, to a CTRL/T or the commands:

INFORMATION ADDRESS-BREAK, INFORMATION FORK-STATUS, INFORMATION PROGRAM-STATUS, and EXAMINE. Note that only NUMERIC typeout is displayed for execute-only programs. Check with INFORMATION PROGRAM-STATUS.
Default - NUMERIC

UUO-SIMULATION — allows the system to execute programs originally written for the TOPS-10 operating system, by calling the TOPS-10 compatibility package, PA1050.EXE. Check the current setting with INFORMATION PROGRAM-STATUS.
Default

Characteristics

Affect Only Current Terminal Session

The SET command, except for SET DIRECTORY and SET FILE, applies to the current terminal session only, and in most cases only to the current level of TOPS-20 in that session. Therefore put SET DEFAULT, SET CONTROL-C-CAPABILITY, SET PROGRAM, and other SET commands into your COMAND.CMD file if you want them to be in effect every time you log in or give the PUSH command. Place commands that apply to any level of TOPS-20 in your LOGIN.CMD file.

SYSTEM:REMOTE-PRINTING.CMD

The system-wide file REMOTE-PRINTING.CMD contains SET REMOTE-PRINTING commands to establish printer aliases and characteristics values. The SET REMOTE-PRINTING SYSTEM-DEFINITIONS command sets up remote printing information for a job based on the settings in REMOTE-PRINTING.CMD. Internal tables are built that consist of the command arguments. These tables are used to validate the /CHARACTERISTIC and /REMOTE-PRINTER switch values specified by the user.

You can invoke the SET REMOTE-PRINTING SYSTEM-DEFINITONS command at command level or within a command file.

Hints

Using SET PAGE-ACCESS

A SET PAGE-ACCESS command can take several arguments on the

same line, with cumulative effect; contradictions are resolved in favor of the last item given. So SET PAGE-ACCESS 6 EXECUTE NO COPY-ON-WRITE NO WRITE allows a user to execute page 6 but not to change it; SET PAGE-ACCESS 7 NO WRITE WRITE allows changes to page 7.

To Make Modifiable Copies of Write-protected Programs

Because the SAVE command preserves the write protection of files, you should use the SET PAGE ACCESS WRITE or SET PAGE-ACCESS COPY-ON-WRITE command before giving SAVE if you want to save a modifiable copy of a program.

Using SET TIME-LIMIT

Although the SET TIME-LIMIT command is ordinarily used by the batch system to limit the runtime of jobs, you can employ it as a timesharing user to give you a fatal error message when the specified amount of CPU time has been spent. To find out how much of this time you have left, give the SYSTAT . LIMIT and INFORMATION PROGRAM-STATUS commands. The difference between the SYSTAT . LIMIT time and the "Used" time reported by INFORMATION PROGRAM-STATUS tells you the approximate time remaining.

SET Commands Useful for Debugging Programs

SET ADDRESS-BREAK

SET ADDRESS-BREAK shows you how often and for what purpose a memory address is referenced. When an address break occurs, a message will show the memory location at which execution of your program will resume.

SET NO CONTROL-C-CAPABILITY, SET UUO-SIMULATION, SET PAGE-ACCESS

If you are debugging a program, use the SET NO CONTROL-C-CAPABILITY command to ensure that you can use CTRL/C to leave the program. Test a program that traps CTRL/Cs by having it trap, say, CTRL/As instead during debugging. Also, setting NO CONTROL-C-CAPABILITY, NO UUO-SIMULATION, or PAGE-ACCESS NO WRITE NO COPY-ON-WRITE will show you what part of the program (if any) is attempting to use these features.

Alternative to SET FILE PROHIBIT for Non-privileged Users

Even if you do not have sufficient privileges to use the SET FILE PROHIBIT command, you can still do something to delay the removal of important files to off-line storage.

Create a file named MIGRATION.ORDER in each directory for which you wish to control migration. The contents of this file should be the specifications of files that you want to be migrated first, when migration is performed. You may use wildcard characters (* and %) to specify more than one file. To protect source programs, for example, you could specify that executable programs and binary files be migrated first, by listing "*.EXE, *.REL" in MIGRATION.ORDER. To protect edited files, you could list "*.Q*" (this ensures that unedited back-up files produced by the EDIT program be migrated before the edited versions).

Any files not listed in MIGRATION.ORDER will be protected from migration until all listed files have been migrated. Remember that, even without being listed in MIGRATION.ORDER, files are not usually migrated to off-line storage if they have been used or changed within a period of time specified by your system manager.

The SET FILE RESIST command also offers limited protection against involuntary migration.

Using SET REMOTE-PRINTING PRINTER

It is possible that two remote queue names on different clusters may be the same or that a remote queue name may be the same as a LAT port or service name. The target node name or actual name form of the SET REMOTE-PRINTING PRINTER command resolves this problem. The node name or server name distinguishes one identically named printer from another. For example, the following two commands would help avoid confusion in such an instance:

```
SET REMOTE-PRINTING PRINTER ODIE LASER LAT1
SET REMOTE-PRINTING PRINTER GARFIELD LASER VAXNOD
```

Restrictions

Using SET Commands in Batch Jobs

Put SET commands into a BATCH.CMD file in your log-in directory if you want them to apply to the first (highest) level of TOPS-20 in batch jobs you submit; put them into COMAND.CMD in your log-in directory if you want them to apply to all levels of TOPS-20 in both batch and interactive jobs. Remember, though, that you must not give SET

CONTROL-C-CAPABILITY, SET NO TIME-LIMIT, or SET TIME-LIMIT (or the ATTACH command) within any batch job.

Using SET DIRECTORY Commands

You will be able to use the SET DIRECTORY commands only if your system is instructed at system start-up time to allow them. Otherwise, the system will send you error messages in response to SET DIRECTORY commands.

Using SET REMOTE-PRINTING Commands

In supporting host initiated connections to LATprinters, TOPS-20 users are limited to six character server names.

The remote printer functionality on TOPS-20 does not include features to allow remote systems to access a printer facility local to a TOPS-20 system.

Examples

1. Set the LATE-CLEAR-TYPEAHEAD parameter for your job.

```
@SET LATE-CLEAR-TYPEAHEAD
```

2. Find out the placement of your program in memory; set an address break to occur at location 2412 when the instruction it contains has been executed six times. Then give the INFORMATION ADDRESS-BREAK command to see the location and operation for which the current address break has been set.

```
@INFORMATION MEMORY-USAGE

 5. pages, Entry vector loc 400010 len 254000

0-3           Private   R, W, E
400           Private   R, W, E

@SET ADDRESS-BREAK 2412,
@@AFTER 6
@@EXECUTE
@@
@INFORMATION ADDRESS-BREAK
Address break at 2412 on execute.
```

3. Set defaults for PRINT command switches, then print a file immediately by explicitly supplying an /AFTER switch with an early hour as argument.

```
@SET DEFAULT PRINT /LOWERCASE/AFTER:17:00
```

```
@PRINT /AFTER:+0 4-UPED.TXT
[Job 4-UPED Queued, Request-ID 346, Limit 200]
@INFORMATION OUTPUT-REQUESTS /USER
Printer Queue:
Job Name  Req#  Limit            User
--------  ----  -----  ------------------------
* 4-UPED   346    200  LATTA                        On Unit:0
  Started at 16:11:11, printed 0 of 200 pages

There is 1 Job in the Queue (1 in Progress)
```

4. Put an executable program into memory and set the page access of its first page to NO COPY-ON-WRITE; try to deposit a value (32) in memory location 500 of the page (this fails). Then set its page access to COPY-ON-WRITE and try once more, succeeding this time. Give the INFORMATION MEMORY-USAGE command again. Notice that you now have your own copy of the page in memory; it is no longer mapped from the file TESTF1.EXE in your connected directory.

```
@GET TESTF1
@INFORMATION MEMORY-USAGE

 1. pages, Entry vector loc 145 len 254000

0             TESTF1.EXE.3  1   R, CW, E

@SET PAGE-ACCESS 0 NO COPY-ON-WRITE
@DEPOSIT 500 32
?Can't write that page
@SET PAGE-ACCESS 0 COPY-ON-WRITE
@DEPOSIT 500 32
 [Shared]
@INFORMATION MEMORY-USAGE

 1. pages, Entry vector loc 145 len 254000

0             Private   R, W, E
```

5. Learn what files are opened when you edit a file.

```
@SET TRAP FILE-OPENINGS
@EDIT LOGIN.CMD
[Fork EDIT opening SWITCH.INI.3 for reading]
[Fork EDIT opening LOGIN.CMD.33 for reading]
Edit: LOGIN.CMD.33
[Fork EDIT opening EDIT-BUFFER.OUT.100042 for writing]
*EU
[Fork EDIT opening EDIT-BUFFER.OUT.100042 for reading]
[Fork EDIT opening LOGIN.CMD.34 for writing]
```

```
[LOGIN.CMD.34]
```

6. Cause a trap to occur whenever the GTFDB JSYS is executed. Then edit a file. The EDIT command invokes the GTFDB JSYS and causes a line to type out in the following format:

```
[fork "trap" <location>/<jsys name> "Ac's 1-4:" -
<ac contents>]
```

Note that the location is in symbolic form if you so specified in the SET TYPEOUT MODE command.

```
@SET TRAP JSYS GTFDB
@EDIT LOGIN.CMD
[EDIT trap 3515/ GTFDB Ac's 1-4: 11 1000004 20321 424153000000]
[EDIT trap 3562/ GTFDB Ac's 1-4: 11 2000011 4 424153000000]
Edit: LOGIN.CMD.42
*EU

[LOGIN.CMD.43]
```

7. Specify that program execution is to halt whenever a GTFDB JSYS causes a trap. Then edit a file. The EDIT command invokes the GTFDB JSYS, causing a trap to occur, which causes the EDIT process to immediately halt.

```
@SET TRAP NO PROCEED
@SET TRAP JSYS GTFDB
@EDIT LOGIN.CMD
[EDIT trap 3515/ GTFDB Ac's 1-4: 10 1000004 20321 424153000000]
```

8. Arrange for the system to remind you of a future obligation. Then verify that you will be reminded.

```
@SET ALERT MONDAY +11:00:00 Turn in time card
@SET AUTOMATIC
@INFORMATION ALERTS
 Next alert at 8-Jun-84 16:55:00 - Almost time to go home!!
 Other alerts set for:
   11-Jun-84 08:55:00 - Project meeting at 9:00
   14-Jun-84 11:00:00 - Turn in last week's time card by noon

  Alerts are automatic
```

9. Set the CHANGE and RADIUS programs to be automatically placed in kept forks when they are run. Then give the INFORMATION PROGRAM-STATUS command to display all the SET PROGRAM settings. Finally, run the CHANGE program. Note that the message [Keeping CHANGE] indicates that the program is being loaded into a kept fork.

```
@SET PROGRAM RADIUS KEEP CONTINUE
@SET PROGRAM CHANGE KEEP CONTINUE
@INFORMATION (ABOUT) PROGRAM
 Used 0:00:35 in 0:24:09
 TOPS-20: 0:00:00.8
 SET UUO-SIMULATION (FOR PROGRAM)
 SET TYPEOUT MODE NUMERIC
 SET PROGRAM RADIUS KEEP (AND) CONTINUE (WHEN INVOKED AS A COMMAND)
 SET PROGRAM CHANGE KEEP (AND) CONTINUE (WHEN INVOKED AS A COMMAND)
 SET PROGRAM MS KEEP (AND) START (WHEN INVOKED AS A COMMAND)
 SET PROGRAM DSRPLUS KEEP (AND) START (WHEN INVOKED AS A COMMAND)
 => MS (1): Kept, C from IO wait at 104062, 0:00:01.6
@RUN CAN
[Keeping CAN]
CAN>
```

10. Arrange for the system to check for new mail in your MAIL file and the MAIL file of user AI.GROUP. Notice the two messages indicating that you and AI.GROUP have new mail. Then, cancel mail watching for user AI.GROUP.

```
@SET MAIL-WATCH
@SET MAIL-WATCH AI.GROUP
[You have mail from SMITH at 10:12:11]
[AI.GROUP has mail from NELSON at 10:12:14]
@SET NO MAIL-WATCH AI.GROUP
```

11. Use the DIRECTORY command to learn the name of the directory that contains a system program. Then enable your Wheel or Operator privileges and set the system file ephemeral.

```
@DIRECTORY SYS:ISPELL

   RANDOM:<UNSUPPORTED>
 ISPELL.EXE.1
@ENABLE
$SET FILE EPHEMERAL RANDOM:<UNSUPPORTED>ISPELL.EXE
 RANDOM:<UNSUPPORTED>ISPELL.EXE.1 [OK]
$DISABLE
@
```

12. Add one of your own directories to the definition of SYS: so that you can run programs in that directory by typing just the program name.

```
@DEFINE SYS: => SYS:, STUDENTS:<DBONIN.TOOLS>
```

Next, set ephemeral a file in this directory. Run the program in an ephemeral fork by typing the program name. Then CTRL/C from the program. Give the INFORMATION FORK-STATUS command and note that the ephemeral fork CHANGE

has been reset.

```
@SET FILE EPHEMERAL CHANGE.EXE
@CHANGE
CHANGE>^C
@INFORMATION FORK-STATUS
=> EDIT (1): HALT at 6253, 0:00:00.6
```

13. Run an ephemeral system program and disable the program's ephemeral attribute.

```
@SET PROGRAM CHANGE NO-EPHEMERAL
@CHANGE
CHANGE>
```

or

```
@R CHANGE
CHANGE>
```

14. Give the SET STATUS-WATCH command with the INTERRUPT subcommand to specify CTRL/B as the interrupt character. Then give the PAGES subcommand to specify the range of pages to be checked by the interrupt character. Display both settings with the LIST-PARAMETERS subcommand:

```
@SET STATUS-WATCH,
@@INTERRUPT "^B"
@@PAGES 0:777
@@LIST-PARAMETERS
Enabled on "^B",  Checking pages: 1-512
@
```

Next run the DSRPLUS program and check its status by typing CTRL/B. This program reads the input file MEMO.RNO and creates the output file MEMO.MEM. Note that the ^B is not displayed on the terminal.

```
@DSRPLUS
DSRPLUS>MEMO.RNO

^B
Connected to BLAZE:<ROBBERTS>
6   PUBLIC:MEMO.RNO.1  [Page 1 of 9. Byte 128(36). Read]

^B
Connected to BLAZE:<ROBBERTS>
7   MEMO.MEM.1  [Page 3. Byte 512(36). Read Write]
6   PUBLIC:MEMO.RNO.1  [Page 4 of 9. Byte 1280(36). Read]

^B
```

```
Connected to BLAZE:<ROBBERTS>
7   MEMO.MEM.1  [Page 8. Byte 3840(36). Read Write]
6   PUBLIC:MEMO.RNO.1  [Page 8 of 9. Byte 3968(36). Read]

DSRPLUS>
```

Now look at the above display. The first time CTRL/B is typed, DSRPLUS is reading the first page of the nine page input file MEMO.RNO. The second CTRL/B shows that DSRPLUS is reading page four of the input file and has created three pages of the output file MEMO.MEM. Nearing its completion, DSRPLUS has read eight of the nine input pages and has created an eight page output file.

15. Define the name of the remote printer queue SI$8700 on node OURVAX to XEROX.

```
@SET REMOTE-PRINTING PRINTER XEROX SI$8700 OURVAX
@
```

Now, assign the alias FAST for the name of the same remote printer queue from XEROX.

```
@SET REMOTE-PRINTING PRINTER FAST XEROX
@
```

16. Define the name of a LATprinter connected to port LBBNA1297Y10X on a server named LAT990 to the alias LN03.

```
@SET REMOTE-PRINTING PRINTER LN03 LBBNA1297Y10X LAT990
@
```

Now, direct a print request to the LATprinter:

```
@PRINT MYFILE.MEM/REMOTE-PRINTER:LN03
```

17. Set some remote printer characteristics.

```
@SET REMOTE-PRINTING CHARACTERISTIC P90 52 ;portrait 90 wide
@SET REMOTE-PRINTING CHARACTERISTIC BOLD 61
@
```

2.72 SET HOST

Function

Connects your terminal to another system.

Format 1

```
@SET HOST node-name::  /switch
```

Format 2

(Omitting the node name on the command line allows you to define an interrupt sequence)

```
@SET HOST /switch
Two character interrupt sequence (^\,<RET>): interrupt-sequence
Node name: node-name::
```

where:

node-name::	is the name of the remote host that you want to connect your terminal to. Two colons (::) following the node name are optional.
/switch	is an optional keyword that selects the service used to connect your terminal to a remote node. Default - /CTERM
interrupt-sequence	is the characters that switch control of the terminal back to the local host. You are prompted for an interrupt sequence only if you give the /CTERM switch. Default - CTRL\,<RET>

NOTE

This command description assumes you are connecting to another TOPS-20 system. For information on accessing other operating systems, see the DIGITAL Networking Pocket Guide. This command description also assumes that you are establishing a connection with the CTERM program.

SET HOST Command Switches

/CTERM — connects your terminal to the remote node by running the CTERM communications program. Both processors must be running DECnet Phase 4 software (available under a separate license). The SET HOST command runs the CTERM communications program by default. If the remote node does not support the CTERM protocol, SET HOST attempts the connection again. In the second attempt however, SET HOST runs the communications program defined by the logical name NRT: (Network Remote Terminal).

Default

/NRT — runs the communications service program defined by the logical name NRT: (Network Remote Terminal). NRT: can be a system or job logical name. When both exist, the job definition takes precedence.

Characteristics

Logging into the Remote Node

Once your terminal is connected to the remote node, the system responds by identifying itself and prompting you to log in. You can then log in to the system.

Path of Terminal Input and Output

The SET HOST command passes terminal input through the local host to the remote host. Output from the remote host passes through the local host to your terminal.

Making a Series of Host Connections

Once you have logged in to a remote node, you can give EXEC commands and run programs just as you would on your local node. You can then establish a connection to another remote node. For example, if your local host is AURORA, you can give the command SET HOST BOSTON to connect to the node BOSTON; after logging in to BOSTON, you can use the command SET HOST DENVER to connect to node DENVER.

Returning to Your Local Host

To return your terminal to your local host, type the interrupt sequence to temporarily break the connection to the remote host. Note that the connection to the remote

host remains intact until you reset the CTERM program. You can reconnect your terminal to the remote host by giving the CONTINUE command.

When you log out of the remote host, the connection is broken and you are returned to your local host.

If you have established a series of connections, the interrupt character defined in your first SET HOST command returns you to your local host. For example, your local host is AURORA and you SET HOST to BOSTON, specifying the CTRL\<RET> interrupt sequence to return to AURORA. Then from BOSTON you connect to DENVER. Typing the CTRL\<RET> interrupt to DENVER returns you to AURORA, not BOSTON.

Specifying the Interrupt Sequence

- The interrupt sequence can be a combination of two characters or control characters. For example, KL ^KL, K^L, and ^K^L are valid interrupt sequences.

- If you specify only one interrupt character, the second character is ^J. For example if you specify only ^N as the interrupt sequence, the actual interrupt sequence is ^N^J. If you type only F, the interrupt sequence is F^J.

- Each interrupt character must be different. For example, ^K^K is invalid, while ^KK is a valid interrupt sequence.

- Do not type a comma or a space between interrupt characters as it will be interpreted as the second character in the sequence.

- The RETURN key can only be used in the default interrupt sequence, ^\<RET>.

- The interrupt sequence cannot contain predefined TOPS-20 control characters. For example, if you attempt to specify an interrupt sequence as ^T^I, the ^T will print the run status and the ^I will print a tab. See Appendix D in this manual for a list of TOPS-20 control characters.

Controlling Scrolling on a Remote Host

On your local host, <CTRL/S> and <CTRL/Q> are the default control characters that pause and continue scrolling. Typing <CTRL/Q> continues scrolling whether scrolling paused because you typed <CTRL/S> or the output paused on an

end-of-page.

However, CTERM does not pass these characters to the remote host. When using <CTRL/S> and <CTRL/Q>, it is the local host that actually controls scrolling. Therefore if output from the remote host has paused on an end-of-page, <CTRL/Q> will not continue scrolling. Other pause and continue characters are passed to the remote host. You can use <CTRL/A> to both pause and resume scrolling or you can use the TERMINAL PAUSE command to assign any two characters of your choosing to control scrolling. For consistency, it is recommended that you define the same pause and continue characters on your local and remote node.

Hints

Listing Available Nodes

Use the INFORMATION DECNET command to display the names of DECnet nodes accessible to your node.

Effect on Terminal

The SET HOST command connects your terminal to the remote system. After the remote system's herald message is printed, you can log in.

Related Commands

CONTINUE	for resuming a connection that was broken with an escape sequence.
INFORMATION DECNET	for displaying the names of nodes reachable from your node.
INFORMATION JOB-STATUS	for displaying the name of the host node and other information about your job.
INFORMATION LOGICAL-NAMES NRT:	for displaying the name of the communications program run by /NRT.
SYSTAT	for displaying (in the ORIGIN column) the name of the local system (the system you connected to before connecting to the current remote system).

Examples

1. Connect your terminal to a remote TOPS-20 node named AURORA and then login.

```
@SET HOST AURORA

[Attempting a connection, connect OK, ]
[Remote host is a TOPS-20 system]
[TYPE ^\,<RET> to return to node ROMAX]

 AURORA - Claims Tracking System, TOPS-20 Monitor 7(21002)
@LOGIN RSMITH
 Job 4 on TTY315 15-Nov-87 09:35:03, Last Login 15-Nov-87 08:18:48
```

2. Give the INFORMATION DECNET command to find out if node BOSTON is reachable from your host node. Then give the SET HOST command without typing the node name on the command line. Omitting the node name causes the system to prompt you for an interrupt sequence. After typing your own interrupt sequence, the system prompts you for the name of the remote node.

```
@INFORMATION DECNET BOSTON
 Node BOSTON is reachable
@SET HOST
Two character interrupt sequence (^\,<RET>): ^ED
Node name: BOSTON

[Attempting a connection, connect OK, ]
[Remote host is a TOPS-20 system]
[TYPE ^E,D to return to node AURORA]

 BOSTON, AI Engineering Center, TOPS-20 Monitor 7(21002)
@
```

3. Connect to a remote node named TEAL. After logging in and doing some work on TEAL, type the ^\<RET> interrupt sequence to return to your local host.

```
@SET HOST TEAL

[Attempting a connection, connect OK, ]
[Remote host is a TOPS-20 system]
[TYPE ^\,<RET> to return to node FLYWAY]

 TEAL - Migratory Bird Banding, TOPS-20 Monitor 7(21002)
@LOGIN LOWELL
 Job 4 on TTY315 15-Nov-87 09:35:03, Last Login 15-Nov-87 08:18:48
    .
    .
```

^\<RET> (Interrupt sequence not displayed on terminal)
[Connection interrupted, back at node FLYWAY,
Type CONTINUE to resume connection]

After working on node FLYWAY, type the CONTINUE command to reconnect your terminal to node TEAL. Then give the INFORMATION JOB-STATUS command to verify that you are connected to TEAL.

@CONTINUE

@INFORMATION JOB-STATUS
 Host TEAL
 Job 17, TTY4, User LOWELL, REPORTS:<LOWELL>
 Account 341

4. Attempt a connection using the CTERM program. The system attempts the connection and finds that the remote node does not support CTERM. It then attempts another connection using the program defined by NRT:

@SET HOST ROMAX

[Attempting a connection, Connect failed -

 Host did not accept CTERM connection, trying NRT:

[Attempting a connection, connect OK]
[Remote host is a TOPS-20 system]
[TYPE ^P to return to node AURORA]

 ROMAX - Acme's Timesharing System, TOPS-20 Monitor 7(21002)
@

2.73 SKIP

Moves a magnetic tape set forward over a specified number of files or records, or to the logical end of the tape set.

Format

@SKIP (DEVICE) **dev: n units**

where:

dev: is the name of the tape set or magnetic tape drive that you want to move forward.

n is the number of files or records over which you want to skip. The colon after the device name is optional.

units is either FILES or RECORDS, where records are sections of a file; or LEOT, to skip to the logical end of the tape set, which is the next point on the tape set having two adjacent EOF (end-of-file) marks.
Default units - FILES

Restrictions

SKIP With Open Files

If you have given a CTRL/C to exit from a program that has opened a file in a magnetic tape set and you then give the SKIP command for that tape set, the system will first ask if you want to close the associated file. You must do so for SKIP to succeed, but you will probably be unable to continue the program from that point because the file will now be closed.

RECORDS Argument Used for Unlabeled Tapes Only

You cannot use the RECORDS argument to the SKIP command when using a labeled tape, because read and write operations for labeled tapes always move the tape to the beginning of a file.

Warning

Skipping Past LEOT (Unlabeled Tapes Only)

If you specify too large a value for n in the SKIP command line, you can move past the logical end of tape (LEOT). In

this case, the operator may have to intervene before your tape control commands will have effect again. You must be sure how many files you have in the tape set if you use SKIP n rather than SKIP LEOT. This problem can occur for any tapes mounted on drives of the form MTAn:, or for unlabeled tapes mounted on drives of the form MTn:.

Related Commands

BACKSPACE for moving a magnetic tape backward a specified number of files or records

REWIND for returning a magnetic tape to its load point

UNLOAD for rewinding a magnetic tape completely onto the source reel (only for tapes mounted on drives having device names of the form MTAn:)

Examples

1. Skip over the next 2 files on the magnetic tape you are using (mounted on magnetic tape drive MT0: in this case).

   @SKIP MT0: 2 FILES

2. Skip over the next two records on an unlabeled tape.

   @SKIP MTA0: 2 RECORDS

3. Use the MOUNT command to ask the operator to mount your tape in write-enabled mode, then copy 3 files to the tape from your directory on structure SNARK:. Use the REWIND command to go back to the beginning, and the SKIP command to skip over the first file. Use the COPY command to have the next file (FIL-2) printed on your terminal, then give the SKIP command again to skip to the logical end-of-tape. You are skipping only one file, FIL-3, in this case.

   ```
   @MOUNT TAPE DAY:/WRITE-ENABLED
   [Mount Request DAY Queued, Request-ID 187]
   [Tape set DAY, Volume DAY mounted]
   [DAY: defined as MT0:]
   @REWIND DAY:
   @MOUNT STRUCTURE SNARK:
   Structure SNARK: mounted
   @ACCESS SNARK:
   @COPY SNARK:FIL-1.TAP DAY:
    SNARK:FIL-1.TAP.1 => MT0:FIL-1 [OK]
   @COPY SNARK:FIL-2.TAP DAY:
   ```

```
 SNARK:FIL-2.TAP.1 => MT0:FIL-2 [OK]
@COPY SNARK:FIL-3.TAP DAY:
 SNARK:FIL-3.TAP.1 => MT0:FIL-3 [OK]
@REWIND DAY:
@SKIP DAY: 1
@COPY DAY: TTY:
 T0: => TTY:

!THIS IS THE SECOND FILE.!
@SKIP DAY:LEOT
```

2.74 START

Begins execution of the program in the current fork.

Format

@**START** (PROGRAM) **location/switch**

where:

location — is the octal or symbolic address where you want the program to start.
Default location – the normal starting address, that is, the first word in the program's entry vector

switch — is a keyword, chosen from the list below, indicating your choice of START command options.

START Command Switches

/BACKGROUND — keeps your terminal at TOPS-20 command level and starts execution of the program in a "background" fork. When the program attempts to do terminal input or output, it halts and displays the message [FORK-NAME wants the TTY].

/NORMALLY — restores your terminal to command level (if any) within the program
Default

/STAY — keeps your terminal at TOPS-20 command level. Output from the program is sent to the terminal and is intermixed with whatever output is currently displayed. When the program attempts to read from the terminal, it can randomly intercept input intended for the EXEC or another program. Therefore, use this switch with programs that, once started, do not request further terminal input.

Characteristics

Starting a Noncurrent Fork

When you START a noncurrent fork, the fork becomes your current fork.

Hints

Further Information

The START command is one of the TOPS-20 multiforking-class commands. For more information about multiforking, see the section named Running Multiple Programs in the TOPS-20 User's Guide.

For more information about entry vectors, see the TOPS-20 Monitor Calls Reference Manual.

Special Cases

Running COBOL Programs a Second Time

After running a program (with a RUN or EXECUTE command, or with a GET and START or LOAD and START combination) you can usually run it again using START. COBOL programs are an exception: to run them again you must reload them.

Restrictions

Programs Competing for Terminal Input

If you use START /STAY to run a program in a background fork, either at the current or at a lower EXEC command level, (see Hints - Using PUSH to Get a New TOPS-20 Command Level, above), the program can request input from the terminal while you are giving input to the EXEC or another program. This input can be randomly intercepted by the background program when it requests terminal input. Usually though, the EXEC or the current program receives the input.

When terminal input is intercepted by the background program, the program usually types input error messages. To give input to the program, stop the program by typing two CTRL/Cs or the program's exit command. Then, if the background program is at a higher EXEC command level, give POP commands to return to the EXEC level that holds the background program. (POP terminates the current EXEC and erases programs in its memory.) Finally, give the CONTINUE /NORMALLY command; this puts you at program command level so that you can give the requested input.

Input is intercepted by the background program randomly. Therefore, you may have to type extra CTRL/Cs, program exit commands, and POPs. To reduce confusion about the direction of terminal input, it is recommended that you use START /STAY only when you plan to work at the current EXEC level

while a program runs in a background fork. Use START /BACKGROUND when you plan to work at a lower EXEC level or at another program command level.

When a program started with START BACKGROUND requests terminal input, it sends the message, [FORK-NAME wants the TTY]. No input is taken by the background program until you return to program command level with CONTINUE /NORMALLY.

No I/O Control with Some Programs

Most programs read and write data to the terminal through standard input and output designators. Some programs however, use different methods of communicating with the terminal. Therefore, when you use /BACKGROUND and /STAY to control terminal input and output from a background fork, the input and output behavior of programs with nonstandard designators can be unpredictable.

Execute-only Programs

Programs that are execute-only can only be started at their normal starting address.

Related Commands

CONTINUE	for resuming execution of a halted program in memory
FORK	for changing the current fork
FREEZE	for halting a program in a background fork
GET	for placing executable programs in memory
INFORMATION FORK-STATUS	for displaying the number and the status of each fork in your job
KEEP	for giving a fork a kept status
LOAD	for loading source or object programs into memory
REENTER	for starting the program in memory at its alternate entry point (if any)
SAVE	for saving a loaded program in an .EXE file

ERUN, RESET, SET NAME, SET PROGRAM, UNKEEP	other multiforking-class commands for performing related functions

Examples

1. Start the program currently in memory.

```
@START
```

2. Put an executable program in memory and start it. Then run it again.

```
@GET TESTF1.EXE
@START

THIS IS A TEST.

END OF EXECUTION
CPU TIME: 0.04  ELAPSED TIME: 0.23
EXIT
@START

THIS IS A TEST.

END OF EXECUTION
CPU TIME: 0.02  ELAPSED TIME: 0.02
EXIT
```

3. Begin using the FILCOM program to compare two files. Give a CTRL/C to halt FILCOM, then a CTRL/T to determine the location where it was stopped. Give the DDT command, and do some work within the DDT program; leave DDT with a CTRL/Z, returning to TOPS-20 command level. Give the START command to start FILCOM again, using as argument the address reported by CTRL/T above.

```
@FILCOM

*TTY:=DUMPER.MAC, BACKUP.MAC
^C
^T 14:49:03 FILCOM ^C from Running at 400543  Used 0:00:03.1 -
in 0:04.39, Load 2.44 in 0:01:33

@DDT
DDT
3/   PAT..+361,,3066
4/   56
^Z
@START 400543
```

```
No differences encountered
```

4. Place the CLOCK program (not DIGITAL supported) in memory. KEEP the CLOCK program so that it will remain in memory when you run other programs. Then, start the CLOCK program in the background. CLOCK will display the time every half hour while you run other programs and EXEC commands.

```
@GET CLOCK.EXE
@KEEP
@START CLOCK.EXE /STAY
@
        .
        .
        .
[11:30 AM]
```

2.75 SUBMIT

Enters a command procedure into the batch job queue.

Format

@**SUBMIT** (BATCH JOB) **/switch(es) filespec/switch(es),...**

where:

switches — are keywords, chosen from the list below, indicating your choice of SUBMIT command options. These switches have different effects according to their position in the command line: placed before all files in the command, they act as defaults for all; otherwise they affect only the nearest preceding file.
Defaults are shown in the list of switches

filespec — is the specification of a batch control file (see the TOPS-20 User's Guide), containing batch commands and the commands with which you would have done the job as a timesharing user instead of as a batch user
Default file type - .CTL

Summary of SUBMIT Command Switches (defaults in boldface)

/ACCOUNT:account	**Default** account - your current account
/AFTER:date and/or time	
YES	
/ASSISTANCE:NO	
APPEND	
/BATCH-LOG:SUPERSEDE	
SPOOL	
/BEGIN:n	**Default** n - 0
/CARDS:n	**Default** n - 1000
/CONNECTED-DIRECTORY:dev:<directory>	
/DELETE	
/DEPENDENCY-COUNT:n	**Default** n - 0
/DESTINATION-NODE:node name	
/FEET:n	**Default** n - 200
/JOBNAME:6-character name	**Default** name - first six characters of control filename
KEEP	
/LOGDISPOSITION:DELETE	
/LOGNAME:filespec	**Default** filespec - control filename, file

type .LOG

YES
/**NOTIFY:NO**
ALWAYS
/**OUTPUT:**ERRORS
NOLOG
/PAGES:n **Default** n - 200
/**PRIORITY:**n **Default** n - 10
/PROCESSING-NODE:node name
/READER
NO
/**RESTARTABLE:**YES **Default** argument (if switch is given) - YES

/SEQUENCE:n
/TAG:6-character label
/**TIME:**hh:mm:ss **Default** time limit (if switch is omitted) - 00:05:00
Default hh:mm:ss (if switch is given without colon or argument) - 60 (minutes)
/TPLOT:n **Default n - 200**
NO or 0
/UNIQUE:YES or 1
/USER:user name

SUBMIT Command Switches

/ACCOUNT:account specifies the account of 39 or fewer characters to charge for your batch request. This account must be valid for your user name.
Default account - your current account (check with INFORMATION JOB-STATUS)

/AFTER:date and/or time, **or**
day of week (or TODAY)
and/or time ensures that the job will not be started until after the date and/or time specified. NOV-12-79, and 18:00:00 illustrate two arguments to this switch. If you give both date and time, separate them with a space. When given alone, the time may be preceded with a plus sign (+), which

will delay processing by the indicated length of time from the present.

Alternatively, you may give a day of the week (such as MONDAY) or TODAY as argument; then the batch job will not be started until the beginning of the following day. If you follow this argument with a plus sign and a time, the job will be further delayed by this amount.

YES
/ASSISTANCE:NO — tells the system whether your job will require the assistance of the operator (for example, to mount a structure or magnetic tape) when it is run
Default - YES

APPEND
/BATCH-LOG:SUPERSEDE
SPOOL — tells the system either to append the log file of the batch job to any existing log file of the same name, or to write a new generation of the log file, or to send the log file to the spool area only.
Default - APPEND

/BEGIN:n — starts processing the control file at line n of the file. Use this switch for a control file that can fit different applications depending on where processing begins. (See also the /TAG switch.)
Default n - 0

/CARDS:n — limits to n the maximum number of cards to be punched by the job.
Default n - 1000

/CONNECTED-DIRECTORY:dev:<directory>
specifies the connected directory for the batch job. For privileged users only.

/DELETE — tells the system to delete the control file after the batch job has run.

/DEPENDENCY-COUNT:n

sets the job's dependency count to n. Because a batch job does not get processed until its dependency count is 0, you can delay a job by assigning it a positive dependency count and then using the MODIFY command to bring the count to 0 at the proper time.

Default n - 0

/DESTINATION-NODE:node-name

specifies the IBM remote job entry station on whose line printer the log file of your batch job is to be printed. Two colons (::) following the node name are optional.

/FEET:n

limits to n the maximum number of feet of paper tape to be punched by the job.

Default n - 200

/JOBNAME:name

assigns a name (of six or fewer characters) to the batch job.

Default name - first six characters of control filename

KEEP
/LOGDISPOSITION:DELETE

tells the system whether to delete the log file after it has been printed.

Default - KEEP

/LOGNAME:filespec

specifies where to place the log file of the batch job.

Default dev:<directory> - your connected directory at the time of the SUBMIT command

Default filename - control filename

Default type - .LOG

YES
/NOTIFY:NO

tells the system whether to send a message to your terminal /xp /NOTIFY switch when the batch job has been completed.

Default argument - NO

Default argument (if switch is given) - YES

ALWAYS
/OUTPUT:ERRORS
NOLOG

says whether you want the log file to be printed always, or only in the case of unhandled errors occurring within the job, or never. No matter which option you choose, the log file is always written.

Default - ALWAYS

/PAGES:n

limits to n the maximum number of pages of line printer output to be printed by the job.

Default n - 200

/PRIORITY:n

assigns a decimal number n to the job, reflecting the urgency of the batch request. This n must be from 0 to 63, with larger numbers receiving earlier treatment. The system acknowledges this switch by displaying the message [Priority has been modified].

Default n - 10

/PROCESSING-NODE:node name::

specifies the IBM host system on whose CPU the JCL batch job is to be run. Two colons (::) following the node name are optional.

/READER

tells the system that your control file is composed of card images, including control cards, on disk. For details see the TOPS-10/TOPS-20 Batch Reference Manual.

NO
/RESTARTABLE:YES

decides whether the job should be started again if the system crashes and is restarted.

Default argument - NO

Default argument (if switch is given) - YES

/SEQUENCE:n

specifies that n, instead of a number supplied by the system, is to be the sequence number of the job.

/TAG:label — starts processing the control file at the line beginning with label::, where label is an alphanumeric name of six or fewer characters. Use this switch for a control file that can fit different applications depending on where processing begins.

/TIME:hh:mm:ss — limits the maximum amount of CPU time available to the job; given in hours, minutes, and seconds.
Default time limit (if switch is omitted) - five minutes

Default hh:mm:ss (if switch is given without colon or argument) - 60 (minutes)

/TPLOT:n — limits to n the maximum number of minutes of plotter time allowed for plotter time allowed for the job.
Default n - 200

/UNIQUE:NO (or 0)
/UNIQUE:YES (or 1) — when submitting multiple batch jobs, tells the system whether to run the jobs concurrently or at separate times. The control files must be submitted while connected to the same directory; the control files can be located in any directory. This switch applies to batch jobs submitted with a single or multiple SUBMIT commands. See Example 5.
Default - YES

/USER:user name — specifies the user who is to be the owner of the batch request. For privileged users only.

Output

Jobname, Request ID, and Time Limit

As soon as you complete a valid SUBMIT command, the system responds by printing, on your terminal, the jobname, request ID, and time limit for the job. Each control file you submit is a separate batch request, and is described on a separate line.

Characteristics

Switch Defaults Set by System Manager

The defaults shown in the list of switches are correct for most user sites. However, your system manager can change some of these default settings. The changes go into effect during system installation. The switches most commonly affected are: /CARDS, /FEET, /OUTPUT, /PAGES, /PRIORITY, /TIME, and /TPLOT.

Disposition of Log Files

The three SUBMIT command switches /BATCH-LOG, /LOGDISPOSITION, and /OUTPUT, control what happens to the log file of your batch job.

Where Written

The log file is always written as the job runs, either to the batch job's connected directory, or to a directory specified as argument to the /LOGNAME switch, or to the system's output spooling area (it is written to the spooling area only if you give the /BATCH-LOG:SPOOL switch). If the /DESTINATION-NODE switch is also given, the log file will be written into a directory or spooling area at the specified node. Remember that a batch job's connected directory is ordinarily defined to be your connected directory at the time of the SUBMIT command; privileged users may specify a batch job's connected directory by using the /CONNECTED-DIRECTORY switch.

How Written, When Printed

The /BATCH-LOG switch's APPEND and SUPERSEDE arguments describe the manner in which the log file is to be written: either it is appended to any existing file of the same name (usually produced by a previous running of the batch job) or it is written as a new generation of the file. The /LOGDISPOSITION switch tells the system whether to keep this file, wherever it is written, once the batch job is finished. The /OUTPUT switch specifies when you want a listing of the log file to be printed: either always, or never, or only if errors occur when the batch job is run. By using combinations of these switches you can cause any desirable action. Giving /OUTPUT:ALWAYS along with /LOGDISPOSITION:DELETE allows a record of your batch job with only a temporary use of your disk area, and permits you to monitor the progress of the job while it is running (give TYPE commands to view the file at your terminal). Giving just the /BATCH-LOG:SPOOL switch allows a record without any

use of your disk area, although then you must wait for the printed output to see this record.

Execution of Command Files

As soon as one of your batch jobs logs in, the system processes your login directory's command files and the system's command files. The files are processed in this order:

1. SYSTEM:BATCH.CMD
2. BATCH.CMD
3. SYSTEM:COMAND.CMD
4. COMAND.CMD

If a LOGOUT command is included in the batch control file, the system processes your login directory's LOGOUT.CMD file followed by the system's SYSTEM:LOGOUT.CMD file. These files are not processed if the batch job is logged out automatically.

Automatic Logout

If the batch control file is not terminated by a LOGOUT command, the batch job is logged-out automatically and the message "KILLED BY OPERATOR TTYnn" is printed in the log file.

Hints

Using SET DEFAULT SUBMIT

If there are switches that you always or usually supply when using SUBMIT, give the SET DEFAULT SUBMIT command to establish them as defaults for the remainder of your terminal session. The switches will then behave as if you had typed them directly after the word SUBMIT. You can supersede any of these default switches by actually supplying the switch, with another value, when you give the SUBMIT command.

For Future Terminal Sessions

Put SET DEFAULT SUBMIT commands into a file named COMAND.CMD or LOGIN.CMD in your log-in directory if you want these default switches to be in effect for batch jobs you submit during future terminal sessions as well. If both files exist, the system reads LOGIN.CMD first.

For Nested Batch Jobs Only

Put SET DEFAULT SUBMIT commands into a file named BATCH.CMD in your log-in directory if you want them to be in effect at the log-in time of a "nested" batch job only, that is, a batch job started by a SUBMIT command within the control file of another of your batch jobs. Note, however, that the system also reads COMAND.CMD at the log-in time of a batch job if the file exists in your log-in directory. It reads this file after BATCH.CMD.

Monitoring the Progress of a Batch Job

You can include the SEND command or the commands that run the MAIL or DECmail/MS mail programs in your batch control file. Use these commands to send messages to your terminal informing you of the status of your batch job. Since a batch job creates a job in addition to your timesharing job, use the terminal line number argument instead of the user name argument in the SEND command.

More Information

For more information about batch jobs, see the TOPS-10/20 Batch Reference Manual.

Restrictions

Access Rights for Batch Jobs

For Specifying Control Files and Log Files

You cannot use the ACCESS command to obtain the right to submit control files from another directory, because your batch jobs are logged in with rights only to your connected directory and to directories to which you (through your login directory name) have access as a group member. The control file, if not in your connected directory, must be in one to which you have read access as a group member; the log file specification, if you give one, must be for your connected directory or for one to which you have write access as a group member.

For Use Within the Batch Job

Although it is possible to give CONNECT and ACCESS

commands within a batch job to obtain rights beyond those mentioned above, you may then have to include passwords in the job's control file. Because this practice could endanger system security, it is generally best to establish and rely on appropriate group rights when preparing batch jobs for submission.

PUSH During Batch Job Execution

Note that a PUSH command reads the COMAND.CMD file. Therefore, if your batch control file contains a PUSH command, only the defaults set in COMAND.CMD are in effect while in the inferior EXEC.

Editing a Queued Control File

The batch system processes the exact version of the control file specified in your SUBMIT command. Therefore, if you edit a file while it is in the batch queue, the new version of the file will not be processed.

To change the request to process the latest version of the control file, CANCEL the request and resubmit the job.

Related Commands

CANCEL	for withdrawing SUBMIT requests
INFORMATION BATCH-REQUESTS	for examining in the batch input queue
MODIFY	for changing SUBMIT requests before processing has begun
SET DEFAULT SUBMIT	for establishing default switches for subsequent SUBMIT commands

Examples

1. Submit a control file to begin a batch job.

```
@SUBMIT DIFS.CTL
[Batch job DIFS queued, request-ID 461, limit 0:05:00]
```

2. Submit two control files (specifying only the filenames) in the same command. then use the information batch-requests command (with the /USER switch) to examine your entries in the batch input queue.

```
@SUBMIT SUMS, DIFS
[Batch job SUMS queued, request-ID 629, limit 0:05:00]
[Batch job DIFS queued, request-ID 630, limit 0:05:00]
@INFORMATION BATCH-REQUESTS /USER

Batch Queue:
Job Name  Req#  Run Time  User
--------  ----  --------  --------
* SUMS     629  00:05:00  C.BURKE In Stream:2
   Started at 15:21:01
  DIFS     630  00:05:00  C.BURKE
There are 2 jobs in the queue (1 in progress)
```

3. Connect to another user's directory, then submit two of his control files. Prevent the printing of a log file for one job, and allow the second job's to be printed only if errors occur within the job; make both jobs restartable. Request an inclusive listing of your entries in the batch queue - notice that the jobs are logged in under your own user name, although the log files will be stored in user Holland's directory. Note also that an asterisk (*) indicates a job currently in progress.

   Connect back to your directory and submit one of your own control files, specifying a particular jobname, then check on it.

```
@CONNECT <HOLLAND>
Password:___
@SUBMIT /RESTARTABLE:YES FLDTST.CTL/OUTPUT:NOLOG, LODT.CTL -
/OUTPUT:ERRORS
[Batch job FLDTST queued, request-ID 464, limit 0:05:00]
[Batch job LODTST queued, request-ID 465, limit 0:05:00]
@INFORMATION BATCH-REQUESTS /ALL/USER

Batch Queue:
Job Name  Req#  Run Time             User
--------  ----  --------  --------------------------------
* FLDTST   464  00:05:00  C.BURKE In Stream:2  /Uniq:Yes
          /Restart:Yes  /Assist:Yes  /Seq:1993
   Started at  8:40:38
  LODTST   465  00:05:00  C.BURKE /Uniq:Yes  /Restart:Yes
          /Assist:Yes  /Seq:1994
There are 2 Jobs in the Queue (1 in Progress)

@CONNECT MISC:<C.BURKE>
@SUBMIT SUMS/JOBNAME:1-SUMS
[Batch job 1-SUMS queued, request-ID 466, limit 0:05:00]
```

```
@INFORMATION BATCH-REQUESTS /ALL/USER

Batch Queue:
Job Name  Req#  Run Time             User
--------  ----  --------  --------------------------------
* 1-SUMS   466  00:05:00  C.BURKE In Stream:2  /Uniq:Yes
          /Restart:No  /Assist:Yes  /Seq:1995
   Started at 8:41:29
There is 1 job in the queue (1 in progress)
```

4. Give a SET DEFAULT command to ensure that your batch jobs will be run after 5:00 P.M. unless you specify otherwise. Submit a batch job and check that this default is in effect. Then use a MODIFY command to delay the starting time of this job till 11:00 P.M. Finally, give the CANCEL command to withdraw the batch request entirely.

```
@SET DEFAULT SUBMIT /AFTER:17:00
@SUBMIT SUMS
[Batch job SUMS queued, request-ID 467, limit 0:05:00]
@INFORMATION BATCH-REQUESTS /USER

Batch Queue:
Job Name  Req#  Run Time             User
--------  ----  --------  --------------------------------
  SUMS     467  00:05:00  C.BURKE /After: 9-Nov-85 17:00
There is 1 Job in the Queue (None in Progress)
@MODIFY BATCH 467 /AFTER:23:00
[1 Job modified]
@INFORMATION BATCH-REQUESTS /USER

Batch Queue:
Job Name  Req#  Run Time             User
--------  ----  --------  --------------------------------
  SUMS     467  00:05:00  C.BURKE /After: 9-Nov-85 23:00
There is 1 Job in the Queue (None in Progress)

@CANCEL BATCH 467
[1 Job canceled]
```

5. Submit two control files, one located in your connected directory and the other in a directory you are accessing. Use the /UNIQUE:NO switch to allow the jobs to run simultaneously. Display the status of the batch queue and note that both jobs are running.

```
@SUBMIT NETCOM.CTL, RANDOM:[LOWELL]CLEAN.CTL /UNIQUE:NO
[Batch job CLEAN queued, request  58, limit 0:05:00]
[Batch job NETCOM queued, request  59, limit 0:05:00]
@INFORMATION BATCH-REQUESTS
```

```
Batch Queue:
Job Name   Req#   Run Time            User
--------  ------  --------  -------------------------
* CLEAN       58  00:05:00  DBONIN                In Stream:0
     Job# 156 Running EXEC Runtime 0:00:16
* NETCOM      59  00:05:00  DBONIN                In Stream:1
     Job# 156 Running EXEC Runtime 0:00:08
   GTSTK       2  00:15:00  PURRETTA
   CIGIDN      3  01:00:00  CSSE.WAIBLE
There are 4 jobs in the queue (2 in progress)
```

2.76 SYSTAT

Displays information about the jobs on the system.

Format*

@SYSTAT [NODE node name],
@@subcommand

where:

NODE node name is an optional keyword and argument that is used to display information about the jobs on a specified node in the TOPS-20 cluster. If an asterisk is specified as the node name, the command displays information on all nodes in the TOPS-20 cluster.

@@subcommand means that, after a comma, you can give one or more subcommands on successive lines

Summary of SYSTAT Subcommands (defaults in boldface)

ALL
CLASS
| CONNECT-TIME
CONTROLLING
DIRECTORY
HEADER
JOB job number n
LIMIT
LINE octal line number, or DETACHED
LPT
NO subcommand name, or OPERATOR, or .
NODE
ORIGIN
OUTPUT file specification
PROGRAM
STATE
SYSTEM
TIME
USER user name
WHAT
WHERE
WHO

* For information on the in-line subcommand format, see the "Hints"

section below.

SYSTAT Subcommands

ALL	gives all available SYSTAT information
CLASS	prints the scheduler class in which each job is running; the share of total CPU time allotted to the job, expressed as a decimal fraction; and the fraction of total CPU time actually used by the job. A job's actual use may be larger than its allotted share if some jobs in its class are inactive; it can be larger still if other classes are inactive and this unused fraction of CPU time is being allocated among active jobs.
CONNECT-TIME	prints how long each user has been connected to the system.
CONTROLLING	prints, in the column headed CJB, the number of the controlling job (if any), that is, a job owning a PTY (pseudo-terminal) that controls the job being described; when used in a SYSTAT command requesting descriptions of particular jobs, this subcommand causes jobs controlled by these jobs to be described also.
DIRECTORY	requests the name of the directory to which each job is connected, if not the job's log-in directory.
HEADER	calls for a headline identifying the columns of information printed **Default** (unless you are requesting information about specific users, jobs, or lines only; in such cases the default is NO HEADER.)
JOB n	restricts output to description of job number n; can be used more than once.
LIMIT	prints any time limit set for each job with the SET TIME LIMIT command. Print the amount of CPU time used by your job with CTRL/T or INFORMATION PROGRAM-STATUS.

LINE octal line number **or** DETACHED
restricts output to description of the job attached to the given line number, or to descriptions of all detached jobs; can be used more than once.

LPT sends output to the line printer instead of to your terminal.

| . (period)
| CLASS
| CONNECT-TIME
| CONTROLLING
| DIRECTORY
| HEADER
| LIMIT
| NODE eliminates the indicated category of
| OPERATOR information, when used with one of the
NO | ORIGIN keywords shown (. refers to your own job)
| STATE
| SYSTEM
| TIME
| WHAT
| WHERE
| WHO

NODE displays information about the jobs on the specified node.

ORIGIN displays the job's originating system, that is, the system from which the user connected to this system.
Default

OUTPUT filespec sends the output information to the file you specify, instead of to your terminal.
Default filespec - SYSTAT.LST

PROGRAM program name restricts SYSTAT output to descriptions of jobs using the program (or TOPS-20 command) specified. The argument you supply must be of six or fewer characters.

STATE prints the current state of each job, for example RUN (running), or TI (waiting for terminal input)

SYSTEM [NODE node name]
begins output with system-wide information

(the first two lines of regular output). If SYSTEM is the only subcommand given, SYSTAT output is restricted to this. NODE node name is an optional keyword and argument that displays information on the specified node(s) in the TOPS-20 cluster. If you specify an asterisk as the node name, information on all nodes in the TOPS-20 cluster appears.

Default (unless you give subcommands requesting information about specific users, jobs, or lines only; in such cases the default is NO SYSTEM.)

TIME — prints the accumulated runtime (CPU time) for each job

USER user name — restricts output to descriptions of jobs logged in under the given user name; can be used more than once.

WHAT — prints the name of the program that each job is running; given explicitly only with subcommand NO, to restrict SYSTAT output.

Default

WHERE — prints the line number associated with each job; given explicitly only with subcommand NO, to restrict SYSTAT output.

Default

WHO — prints the user name under which each job is logged in; given explicitly only with subcommand NO, to restrict SYSTAT output.

Default

Output

Sample of SYSTAT Output

The SYSTAT command displays on your terminal columns of information about all the jobs on the system. Below is a sample of the output you would receive in response to a SYSTAT command that eliminates the two rightmost columns (User and <Directory>):

```
@SYSTAT ALL NO WHO NO DIRECTORY NO CONNECT TIME
 Tue 14-Aug-79   15:48:37   Up 1:12:59
 45+11 Jobs   Load av (class 0)   3.70   3.54   3.71
```

```
 Job  CJB Line  Program State    Time     Cls  Shr   Use
   5       25   TV      RUN    0:01:02   0  0.01   0.03
   7        6   TV      TI     0:00:35   0  0.01   0.02
  13   35  217   EXEC    RUN    0:01:02   1  0.15   0.00
   .    .    .      .      .        .      .    .      .
   .    .    .      .      .        .      .    .      .
   .    .    .      .      .        .      .    .      .
```

First you see the current date and time (in 24-hour notation: the sample above was obtained 37 seconds after 3:48 P.M.), and the length of time since the system was started (here, just over 1 hour).

The second line displays the number of user jobs (45) and operator jobs (11) currently running. The next three numbers are the "load averages" for the system: these are weighted averages of the number of runnable processes on the system over the last minute, 5 minutes, and 15 minutes, respectively. (If class scheduling is enabled, the three load averages are the average number of jobs having at least one runnable process, and apply to the class in which your job is running.) If you are about to start a job requiring 5 minutes of CPU time, and the load average remains stable over the period in question, for example, becomes 4.54 (3.54 + your job = 4.54), then you can count on your job's getting about 1/4.54 of your class's share of the system's attention. If your class is assigned one third of the system's CPU time and you do not receive any windfall (unclaimed CPU time), your job will be finished in a little over one hour.

After this comes the line of headings labeling each column of data that follows. All but the User, Origin and <Directory> columns of information appear in the sample above, describing all jobs (rows). The unmodified command SYSTAT displays the Job, Line, Program, User and Origin columns. The Origin column displays the job's originating host followed in parentheses by the network terminal type. For example, AURORA (CTM) means that the user logged in to node AURORA and then used the CTERM-SERVER protocol to connect his terminal to this system. The definitions of the network terminal types are:

CTM	CTERM-SERVER protocol terminal
LAT	Local Access Terminal
NRT	Network Remote Terminal
TCP	Internet TCP/IP Terminal

By using appropriate subcommands you can select the categories of information, as well as the specific jobs. (The Class (Cls) and Share (Shr) categories appear only if

class scheduling is enabled.) User jobs, both timesharing and batch, are listed first (in ascending order by job number), and then, after a blank line, operator jobs. The number of the job attached to your own terminal appears with an asterisk (*) next to it in the Job column.

Sending Output Elsewhere

By giving the OUTPUT subcommand you can direct SYSTAT information to a file instead of to your terminal. The subcommand LPT sends the information to the printer instead of to your terminal.

Characteristics

Log-in Not Necessary

You do not have to be logged in to give the SYSTAT command. However, the system manager can disable the SYSTAT command for users not logged in.

Hints

Giving Subcommands as Arguments on the Command Line

To simplify your typing, SYSTAT accepts subcommands as arguments given on the same line as the command, subject to these rules:

There will be no @@ prompt: simply type a space between successive subcommands and between subcommand names and arguments.

To get information about one or more specific job numbers, give the numbers only; do not type JOB.

To get information about one or more specific user names, give the names only; do not type USER. But if the user name is by coincidence the same as a SYSTAT command argument, you must use the subcommand mode to request information about his job.

To get information about one or more specific log-in directories, give the directory names.[1]

[1] For specific connected directories, specify the directory names (and structures, if not the public structure) along with either the ALL or DIRECTORY subcommand.

To get information about your own (attached) job only, give a period (.) as argument.

To get information about all other jobs logged in under your user name, give your user name and NO . as arguments.

The system will not accept the OUTPUT subcommand in this format; use the subcommand mode instead.

Special Cases

OPERATOR as a User Name

You can request or refuse information about operator jobs by treating OPERATOR as a user name. The system accepts these commands:

```
@SYSTAT OPERATOR
```

and

```
@SYSTAT,
@@USER OPERATOR
```

as well as the special commands

```
@SYSTAT NO OPERATOR
```

and

```
@SYSTAT,
@@NO OPERATOR
```

Related Commands

INFORMATION	for finding out other information about the system

Examples

1. Find out the status of all jobs on the system. (Your current (attached) job is marked with an asterisk (*).)

```
@SYSTAT
 Thu 17-May-90 12:27:44  Up 9:32:11
```

```
|            12+8 Jobs   Load av   0.33   0.27   0.20
|
|            Job  Line Program  User              Origin
|            135   DET  DTRSRV  Not logged in
|            136   DET  RMSFAL  Not logged in
|            137   DET  RMSFAL  Not logged in
|            138   DET  RMSFAL  Not logged in
|            139   DET  RMSFAL  Not logged in
|            142   434  MS      BRAITHWAITE       LAT70:24008_70(LAT)
|            143   435  MS      WONG              LAT462:24083_462(LAT)
|            144   437  MS      EKLUND            LAT75:24052_75(LAT)
|            145   DET  EXEC    UVA
|            146   436  MS      JMCGREAL          LAT1:LAT1_17(LAT)
|            147   440  EMACS   MONTEIRO          LAT1:LAT1_27(LAT)
|            148   441  EXEC    JBREWER           LAT73:24064_73(LAT)
|            149   314  EXEC    GSCOTT            klipa.tops20.dec.com(TCP)
|            150   442  EXEC    LOMARTIRE         LAT1:24087_1_1(LAT)
|            151   443  EXEC    GSCOTT            LAT1:24086_1(LAT)
|            152   243  NFTP    GSCOTT
|            153*  445  SYSTAT  ANDERSON          LAT423:24081_1_423(LAT)
|
|            129   232  PTYCON  OPERATOR
|            130   233  GALAXY  OPERATOR
|            131   234  NEBULA  OPERATOR
|            132   235  UNIVER  OPERATOR
|            133   236  EXEC    OPERATOR
|            134   237  MX      OPERATOR
|            140   240  DTR     OPERATOR
|            141   241  DIU     OPERATOR
```

2. Find out the status of all jobs on the NODE VENUS.

```
@SYSTAT NODE VENUS
 Thu 13-Aug-87 13:08:12
 VENUS  Up   0:10:33 17+6 Jobs   Load av   0.11   0.13   0.12

 Job  Line Program  Node    User            Origin
 231   DET  DTRSRV  VENUS   Not logged in
 232   DET  RMSFAL  VENUS   Not logged in
 233   DET  RMSFAL  VENUS   Not logged in
 234   434  EXEC    VENUS   DAVE
LAT423:2408_1_423(LAT)
 235   435  EXEC    VENUS   RICH         LAT73:24064_73(LAT)
 236   436  OPR     VENUS   GAGNE
LAT75:24067_2_75(LAT)
 237   437  EXEC    VENUS   GSCOTT
klipa.tops20.dec.com(tcp)
 238   440  NFT     VENUS   rich
pmap.tops-20.dec.com(tcp)
 239   441  EXEC    VENUS   BROOKS       LAT1:LAT1_27(LAT)
 240   442  EXEC    VENUS   PUCHRIK      LAT1:24053_1(LAT)
```

```
 241   447  EMACS   VENUS   ROSSELL
LAT473:24112_473(LAT)
 242   443  EXEC    VENUS   PRATT        LAT70:24008_70(LAT)
 243   444  SYSTAT  VENUS   RASPUZZI     LAT1:24086_1(LAT)

 225   232  GALAXY  VENUS   OPERATOR
 226   233  PTYCON  VENUS   OPERATOR
 227   234  DIU     VENUS   OPERATOR
 228   235  MAILS   VENUS   OPERATOR
 229   236  WATCH   VENUS   OPERATOR
 230   237  EXEC    VENUS   OPERATOR
```

3. Determine how much CPU time has been charged to the jobs of two users on the system.

```
@SYSTAT,
@@TIME
@@USER KONEN
@@USER ALUSIC
@@
  27    66    EXEC    0:00:01    ALUSIC
  43    11    EXEC    0:00:02    KONEN
```

4. Repeat Example 2 by giving the subcommands as arguments on the same line.

```
@SYSTAT TIME KONEN ALUSIC
  27    66    EXEC    0:00:01    ALUSIC
  43    11    EXEC    0:00:02    KONEN
```

5. Find out who is using line 11.

```
@SYSTAT LINE 11
  43    11    EXEC    KONEN
```

6. Ask for information about jobs 5 and 45.

```
@SYSTAT 5 45
   5    56    MACRO   D.SCHEIFLER
  45    205   PTYCON  OPERATOR
```

7. Set a time limit of 4 seconds for your attached job, then ask for complete information, including headings, for the job. (The period (.) specifies your attached job.) The value reported under the Limit heading is actually the sum of the time limit you set (4 seconds) and the amount of CPU time already used at the time of your SET command (2 seconds). This CPU time is reported as 2 seconds under the Time heading because you gave the SYSTAT command immediately after SET.

```
@SET TIME-LIMIT 4
```

```
@SYSTAT ALL HEADER .
  Job   CJB  Line   Program  State  Time    Cls   Shr   Use
Limit   User, <Directory>  Origin
  14*  51   207    EXEC     RUN    0:00:02   0  0.02  0.02
0:00:06 LATTA, MISC:<LATTA>
```

8. Ask for system-wide SYSTAT information only.

```
@SYSTAT SYSTEM
 Fri 1-Mar-84 12:35:44  Up 33:43:36
 18+15 Jobs   Load av (class 0)   5.19   3.36   2.92
```

Ask for system-wide SYSTAT information for nodes GIDNEY and CLOYD.

```
@SYSTAT SYSTEM NODE GIDNEY NODE CLOYD
 Thu 13-Aug-87 13:02:00
 GIDNEY Up 223:12:12 17+6 Jobs  Load av  0.36  0.27  0.14
 CLOYD  Up  26:34:31  6+8 Jobs  Load av  1.33  1.21  0.99
```

Now ask for system-wide SYSTAT information for all nodes in the cluster.

```
@SYSTAT SYSTEM NODE *
 Thu 13-Aug-87 13:02:00
 GIDNEY Up 223:12:12 17+6 Jobs  Load av  0.3   0.27   0.14
 VENUS  Up   0:10:33 11+5 Jobs  Load av 10.36 10.27  10.14
 CLOYD  Up  26:34:31 6+8  Jobs  Load av  1.33  1.21   0.99
 RONCO  Up  12:13:14 2+5  Jobs  Load av  5.01  4.95   4.99
```

9. Find out only which programs are in use.

```
@SYSTAT NO WHO NO WHERE NO SYSTEM
 Job Program
   135   DTRSRV
   136   RMSFAL
   137   RMSFAL
   138   RMSFAL
   139   RMSFAL
   142   MS      LAT70:2400870(LAT)
   143   MS      LAT462:24083462(LAT)
   144   MS      LAT75:2405275(LAT)
   145   EXEC
   146   MS      LAT1:LAT117(LAT)
   147   EMACS   LAT1:LAT127(LAT)
   148   EXEC    LAT73:2406473(LAT)
   149   EXEC    klipa.tops20.dec.com(TCP)
   150   EXEC    LAT1:2408711(LAT)
   151   EXEC    LAT1:240861(LAT)
   153*  SYSTAT  LAT423:240811423(LAT)
```

```
|              129   PTYCON
|              130   GALAXY
|              131   NEBULA
|              132   UNIVER
|              133   EXEC
|              134   MX
|              140   DTR
|              141   DIU
```

10. Ask for a list of jobs controlled by job 51. (Your attached job, marked with an asterisk (*), happens to be one of these; job 51 itself is the other.)

```
@SYSTAT 51 CONTROLLING
  14*  51  207  EXEC    LATTA
  51         41  NEWRUN  LATTA
```

11. Do a SYSTAT command that displays the amount of time each user has been connected to the system.

```
@SYSTAT,
@@CONNECT-TIME
@@NO OPERATOR
@@
Tue 13-Sep-88 13:05:54  Up 303:54:45
10+7 Jobs   Load av   0.08   0.10   0.09

Job  Line Program Connected  User             Origin
  80   434  MS       3:15:04  BRAITHWAITE       LAT70:24008_70(LAT)
  81   435  EXEC     0:32:30  BARR              LAT1:LAT1_17(LAT)
  82   314  MS     118:07:35  GSCOTT            klipa.tops20.dec.com(TCP)
  83   315  EXEC   117:38:41  GSCOTT            klipa.tops20.dec.com(TCP)
  84   440  EXEC   117:26:00  WONG              LAT462:24083_462(LAT)
  85   437  SORT     2:03:23  FONG              LAT64:24062_2_64(LAT)
  86   441  EXEC    22:50:16  JROSSELL          LAT462:24085_A_462(LAT)
  87   444  MS       1:59:40  EKLUND            LAT75:24052_75(LAT)
  88   443  EXEC    22:31:22  JROSSELL          SCROOM:TWA94(LAT)
  89   436  EMACS    1:44:53  FONG              MATRIX:TWA14(LAT)
  90   442  MS       1:28:19  BHAMILTON         GNOME:TWA48(LAT)
  91   445  EXEC     1:18:12  JBREWER           LAT73:24064_73(LAT)
  92*  446  SYSTAT   0:01:22  ANDERSON          LAT423:24081_1_423(LAT)
```

2.77 TAKE

Processes a TOPS-20 command file.

Format

@TAKE (COMMANDS FROM) **filespec**,
@@subcommand

where:

filespec — is the specification of the file containing the commands to be processed.
Default file type - .CMD

@@subcommand — means that after a comma you can type one of the following subcommands:

ALLOW — tells the current level of TOPS-20, for the remainder of the terminal session (not merely the current command), to continue processing a command file if it encounters errors.

DISALLOW — tells the current level of TOPS-20, for the remainder of the terminal session (not merely the current command), to ignore any remaining commands in a command file after it encounters an error in the file.
Default

ECHO — tells the system to print (on your terminal or in the specified file) the commands that it carries out while executing the current TAKE command. Ordinarily only the output, if any, produced by the commands is printed.

NO ECHO — tells the system not to print the commands that it carries out while executing the current TAKE command. A final message is sent, however, indicating whether all the commands were executed. See also Hints - Suppressing the Final Message, below.
Default

LOG-FILE filespec

tells the system to save the output from the current TAKE command in the specified file.

Output

The output from a TAKE command consists of the output for each command in the command file you specify as argument, followed by the message, End of filespec, that indicates successful execution of all the commands in this file.

Characteristics

Running Programs From a Command File

If you put commands that run programs (including the PUSH command) into a command file, and these programs ask for arguments, you must be ready to type in these arguments at your terminal. Only TOPS-20 commands and command arguments can be put into a command file executed by the TAKE command.

Hints

Suppressing the Final Message

If you want to suppress the final message (of the form, End of filespec) that indicates successful execution of a command file by TAKE, give a TAKE command with no arguments as the last line of your command file.

Special Cases

Nested TAKE Commands

In the case of nested TAKE commands (those given as commands within command files), the destination for output of commands given in an inner command file will default to that specified or assumed for the output of commands given in the nearest surrounding command file.

Effect on Memory and Terminal

The TAKE command affects memory and your terminal according to the commands stored in the command file you specify as argument.

Related Commands

INFORMATION commands	(when put into a command file) for tracing the progress of TAKE
LOGIN	for logging in; reads LOGIN.CMD then COMAND.CMD, in your log-in directory.
PUSH	for obtaining a new level of TOPS-20; reads COMAND.CMD in your log-in directory.
SUBMIT	for processing command files that run programs and contain program commands as well as TOPS-20 commands; reads BATCH.CMD, then COMAND.CMD, in your log-in directory.

Examples

1. Process a command file.

```
@TAKE BACKUP.CMD
 End of BACKUP.CMD.1
```

2. Type a command file that reports system statistics, then give the TAKE command with this filename as argument; send the output to the line printer. Check for this listing as it is being printed.

```
@TYPE STATUS.CMD
INFORMATION DISK-USAGE
INFORMATION MONITOR-STATISTICS
INFORMATION SYSTEM-STATUS
INFORMATION MEMORY-USAGE
SYSTAT ALL

@TAKE STATUS LPT:
 End of STATUS.CMD.1
@INFORMATION OUTPUT-REQUESTS /USER

Printer Queue:
Job Name  Req#  Limit  User
--------  ----  -----  ----------------------
  EXEC     507     27  LATTA
There is 1 Job in the Queue (None in Progress)
```

2.78 TALK

Allows you to converse with other users on your system by linking terminals.

Format

@TALK (TO) **argument**

where:

argument is a user name or terminal line number.

Characteristics

Typing TALK Conversation

During a TALK session, you must tell the system to regard your conversation as comments. Otherwise, the system interprets your input as attempts to give EXEC commands and responds with the message ?Unrecognized command. To signal your input as comments, begin each line with the exclamation point (!) or semicolon (;) comment character. Or, if your comment is several lines long, use the REMARK command.

Other Job Not Affected

As soon as you give a successful TALK command, both terminals begin printing both users' input as well as the system's responses to that input. Each job, however, will receive input from its own terminal only.

Ending TALK

To end a conversation link between terminals, either user can give the BREAK command.

Refused TALK

Terminals can be set to refuse links with other terminals with the REFUSE LINKS or TERMINAL INHIBIT command. If you attempt to TALK to a user who has refused links from another terminal, the system rings the bells on both terminals six times, and then prints the message, ?Refused, Send mail to user instead. If the user has refused all terminal communication with the TERMINAL INHIBIT command, the system does not ring the bell on his terminal.

If you have Wheel or Operator capabilities enabled, you can

TALK to any user who has given the REFUSE LINKS command, but not the TERMINAL INHIBIT command.

Maximum of Four Terminals

By using TALK commands, you can link up to four terminals at once. For all terminals to share the same display, each pair of terminals must establish a link. For example, if terminal A is linked to B and C, terminals B and C will display only A's input. B and C must establish a link to display each other's input.

Hints

Signaling a Linked User

Once you have established links with another user's terminal via the TALK command, you can get his attention by typing a series of CTRL/Gs. Depending on the kind of terminal he has, these will be reproduced as ringing bells or high-pitched beeps. This action can be especially useful when establishing links with the owner of a display terminal, as display terminals are silent in ordinary operation.

Special Cases

User Has More Than One Job

If more than one job is logged in under the user name you specify, the system responds with a list of that user's terminal line numbers and the programs being run. Type your choice of terminal line number (if available, the one running the EXEC) after the TTY: prompt.

Talking to a Batch Job or PTYCON Job

When you link to a PTY (pseudo-terminal) to talk to the owner of a batch job or PTYCON job, the system informs you of this with a message, to which you must reply with a carriage return to confirm the link. To decline the link, give a CTRL/C. See also Warning, Talking to a Batch Job, below.

Warning

Talking to a Batch Job

Use caution when communicating through a PTY (pseudo-terminal) that is controlling a batch job: do not send a question mark (?) or percent sign (%), because these characters can be attributed to errors occurring within the job. Also, if an error actually does occur in the batch job and the batch system's question mark is displaced (by your remarks) from the beginning of a line, the system may not recognize it as an error.

Talking Between a VT100 and a VT52

If links between VT100 and VT52 terminals are established using a TALK (or ADVISE) command, the VT52 may function improperly during or after the linked interval (such as by requiring frequent CTRL/Q commands to print multiple lines of output). Turning the terminal off and then on again (after the linked interval) will correct this problem.

Related Commands

ADVISE	for sending commands to another user's job
BREAK	for ending communications links involving your terminal
RECEIVE LINKS	for allowing other users to talk to you
REFUSE LINKS	for preventing other users from talking to you
REMARK	for telling the system to regard your terminal input as comment only
SEND	for sending a message to another user's terminal
TERMINAL INHIBIT	for refusing all types of terminal communication including advice, links, system messages, user messages, and notices of new mail.

Examples

1. Give the TALK command to establish links to another user.

```
@TALK H.DAVIES

[Link from LATTA, TTY 230]
```

2. Try to talk to a user who has given the REFUSE LINKS command, then use the MAIL program to send your message.

```
@TALK GEBHARDT
?Refused, send mail to user instead
@MAIL
To: GEBHARDT
CC: LATTA
Subject: HUNCH
     .
     .
     .
```

3. Talk to another user, giving the REMARK command immediately after TALK. (The other user's reply must still be preceded by semicolons (;) or exclamation marks (!).) Give a CTRL/Z to end REMARK before typing the BREAK command to end the conversation.

```
@TALK CARNAVON

[Link from LATTA, TTY 230]
@REMARK
Type remark. End with CTRL/Z.
WHERE DO I PUT "REQMD" RECORDS AFTER EXTRACTING THE ID'S?
@;in <accts>deft-77.cbl
@;you should have group access there...
THANKS
^Z
@BREAK
```

4. Give the TALK command to establish links to a user who has 3 jobs on three different terminals; choose one of the terminals running the TOPS-20 command processor.

```
@TALK MCKAY
 TTY19, DUMPER
 TTY26, EXEC
 TTY27, EXEC
TTY: 27

[Link from LATTA, TTY 230]
```

2.79 TDIRECTORY

The TDIRECTORY (Time-ordered DIRECTORY) command is equivalent to the DIRECTORY command with the subcommands CHRONOLOGICAL WRITE, REVERSE, and TIMES (AND DATES OF) WRITE. Use the same format and subcommands with TDIRECTORY as with DIRECTORY. For further information, see the DIRECTORY command description.

When used with magnetic tapes, the TDIRECTORY command is equivalent to DIRECTORY for magnetic tapes.

Examples

1. Give a TDIRECTORY command, cancel the command with a CTRL/C after the first few (most recent) files are displayed.

```
@TDIRECTORY

                Write


   MISC:<LATTA>
 TBATCH.CMD.1   10-May-79 13:11:57
 B.DIRECTORY.1   9-May-79 12:54:00
 A.DIRECTORY.1   2-May-79 13:14:52
T.CMD.1 ^C
```

2. Access another user's directory, and request a time-ordered directory listing of all his files of a certain name.

```
@ACCESS <DEVRIES>
Password:___
@TDIRECTORY <DEVRIES>SYSTEM.*

                Write


   MISC:<DEVRIES>
 SYSTEM.MEM.1    19-May-79 09:03:48
   .TXT.1        19-May-79 09:02:08
   .RNO.1        19-May-79 09:02:00

 Total of 3 files
@END-ACCESS <DEVRIES>
```

2.80 TERMINAL

Sets the characteristics of your terminal.

Format

@**TERMINAL** (FEATURE or TYPE) **argument**

where:

argument — is a keyword, chosen from the list below, representing your choice of TERMINAL command options; some arguments further require a decimal number to complete their meaning.

Summary of TERMINAL Command Arguments (defaults in boldface)

The TERMINAL command arguments are divided into two categories, feature and type. Feature arguments set individual terminal characteristics and type arguments set a group of characteristics that are defined for the model of your terminal.

Feature Arguments

FLAG
FORMFEED
FULLDUPLEX
HALFDUPLEX
HELP
IMMEDIATE
INDICATE
INHIBIT
LENGTH n **Default** n - 66
LINE-HALFDUPLEX
LOWERCASE

```
      ---
      | FLAG
      | FORMFEED
      | IMMEDIATE
      | INDICATE
      | INHIBIT
NO    | LOWERCASE
      | PAGE
      |       ---
      |       | CHARACTER x y
      | PAUSE | COMMAND
      |       | END-OF-PAGE
      |       ---
      | RAISE
      | RECEIVE
      |TABS
      ---
PAGE
PAUSE
RAISE
RECEIVE
      ---
      |   50
      |   75
      |  110
      |  134
      |  150
      |  100
SPEED |  300
      |  600
      | 1200
      | 1800
      | 2400
      | 4800
      | 9600
      ---
TABS
TYPE 0-36
WIDTH n  Default n - 72
```

Type Arguments

```
33
35
37
EXECUPORT
H19
LA120
LA30
```

LA36
LA38
SYSTEM-DEFAULT
TERMINET
TI
VK100
VT05
VT100
VT102
VT125
VT131
VT200-SERIES
VT300-SERIES
VT50
VT52

Feature Arguments

FLAG	instructs the system to print a single quotation mark (') before it prints an uppercase character. This takes effect only if you set the NO LOWERCASE parameter.
FORMFEED	informs the system that your terminal has a form feed mechanism; otherwise, the system simulates form feeds by printing the correct number of line feeds (set by the TERMINAL LENGTH command) if you have set TERMINAL NO INDICATE, or by printing an ^L if you have set TERMINAL INDICATE.
FULLDUPLEX	instructs the system to send to your terminal each character as the program reads it. Your terminal does not print what you type until the system sends the character back to the terminal. See also IMMEDIATE. **Default**
HALFDUPLEX	inhibits the system from sending to your terminal each character, and assumes that your terminal prints each character itself; causes echoing of format control characters (for example, TAB and line feed). Be sure also to set any corresponding switch physically located on your terminal.
HELP	prints information about the TERMINAL command.

IMMEDIATE — instructs the system to echo each character as soon as you type it, instead of waiting until the program receives the character. Immediate echoing has effect only when the FULLDUPLEX parameter is also set.

INDICATE — instructs the system to print a ^L instead of advancing the proper number of lines whenever encountering a form feed or CTRL/L (ASCII character 14).
Default

INHIBIT — notifies the system that you are not willing to receive links, advice, system messages, and user messages. Also stops beep or bell signals from users attempting to TALK to your terminal. Only output from your own job is displayed on your terminal. This command disables the settings established with the RECEIVE and REFUSE commands. Reestablish the RECEIVE and REFUSE settings with NO INHIBIT.

LENGTH n — sets the number of lines printed on each page. (If you have TERMINAL PAUSE END-OF-PAGE set as well, the system stops after printing n lines and continues only when you type CTRL/Q.) If you set the page length to 0, the system stops printing only when you type CTRL/S (as long as TERMINAL PAUSE COMMAND is in effect also); it does not automatically stop at the end of a page.
Default n - 66

LINE-HALFDUPLEX — inhibits the system from sending to your terminal each character, and assumes that your terminal prints each character itself; does not cause echoing of format control characters (for example, TAB and line feed).

LOWERCASE — tells the system that your terminal handles lowercase output characters properly, by printing either the lowercase character or the corresponding uppercase character. When NO LOWERCASE is set, the system converts lowercase output characters to the appropriate uppercase characters before sending them. See also the FLAG and RAISE parameters.
Default

NO argument — reverses any of the arguments FLAG, FORMFEED,

IMMEDIATE, INDICATE, INHIBIT, LOWERCASE, PAGE, PAUSE, RAISE, RECEIVE, and TABS

Defaults – NO FLAG, NO FORMFEED, NO IMMEDIATE, NO INHIBIT, NO PAUSE END-OF-PAGE, NO TABS

PAGE n — instructs the system to stop printing when it reaches the end of a page, or when you type a CTRL/S. Continue the output by typing a CTRL/Q. To set the page length, give the number n or give a TERMINAL LENGTH command. If you set the page length to 0, the system stops printing only when you type a CTRL/S.

Default n – argument of any TERMINAL LENGTH command given in the current terminal session, or the default page length for your terminal type

PAUSE — CHARACTER x y / END-OF-PAGE / COMMAND — instructs the system to stop sending output whenever it has sent a full page (END-OF-PAGE), or whenever you type CTRL/S (COMMAND) or x (CHARACTER).

For argument END-OF-PAGE to stop your output, argument COMMAND must also be in effect. You continue the output by typing CTRL/Q or the y parameter of the CHARACTER argument.

For argument CHARACTER to stop your output, the COMMAND and END-OF-PAGE arguments must be in effect. With the CHARACTER argument, you continue output by typing the y parameter. You can specify x and y in various ways: as the octal ASCII code for any character or control key; as any printing character in double quotes (" "); as the word "control" followed by the printing representation of a control character in double quotation marks (for example, CONTROL "A"); and as the word "space" to specify the space bar. If you specify x and y to be the same, or if you omit y, you get a toggle effect. You can specify CTRL/S and CTRL/Q as x and y parameters, respectively, only on local terminals. (Network terminal connections do not allow for CTRL/S and CTRL/Q.) But even some local terminals require that you select

characters other than CTRL/S and CTRL/Q, for example, the VT125 and the VT100 with the printer port option.

The default values for x and y are CTRL/S and CTRL/Q for local terminals, and CTRL/A/CTRL/A for network terminals. You can achieve consistency between local and network terminals by placing the same TERMINAL PAUSE CHARACTER command in your LOGIN.CMD files on the various TOPS-20 systems.

To set the page length, use the TERMINAL LENGTH command. If you set the page length to 0, the system stops sending output only when you type CTRL/S or the x parameter of the CHARACTER argument.

Default - COMMAND (for all terminal types)
- END-OF-PAGE (for display terminals, for example, VT05, VT50, VT52, VT100)
- CHARACTER (for all terminal types)

RAISE

instructs the system to interpret all lowercase terminal input as the corresponding upper characters. (This setting converts the tilde (~) and right brace (}) to the <ESC> key.)

Default

RECEIVE

same as the RECEIVE command. NO RECEIVE is the same as the REFUSE command.

SPEED n1 n2

sets the baud rate at which the TOPS-20 monitor receives characters from your terminal (n1) and sends characters to your terminal (n2). Be sure also to set any corresponding switch physically located on your terminal.

Default n1 - 300
n2 - n1

TABS

informs the system that your terminal has mechanical tab stops. Causes the TAB key to advance the cursor according to the tab stops on your terminal. (Some terminals let you select tab stops while others have tab stops

every eight spaces). If NO TABS is set, the system simulates a tab by printing eight spaces.

TYPE n — instructs the system to treat your terminal as terminal type n, in accordance with the table below:

Terminal Type	Characteristics
0	Model 33
1	Model 35
2	Model 37
3	EXECUPORT and TI
4-7	reserved for customer use
8	TERMINET
9	IDEAL (has a TAB and FORMFEED mechanism, prints lower case, has infinite line width and infinite page length)
10	VT05
11	VT50
12	LA30
13	VT52, except for not having tabs, and having a page length of 30; used for a Digital Equipment Corporation GT40.
14	LA36
15	VT52
16	VT100
17	LA38
18	LA120
19-34	reserved for customer use
35	VT125
36	VK100

Default - 8

WIDTH n — tells the system the width, in number of characters, of your terminal line. When the system prints a line longer than your terminal width, it prints the first n positions and advances a line to print the rest.

Default width - 72

Type Arguments

33 — informs the system that your terminal is a

Teletype Model 33, which

- does not have a form feed or tab mechanism
- prints lowercase letters as uppercase
- needs extra time to print tabs and certain paper-moving characters (form feed and vertical tab)
- has a line width of 72
- has a page length of 66

35 informs the system that your terminal is a Teletype Model 35, which has the same characteristics as a Model 33, except that it has a form feed and tab mechanism.

37 informs the system that your terminal is a Teletype Model 37, which has the same characteristics as a Model 33, except that it prints lowercase letters.

EXECUPORT informs the system that your terminal is an EXECUPORT, which

- does not have a form feed or tab mechanism
- prints lowercase letters
- needs extra time to perform a carriage return
- has a line width of 80
- has a page length of 66

H19 informs the system that your terminal is a Heath Kit H19 terminal. The system assumes the same characteristics as for the VT52.

LA30 informs the system that your terminal is a Digital Equipment Corporation LA30, which

- does not have a form feed or tab mechanism

- prints lowercase letters as uppercase
- needs extra time to perform a carriage return, line feed, tab, and form feed
- has a line width of 80
- has a page length of 66

LA36 informs the system that your terminal is a Digital Equipment Corporation LA36, which

- does not have a form feed or tab mechanism
- prints lowercase letters
- has a line width of 132
- has a page length of 66

LA38 informs the system that your terminal is a Digital Equipment Corporation LA38, which

- does not have a form-feed mechanism
- prints lowercase letters
- has a line width of 132
- has a page length of 66

LA120 informs the system that your terminal is a Digital Equipment Corporation LA120, which

- prints lowercase letters
- has a line width of 132
- has a page length of 66

SYSTEM-DEFAULT informs the system that your terminal has these characteristics (ensuring an acceptable minimum level of performance for all terminal types):

- does not have a form feed or tab mechanism prints lowercase letters
- needs extra time to perform a carriage return, line feed, tab, and form feed

- has a line width of 72
- has a page length of 66

Default for terminal type

TERMINET — informs the system that your terminal is a TERMINET, which

- does not have a form feed or tab mechanism
- prints lowercase letters
- needs extra time to perform a carriage return, line feed, tab, and form feed
- has a line width of 72
- has a page length of 66

TI — informs the system that your terminal is a Texas Instruments terminal, which has the same characteristics as an EXECUPORT.

VK100 — informs the system that your terminal is a Digital Equipment VK100 with the same characteristics as the VT52 and VT100, plus graphics capability (both black-and-white and color).

VT05 — informs the system that your terminal is a Digital Equipment Corporation VT05, which

- does not have a form-feed mechanism
- has a tab mechanism
- prints lowercase letters as uppercase
- needs extra time to perform a linefeed or formfeed
- has a line width of 72
- has a page length of 20

VT50 — informs the system that your terminal is a Digital Equipment Corporation VT50, which

- does not have a form feed mechanism

- prints lowercase letters as uppercase
- has a line width of 80
- has a page length of 12

VT52 informs the system that your terminal is a Digital Equipment Corporation VT52. The system assumes the same characteristics as for a VT50 except that it prints lowercase letters, and has a page length of 24 lines instead of 12.

VT100 informs the system that your terminal is a Digital Equipment Corporation VT100. The system assumes the same characteristics as for a VT52.

VT102 informs the system that your terminal is a Digital Equipment Corporation VT102. The system assumes the same characteristics as for a VT100.

VT125 informs the system that your terminal is a Digital Equipment Corporation VT125, which has full compatibility with the VT100 and the capability of business, laboratory, and scientific graphics in black-and-white or color.

VT131 informs the system that your terminal is a Digital Equipment Corporation VT131.

VT200-SERIES informs the system that your terminal is a Digital Equipment Corporation VT220, VT240, or VT241. The system assumes the same characteristics as a VT100.

VT300-SERIES informs the system that your terminal is a Digital Equipment Corporation VT330 or VT340. The system assumes the same characteristics as a VT100.

Characteristics

TERMINAL Commands Before Log-in

You can use TERMINAL commands, after an initial CTRL/C or RETURN but before logging in, to adjust your terminal's characteristics.

Hints

Setting Your Terminal's Speed

If the initial speed setting of your terminal line is not what you want but your terminal will function at that speed, you can give a TERMINAL SPEED command even before log-in to set the proper value. If your terminal will not work at the initial speed, ask the operator to set an appropriate value.

Using Split Speeds

If you have a terminal that allows split speeds, you can set the input and output speeds to different values. This will allow you to take advantage of fast system response, for example, without providing a needlessly fast input line. A setting of 150 2400 will accomplish this. Note that you cannot use split speeds on a terminal that is part of a DECSYSTEM-2020 system. Note also that using split speeds on VT100, VT125, or VK100 terminals may cause the "smooth scrolling" feature to function improperly. See the appropriate terminal manual, for example, the VT100 User's Guide, for details.

Special Cases

Terminal Types and Defaults Peculiar to Your System

The preceding pages describe terminal types and system defaults as they are shipped with TOPS-20. However, by making changes to the monitor and the TOPS-20 command processor, your installation can add different terminal types and change the default characteristics associated with terminals. Check with your system manager to find out what changes, if any, are in effect for your system.

Terminal Speed Retained from Last Session

Although most terminal characteristics revert to default settings when you log in, the terminal line will retain the value for speed set by the last user of the line, even if he was using a different kind of terminal. However, if the system failed and was restarted after the terminal line was last used, the initial speed will be determined by the appropriate TERMINAL SPEED command in the system configuration file. Also, dial-up lines return to the speed specified in this file after every use.

Restrictions

CTRL/S and CTRL/Q Not Passed to Remote Nodes

CTRL/S and CTRL/Q are always processed by your host node; they are not sent to a remote node. Therefore, when you are connected to a remote node with the SETHOST program and TERMINAL PAUSE (ON) END-OF-PAGE is set on the remote node, CTRL/Q will not continue scrolling. CTRL/A is the default control character recognized by the remote node for pausing and continuing scrolling. You can use the TERMINAL PAUSE (ON) CHARACTER command to specify the pause and continue characters of your choosing - except CTRL/S and CTRL/Q. It is recommended that you define the same pause and continue scrolling characters on your host and the remote node.

Disabling CTRL/S and CTRL/Q on High Speed Terminals

Some terminal models, when set to a high receive baud rate, such as 9600, require that the CTRL/S and CTRL/Q pause and continue characters be enabled in order to correctly format terminal output. If you must disable CTRL/S and CTRL/Q with the TERMINAL NO PAUSE COMMAND, manually set the terminal to fast or "jump" scroll. If the output is still not correctly formatted, set a slower receive baud rate with the TERMINAL SPEED command. Then, manually set the same baud rate on the terminal.

Warning

Setting an Improper Terminal Speed

If you set an incorrect speed for your terminal, for example, one that is too high, you will be unable to use it further. A TERMINAL SPEED command in the LOGIN.CMD file in your log-in directory can cause the same problem. In such a case, obtain your terminal line number if possible (the second column of SYSTAT command output consists of line numbers) and ask the operator to set an appropriate value.

Related Commands

INFORMATION TERMINAL-MODE	for examining your current terminal settings

Examples

1. Declare that your terminal is an VT100.

   ```
   @TERMINAL VT100
   ```

2. Do the same thing, using the corresponding numerical type.

   ```
   @TERMINAL TYPE 16
   ```

3. Prepare your LA36 terminal for you to type in some upper- and lowercase text files on narrow paper.

   ```
   @TERMINAL LA36
   @TERMINAL NO RAISE
   @terminal width 72
   ```

4. Find out your terminal's characteristics, then give the command that causes it to print a full page of blank lines when you type a CTRL/L (or when it encounters an ^L in a file it is printing on your terminal).

   ```
   @INFORMATION TERMINAL-MODE
    TERMINAL LA36
    TERMINAL SPEED 300
    TERMINAL NO INHIBIT (NON-JOB OUTPUT)
    RECEIVE LINKS
    REFUSE ADVICE
    RECEIVE SYSTEM-MESSAGES
    RECEIVE USER-MESSAGES
    TERMINAL PAUSE (ON) COMMAND
    TERMINAL NO PAUSE (ON) END-OF-PAGE
    TERMINAL LENGTH 66
    TERMINAL WIDTH 132
    TERMINAL LOWERCASE
    TERMINAL RAISE
    TERMINAL NO FLAG
    TERMINAL INDICATE
    TERMINAL NO FORMFEED
    TERMINAL NO TABS
    TERMINAL NO IMMEDIATE
    TERMINAL FULLDUPLEX
   @TERMINAL NO INDICATE
   ```

2.81 TRANSLATE

Displays the project-programmer number corresponding to a directory name, or the directory name corresponding to a project-programmer number.

Format

@TRANSLATE (DIRECTORY) **dev:<directory>**

or

@TRANSLATE (DIRECTORY) **dev:[project-programmer number]**

where:

dev:<directory>	is the name of the directory, enclosed in angle brackets, that you want translated. **Default** dev: - your connected structure
dev:[project-programmer number]	is the project-programmer number, enclosed in square brackets, that you want translated. **Default** dev: - your connected structure

Hints

Using Project-programmer Numbers

Use project-programmer numbers instead of directory names when giving file specifications to programs written for the TOPS-10 operating system. These include the assembler MACRO; the FORTRAN, COBOL, and ALGOL compilers; the linking loader LINK; and utility programs CREF (providing cross-reference information) and FILCOM (for comparing files).

If you are unsure whether a system program requires project-programmer numbers, load it into memory (using the R command), give a CTRL/C to return to TOPS-20 command level, and then examine memory with the INFORMATION MEMORY-USAGE command. If the file PA1050.EXE (the TOPS-10 compatibility package) is present in memory, then the program was

originally written for TOPS-10 and may require a project-programmer number where you would ordinarily give a directory name.

Avoiding Project-programmer Numbers

To avoid project-programmer numbers, define a logical name (of 6 or fewer characters) as the directory in question. Then use this logical name in place of the directory when giving file specifications. The system program will accept the logical name as a device name, and will then be using the correct directory.

Related Commands

DEFINE	for defining a logical name as a directory, to avoid using a project-programmer number

Examples

1. Find out the project-programmer number associated with your connected directory.

```
@TRANSLATE <LATTA>
PS:<LATTA> (IS) PS:[4,261]
```

2. Verify that the project-programmer number reported in Example 1 does correspond to your directory on PS:.

```
@TRANSLATE PS:[4,261]
PS:[4,261] (IS) PS:<LATTA>
```

2.82 TYPE

Displays the contents of one or more files on your terminal.

Format

@TYPE (FILE) **filespec,...,**
@@subcommand

where:

filespec — is the specification of the file you want to display on your terminal.

@@subcommand — means that after a final comma you can enter the following optional subcommand:

UNFORMATTED — Disables the formatting of control characters contained in the file. Normally, the TYPE command displays the graphic equivalent for certain control characters, for example, ESCAPE as $ and CTRL/C as ^C. The UNFORMATTED subcommand causes characters to be displayed literally, allowing graphic text (for example REGIS files) to be displayed on the terminal.

Output

Entire Contents of Files

In response to the TYPE command the system prints the entire contents of a file (up to the EOF (end-of-file) pointer), including blank lines and line numbers if there are any. If you specify more than one file, the filespec precedes the contents of each file.

Hints

Stopping TYPE Output

To stop the TYPE command, type two CTRL/Cs. A CTRL/O will also stop the output, but will not stop the processing of the command or the accumulation of CPU charges. Note that a

pair of CTRL/Os causes the system to skip over part of the output and continue printing.

Related Commands

COPY	for copying files to any device
EDIT	for examining specific parts of a file
PERUSE	for editing files in read-only mode
PRINT	for printing files on the line printer

Example

1. Have the system print a file on your terminal.

```
@TYPE TEST.TXT
! This is file TEST.TXT !
```

2.83 UNATTACH

Disengages another job from its terminal.

Format

@UNATTACH (USER) **name** (JOB #) **number**
PASSWORD:**password**

where:

name — is the user name of the job's owner.

number — is the job number.
Default number – the only job, or only job besides your current (attached) job, logged in under the user name you give.

password — is the associated password (not requested if you are currently logged in under the same user name as the job that you are disengaging).

Characteristics

Log-in Not Necessary

You do not have to be logged in to give the UNATTACH command.

Hints

Freeing Hung Terminals

The UNATTACH command is useful for freeing a terminal that, because of program or hardware errors, is no longer under control of the user. The command UNATTACH n can be more effective than LOGOUT n for this purpose.

Effect on Memory and Terminal

The UNATTACH command does not affect memory and leaves your own terminal at TOPS-20 command level. The other job is left in its current state (usually suspended) and the disengaged terminal is left in the state before log-in.

Related Commands

ADVISE	for sending commands to another job
ATTACH	for joining another job to your terminal
DETACH	for disengaging your own job from its terminal

Examples

1. Disengage another user's job from its terminal.

```
@UNATTACH KANE
Password:___
```

2. From a terminal on which you have not yet logged in, give the UNATTACH command to disengage your only logged-in job from its terminal.

```
 BOSTON (KL2871) Development system, TOPS-20 Monitor 7(21722)
@UNATTACH LATTA
Password:___
```

3. Give a SYSTAT command to find out what jobs you have running. Give the UNATTACH command for two of them (you must specify a job number for the first one so the system will know which one you mean), and check them with another SYSTAT command.

```
@SYSTAT LATTA
  28    26  EXEC     LATTA
  36*  230  EXEC     LATTA
  40    27  EXEC     LATTA
@UNATTACH LATTA 28
 [Attached to TTY26, confirm]
@UNATTACH LATTA
@SYSTAT LATTA
  28   DET  EXEC     LATTA
  36*  230  EXEC     LATTA
  40   DET  EXEC     LATTA
```

2.84 UNDELETE

Restores deleted files.

Format

@**UNDELETE** (FILES) **filespec,...**

where:

filespec — is the specification of the file you want to restore.
Default .gen - all generations of the specified files

Restrictions

Erasure of Deleted Files

Ordinarily an UNDELETE command given during the same terminal session as an original deletion will recover the deleted files, unless you included the EXPUNGE subcommand to DELETE or gave a subsequent EXPUNGE command. However, if any user or a batch job logs out while connect to your directory, all deleted files are permanently erased. Also, if available disk space is low on the system, the operator or the system itself may expunge all deleted files. A system warning message is usually sent before this happens.

Special Cases

Restoring Files Deleted With CONTENTS-ONLY Subcommand.

Any files deleted by a DELETE command with a CONTENTS-ONLY subcommand are immediately expunged. You must use the RETRIEVE command to restore these to disk.

Related Commands

DELETE	for deleting files
DIRECTORY-CLASS commands, with the DELETED subcommand	For obtaining lists of deleted files
EXPUNGE	for permanently erasing deleted files

RETRIEVE for restoring off-line files to disk

Examples

1. Undelete a file.

```
@UNDELETE TEST.FIL
 TEST.FIL.1 [OK]
 TEST.FIL.2 [OK]
```

2. Access another user's directory, then restore all his deleted files of type .FIL.

```
@ACCESS <CARSON>
Password:___
@UNDELETE <CARSON>*.FIL
 <CARSON>MEMO.FIL.1 [OK]
 <CARSON>SEARCH.FIL.1 [OK]
 <CARSON>TEST.FIL.1 [OK]
 <CARSON>VERFY.FIL.1 [OK]
@END-ACCESS <CARSON>
```

2.85 UNKEEP

Cancels the kept status of a fork.

Format

UNKEEP (FORK) **fork**

where:

fork is one of the following: Fork name
Fork number
Default - the current fork

Characteristics

Unkept Forks

An unkept fork is a fork that is cleared from memory when another program is loaded or when the RESET command is given. Forks are normally unkept unless kept with the KEEP or SET PROGRAM KEEP commands.

Inferior Forks

The UNKEEP command simultaneously cancels the kept status of a superior fork and its inferior forks.

Hints

More Information

The UNKEEP command is one of the TOPS-20 multiforking-class commands. For more information about multiforking, see the section named Running Multiple Programs in the TOPS-20 User's Guide.

Effect on Memory

The UNKEEP command does not immediately affect memory. It does, however, allow a fork in memory to be cleared when another program is loaded or the RESET command is given.

Related Commands

INFORMATION FORK-STATUS for displaying the fork status

KEEP	for changing an unkept fork to a kept fork
RESET	for clearing forks from memory
CONTINUE, FORK, FREEZE, KEEP, INFORMATION FORK-STATUS, SET NAME, and SET PROGRAM	other multiforking-related commands

Examples

1. Give the INFORMATION FORK-STATUS command to display the fork status. Then, give the UNKEEP command to cancel the kept status of the current fork and redisplay the fork status.

```
@INFORMATION FORK-STATUS
 => EDIT (1): Kept, HALT at 6254, 0:00:00.6
    FILCOM (2): Kept, ^C from IO wait at 700272, 0:00:00.2
@UNKEEP
@INFORMATION FORK-STATUS
 => EDIT (1): HALT at 6254, 0:00:00.6
    FILCOM (2): Kept, ^C from IO wait at 700272, 0:00:00.2
```

2. Display the fork status, and UNKEEP a noncurrent fork. Then, verify the new fork status.

```
@INFORMATION FORK-STATUS
 => EDIT (1): HALT at 6254, 0:00:00.6
    FILCOM (2): Kept, ^C from IO wait at 700272, 0:00:00.2
@UNKEEP FILCOM
@INFORMATION FORK-STATUS
 => EDIT (1): HALT at 6254, 0:00:00.6
    FILCOM (2): ^C from IO wait at 700272, 0:00:00.2
```

2.86 UNLOAD

Rewinds a magnetic tape until it is returned completely to the source reel, and puts the associated tape drive offline. Use UNLOAD only for tapes mounted on drives having device names of the form MTAn:.

Format

@UNLOAD (DEVICE) **dev:**

where:

dev: is the name of the magnetic tape drive that you want to unload.

Restrictions

UNLOAD With Open Files

If you have given a CTRL/C to exit from a program that has opened a magnetic tape drive and you then gave the UNLOAD command for that tape drive, the system will first ask if you want to close the associated file. You must do so for UNLOAD to succeed, but you will probably be unable to continue the program from that point because the file will now be closed.

UNLOAD Not for MOUNTed Drives

Use the UNLOAD command for tape drives obtained with the ASSIGN command. Use DISMOUNT for a tape drive obtained with MOUNT.

Warning

Cannot Access Tape Again

The UNLOAD command makes it impossible to access your tape again unless it is reloaded by the operator.

Related Commands

ASSIGN for assigning a tape drive to your job

DISMOUNT for unloading tapes mounted on devices of the form MTn:

REWIND for rewinding a magnetic tape volume or tape set to its load point (logical beginning)

Example

1. Unload your magnetic tape from drive MTA0:.

   @UNLOAD MTA0:

2.87 VDIRECTORY

The VDIRECTORY (Verbose DIRECTORY) command is equivalent to the DIRECTORY command with the subcommands LENGTH, NO HEADING, PROTECTION, SIZE, and TIMES (AND DATES OF) WRITE. Use the same format and subcommands with VDIRECTORY as with DIRECTORY. For further information see the DIRECTORY command description in this manual.

When used with magnetic tapes, the VDIRECTORY command is equivalent to the DIRECTORY command for magnetic tapes.

Examples

1. Give the VDIRECTORY command, then cut off the output with a CTRL/C.

```
@VDIRECTORY

  MISC:<LATTA>
4-UPED.TXT.14;P777700     0 0(7)          25-Apr-85 09:58:21
A.DIRECTORY.1;P20200      1 0(0)           2-May-85 13:14:52
ARTIFI.CTL.7;P777700      1 215(7)        24-Apr-85 10:10:10
B.DIRECTORY.1;P20200      1 0(0)           9-May-85 12:54:00
C.EXE.1;P777700           3 1536(36)      13-Apr-85 04:27:59
CONFAB.CTL.1;P777700      1 115(7)         3-May-85 13:34:37
DIVIDE.FOR.4;P777700      1 260(7)         8-Mar-85 15:47:41
DUMPER.MAC.1;P777700     53 134442(7)      8-Nov-85 10:47:04
MAGNIF.CTL.2;P777700      1^C
```

2. Ask for a VDIRECTORY listing of certain files; include a line of headings.

```
@VDIRECTORY TEST.FOR,
@@HEADING
@@

                           PGS Bytes(SZ)    Write


  MISC:<LATTA>
TESTF1.FOR.8;P777700      1 115(7)        25-Apr-85 09:44:50
TESTF2.FOR.1;P777700      1 115(7)        20-Apr-85 10:01:56
TESTF3.FOR.1;P777700      1 115(7)        20-Apr-85 10:02:19

Total of 3 pages in 3 file
```

APPENDIX A

FUNCTIONAL GROUPING OF TOPS-20 COMMANDS

This appendix lists and briefly explains all non-privileged commands in the TOPS-20 command language, grouping them in categories of similar use.

A.1 SYSTEM ACCESS COMMANDS

These commands allow you to gain and relinquish access to the system, to activate and deactivate any special capabilities you have been given, and to disengage and engage jobs to your terminal.

ATTACH	Engages a designated job to your terminal.
DETACH	Disengages your current job from your terminal.
DISABLE	Deactivates any special capabilities you have been granted.
ENABLE	Activates any special capabilities you have been granted.
LOGIN	Gains access to the TOPS-20 system.
LOGOUT	Relinquishes access to the TOPS-20 system.
UNATTACH	Disengages another job from its terminal.

A.2 FILE SYSTEM COMMANDS

The file system commands allow you to create, examine, change, and delete files.

ACCESS	Obtains ownership rights to the specified directory, as well as the group rights of the

	directory's owner.
APPEND	Adds contents of one or more files to another file.
ARCHIVE	Makes a permanent off-line copy of files.
BUILD	Creates, modifies, or deletes a subdirectory.
CLOSE	Closes files left open by a program, and releases unopened JFNs.
CONNECT	Connects you to the specified directory.
COPY	Duplicates files.
CREATE	Invokes your defined editor to create a file.
DEFINE	Associates a logical name with one or more filespecs.
DELETE	Marks files for eventual erasure (disk files only), or erases the files (all other devices).
DIRECTORY	Gives information about the files in a directory.
DISCARD	Gives up the tape copy of specified on-line files.
EDIT	Invokes your defined editor to modify a file.
END-ACCESS	Relinquishes ownership rights to the specified directory.
EXPUNGE	Permanently erases any deleted files.
FDIRECTORY	Lists all the information about files.
PERUSE	Edits files in read-only mode.
RENAME	Changes one or more parts of an existing file specification.
RETRIEVE	Restores off-line files to disk.
TDIRECTORY	Lists the names and write dates of files in the order of the date and time they were last changed.
TYPE	Prints files on your terminal.
UNDELETE	Restores files marked for erasure.

VDIRECTORY	Lists the names of files, as well as their protection, size, and the date and time they were last changed.

A.3 DEVICE-HANDLING COMMANDS

These commands allow you to reserve a device prior to using it, to manipulate the device, and to release it once it is no longer needed.

ASSIGN	Reserves a device for use by your job.
BACKSPACE	Moves a magnetic tape backward.
DEASSIGN	Releases a previously assigned device.
DISMOUNT	Gives up access to the specified structure or tape set.
EOF	Writes an end-of-file mark on a magnetic tape.
MOUNT	Requests use of the specified structure or tape set.
REWIND	Moves a magnetic tape backward to its load point.
SKIP	Moves a magnetic tape forward.
UNLOAD	Rewinds a magnetic tape until the tape is wound completely on the source reel.

A.4 PROGRAM CONTROL COMMANDS

The following commands help you run and debug your own programs.

COMPILE	Translates a source program using the appropriate compiler.
CONTINUE	Resumes execution of a program (e.g., one interrupted by a CTRL/C).
CREF	Runs the CREF program, which produces a cross-reference listing and automatically sends it to the line printer.
CSAVE	Saves in a compressed executable format the program currently in memory. (Usually SAVE is better for most purposes.)

DDT	Merges the loaded debugging program (or if none, DDT) with the current program and then starts the debugging program.
DEBUG	Compiles a source program, loads it with a compatible debugging program, and starts the debugging program.
DEPOSIT	Sets the contents of the specified memory location.
ERUN	Runs a system program in an ephemeral fork.
EXAMINE	Checks the contents of the specified memory location.
EXECUTE	Compiles, loads, and begins execution of a program.
FORK	Selects the current fork to which TOPS-20 commands apply.
FREEZE	Stops a running fork.
GET	Places an executable program in memory.
KEEP	Protects a fork from being cleared from memory.
LOAD	Compiles a program and loads it into memory.
MERGE	Places an executable program in memory and merges it with the current contents of memory.
POP	Finishes a level of TOPS-20 and returns control to the previous level of TOPS-20.
PUSH	Starts a new level of TOPS-20.
R	Runs a system program.
REENTER	Starts the program currently in memory at the alternate entry point specified in the program's entry vector.
RESET	Clears memory for the specified fork of your job and its inferiors.
RUN	Places an executable program in memory and starts it.
SAVE	Copies the contents of memory into a file in

	executable format.
START	Begins execution of the program in memory.
UNKEEP	Cancels the kept status of a fork.

A.5 INFORMATION COMMANDS

These commands return information about TOPS-20 commands, your job, and the system as a whole.

DAYTIME	Prints the current date and time of day.
HELP	Gives an explanatory message about specific system programs.
INFORMATION	Provides information about your job and its use of available computing resources, and about the system.
SYSTAT	Gives a summary of information about current jobs on the system.
TRANSLATE	Tells you what project-programmer number is associated with a directory name, and vice versa.

A.6 TERMINAL COMMANDS

The terminal commands allow you to declare the characteristics of your terminal, to clear your video screen, and to control linking to another user's terminal.

ADVISE	Sends whatever you type on your terminal as input to a job engaged to another terminal.
BLANK	Clears your display screen and moves the cursor to line 1.
BREAK	Clears communication links.
RECEIVE	Allows your terminal to receive communication links, advice, or system messages from other users.
REFUSE	Denies links, advice, or system messages to your terminal.

!.pg

REMARK	Notifies the system that your terminal is not sending commands but only comments.
SET	Establishes certain job-wide characteristics for your terminal session.
SET HOST	Connects the terminal to another system.
SEND	Sends messages to terminals.
TAKE	Executes commands contained in the specified file.
TALK	Links two terminals so that each user can observe what the other user is doing, without affecting his job.
TERMINAL	Informs the system of your terminal type, and lets you determine the setting of its variable parameters.

A.7 OUTPUT COMMANDS

These commands allow you to request output listings of files on the plotter, line printer, card punch, or paper tape punch, and to examine, modify, or withdraw these requests.

CANCEL	Withdraws requests from an output queue (waiting list).
INFORMATION OUTPUT-REQUESTS	Lists entries in the output queues.
MODIFY	Changes entries in an output queue.
PLOT	Places requests in a plotter output queue.
PRINT	Places requests in a line printer output queue.
PUNCH	Places requests in a card punch or paper tape punch output queue.

A.8 BATCH COMMANDS

The TOPS-20 system also has a batch system to which you can submit jobs for execution.

CANCEL BATCH	Withdraws entries from the batch input queue (waiting list).
INFORMATION BATCH-REQUESTS	Lists entries in the batch input queue.
MODIFY BATCH	Changes entries in the batch input queue.
SUBMIT	Places a batch control file in the batch input queue.

APPENDIX B

ALPHABETICAL LIST OF TOPS-20 COMMANDS

This appendix consists of an alphabetical list of TOPS-20 commands and short descriptions of each. The varieties of argument needed to complete each command are shown next, in the order that you give them; parentheses indicate that the argument is optional or can be defaulted. Last comes the effect of the command on memory - whether it clears memory (by loading a program or by other means) or otherwise alters it, or leaves it unaffected. Use this appendix along with the question mark and recognition features to refresh your memory once you have become familiar with the contents of this manual.

Command	Function	Arguments	Effect on Memory
ACCESS	gets ownership rights to a directory	dev:<directory>, password	---
ADVISE	sends commands to another user's job	user name or terminal number	---
APPEND	adds files onto end of another file	filespecs, (subcommands)	---
ARCHIVE	requests off-line storage of files	filespecs, (subcommand)	---
ASSIGN	allocates a device for your job	device name	---
ATTACH	engages a job to your terminal	user name, (job number) (password)	---
BACKSPACE	moves a magnetic tape backward	device name, number of records or files	---

ALPHABETICAL LIST OF TOPS-20 COMMANDS

BLANK	clears your video terminal screen	---	---
BREAK	ends links made by a TALK command	(user name or line number)	---
BUILD	creates, modifies, or deletes a subdirectory	dev:<directory>, subcommands	---
CANCEL	withdraws output or batch requests	queue, jobname	---
CLOSE	closes open files	JFN	---
COMPILE	translates source programs into object programs	filespecs, switches	loads compiler
CONNECT	connects your job to a directory	dev:<directory>, password	---
CONTINUE	continues a halted program	NORMALLY or STAY	---
COPY	makes copies of a file	filespec,filespec (subcommands)	---
CREATE	creates a file	(switches), filespec	loads EDIT
CREF	translates .CRF files into listings	(filespec)	loads CREF
CSAVE	stores a copy of memory in a file (in compressed format)	(filespec, memory locations)	---
DAYTIME	tells the date and time	---	---
DDT	starts a debugging program	(switches)	merges debugging program with existing memory (if necessary)
DEASSIGN	gives up a previously assigned device	device name	---

DEBUG	debugs a program	filespec, switches	loads program and debugging program
DEFINE	establishes or withdraws a logical name	logical name, search list	---
DELETE	marks files for later erasure	filespecs, (subcommands)	---
DEPOSIT	changes contents of a memory location	memory location, octal number	changes one location
DETACH	disengages a job from your terminal	(argument)	---
DIRECTORY	gives information about files	(dev:<directory>, filespecs), subcommands	---
DISABLE	deactivates capabilities	---	---
DISCARD	gives up tape copy of on-line files	filespecs	---
DISMOUNT	gives up access to structure or tape set	alias or setname, switches	---
EDIT	edits existing files	(switches), filespec	loads EDIT
ENABLE	activates capabilities	---	---
END-ACCESS	terminates ownership rights to a directory	dev:<directory>	---
EOF	writes an end-of-file mark on a magnetic tape	device name	---
ERUN	runs a system program in an ephemeral fork	name of system program (switches)	loads program
EXAMINE	inspects a memory	memory location	---

	location		
EXECUTE	compiles, loads and starts a program	filespec, switches	loads compiler and/or program
EXPUNGE	erases all deleted files from a directory	(dev:<directory>, subcommands)	---
FDIRECTORY	DIRECTORY command with subcommands CRAM,EVERYTHING, and NO HEADING	(dev:<directory>, filespecs), subcommands	---
FORK	specifies what fork is current	fork name or number	---
FREEZE	stops a running fork	fork name or number	---
GET	places an executable program in memory	filespec, (switches)	loads program
HELP	presents a short description of a system program	name of system program	---
INFORMATION	gives information about system and job parameters	arguments	---
KEEP	protects a fork from being reset	fork name or number	---
LOAD	compiles and loads a program	filespec, switches	loads compiler and/or program
LOGIN	begins a job	user name, password, account	---
LOGOUT	ends a job	---	clears memory
MERGE	merges an executable program with current memory	filespec, (switches)	adds program to

			existing memory
MODIFY	changes output or batch requests	queue, jobname, switches	---
MOUNT	requests use of structure or tape set	alias or setname, switches	---
PERUSE	edits files in read-only mode	(switches), filespec	loads editor
PLOT	plots files on plotter	filespecs, switches	---
POP	returns to superior TOPS-20 command level, ending inferior TOPS-20 command level	---	clears memory of inferior process
PRINT	prints files on line printer	filespecs, switches	---
PUNCH	punches files on card punch or paper tape punch	filespecs, switches	---
PUSH	begins an inferior TOPS-20 command level	---	preserves superior memory
R	runs a system program	name of system program, (switches)	loads program
RECEIVE	allows communication with your terminal	argument	---
REENTER	starts program in memory at the restart address	---	---
REFUSE	disallows communication with your terminal	argument	---
REMARK	informs the system that you are typing only comments, not commands	---	---
RENAME	changes the	filespec, filespec	---

	specification of a file		
RESET	clears the specified fork from memory	fork name or number	clears fork
RETRIEVE	restores off-line files to disk	filespecs	---
REWIND	rewinds a magnetic tape to its load point	device name	---
RUN	places an executable program in memory and starts it	filespec, (switches)	loads program
SAVE	stores a copy of memory in a file	(filespec)	---
SEND	sends messages to terminals	line number	---
SET	sets various job parameters	arguments	---
SET HOST	connects the terminal to another system	system name, switches	
SKIP	moves a magnetic tape forward	device name, number of records or files	---
START	starts program in memory at start address	(memory location)	---
SUBMIT	submits entries (jobs) to the batch system	filespecs, switches	---
SYSTAT	gives information about system and job status	arguments, (subcommands)	---
TAKE	starts processing of a command file	filespec, (subcommands)	---
TALK	makes communication links with another user	user name or terminal number	---

TDIRECTORY	DIRECTORY command with subcommands CHRONOLOGICAL WRITE, REVERSE, and TIMES WRITE	(dev:<directory>, filespecs), subcommands	---
TERMINAL	sets various terminal characteristics	argument	---
TRANSLATE	gives directory names for ppn's and vice versa	dev:<directory> or [project-programmer number]	---
TYPE	prints files on your terminal	filespecs	---
UNATTACH	disengages another job from its terminal	user name, (job number) (password)	---
UNDELETE	restores deleted files	filespec	---
UNKEEP	cancels the kept status of a fork	fork name or number	---
UNLOAD	unloads a magnetic tape and deassigns the drive	device name	---
VDIRECTORY	DIRECTORY command with subcommands LENGTH, NO HEADING, PROTECTION, SIZE, and TIMES WRITE	(dev:<directory>, filespecs), subcommands	---

APPENDIX C

FILE ATTRIBUTES

This appendix lists the attributes that you can include in a file specification. The DECnet-related attributes are described in detail in the DECnet documentation. Information on file attributes is also contained in the TOPS-20 User's Guide, the TOPS-20 Monitor Calls Reference Manual, and the TOPS-20 Tape Processing Manual.

;A:account	System manager defined account string
;BDATA:user data	DECnet optional binary data
;BLOCK-LENGTH:n	Maximum size of a physical block on a labeled tape.
;BPASSWORD:password	DECnet binary password
;CHARGE:account	DECnet account string
;DATA:user data	DECnet optional data
;EXPIRATION-DATE:date	Date the magnetic tape file can be overwritten
;FORMAT:F,D,S, or U	Magnetic labeled tape record format: F=fixed-length D=variable-length S=spanned U=undefined
;OFF-LINE	Designation for a file that is off-line because of migration or archiving
;P:n	File protection value
;PASSWORD:password	DECnet password
;POSITION:n	File sequence number for magnetic tape

	positioning
;RECORD-LENGTH:n	Maximum size of a magnetic labeled tape record
;T	Designation for a file that is to be deleted at log-out time (a temporary file). Only for files in log-in and/or connected directories.
;USERID:id	DECnet user ID string

Note that you can issue TOPS-20 commands as an alternative to specifying many of the file attributes. For example, SET FILE PROTECTION, rather than the P attribute, can be used to set the file protection. Also, such commands as INFORMATION FILE-STATUS and DIRECTORY (with the EVERYTHING subcommand) display your file attribute settings.

APPENDIX D

CONTROL CHARACTERS

This appendix lists the TOPS-20 control characters. To type a control character, use the CTRL key like the SHIFT key. Hold down the CTRL key and at the same time type the character key.

CTRL CHARACTER	FUNCTION
CTRL/C	Cancels a command when typed during command entry. Cancels command execution when typed twice during command execution. (The first ^C does not echo.) Halts a program and returns to TOPS-20 command level. Generally, type two CTRL/Cs to halt a program that is executing; type one CTRL/C to stop a program that is waiting for input.
CTRL/E	Ends an ADVISE command link between two terminals.
CTRL/F	Provides recognition input for file specifications and command names and arguments. Similar to the ESCAPE key, except that it completes only one field at a time with file specifications and it does not supply guidewords with commands.
CTRL/G	Rings the terminal bell. Use with SEND and TALK commands to signal another user.
CTRL/H	Reprints a command line up to the field that is in error. Type immediately after the input error message. Duplicates the function of the BACKSPACE key.
CTRL/I	Duplicates the function of the TAB key.
CTRL/L	Advances the paper on a hard-copy terminal to the top of the next page. To stop the ^L from printing, give the TERMINAL NO INDICATE command.

CTRL/O Stops output to the terminal while the program or command continues to execute.

Output resumes when:
- another CTRL/O is typed
- the program or command finishes
- the program prompts for input

CTRL/Q Continues scrolling terminal output that was paused by CTRL/S or a pause on an end of page. Use the TERMINAL PAUSE (ON) CHARACTER command to define alternate pause and continue characters. Use the TERMINAL NO PAUSE (ON) END-OF-PAGE command to stop pause on an end of page. Use the TERMINAL LENGTH command to set the page length.

CTRL/R Reprints the current command line, incorporating the corrections made with the DELETE key or CTRL/W. Use on hard-copy terminals when backslashes and underscores caused by editing with DELETE and CTRL/W make the command line difficult to read. Use on video terminals when a system message covers the command line.

CTRL/S Pauses scrolling of terminal output until CTRL/Q is typed. Input, though not displayed, is accepted into the input buffer and processed after a CTRL/Q. Use the TERMINAL PAUSE (ON) CHARACTER command to define alternate pause and continue characters.

CTRL/T Displays a line of information that includes:
- the time
- the name and status of the current fork
- the amount of CPU time used
- the time elapsed since login
- the system's load average

CTRL/U Erases the current command line. On hard-copy terminals, cancels the command and prints three Xs at the end of the command line.

CTRL/V Allows special characters (any character other than an alphanumeric, or the special characters - _ . or $) in file specifications and directory names. Type CTRL/V before typing the special character any time you type the file specification.

Special characters are not accepted by the following system programs:

ALGOL	COBOL	CREF	FILCOM	ISAM
LIBARY	LINK	MAKLIB	RERUN	

CTRL/W Erases the previous word. On hard-copy terminals, cancels

the previous word and types an underscore.

CTRL/Z Signals an end-of-file for data entered from the terminal. Use when:

- COPYing from your terminal (device TTY:) to a file.
- Sending a message with one of the mail programs.

-A-

-B-

-C-

-D-

-E-

-F-

-N-

-O-

-P-

-S-

-T-

-U-

-V-

-W-

-X-