SECTION 3

ACCOUNTING COMMANDS

This section contains descriptions of the Multics commands used by accounting administrators. These commands are defined within an exec_com (master.ec) and are designed with special error checking facilities. Most of them prompt the accounting administrator for input. They are intended to be used from within the restricted environment of an accounting administrator and they assume that the accounting administrator's working directory is:

```
>udd>SysAdmin>admin
```

Each description contains the name of the command (including the abbreviated form, if any), discusses the purpose of the command, and shows the correct usage. Notes and examples are included when deemed necessary for clarity.

A system administrator can also use the accounting administrator commands.

To use these commands as an accounting administrator, the system administrator must change his working directory to >udd>SysAdmin>admin and precede each command with the string "ec master" as:

```
ec master command argument1 ... argumentn
```

For convenience, the system administrator may wish to set up an abbreviation to provide the "ec master" string (e.g., ".ab x ec master").

Many of the accounting administrator commands in the master.ec segment execute more than one command program (in addition to checking arguments). For example, the accounting administrator command edit_proj executes three commands: edit_proj, daily_summary, and install (to install the changed SAT). However, if the system administrator types "edit_proj" rather than "ec master edit_proj," only the edit_proj command is executed. If the administator then types "install smf.cur.sat" rather than "ec master install smf.cur.sat," the SAT is installed but the daily_summary command is not executed. This omission means that any changes to a project's cutoff limits, that were just made, are ignored.

These two examples illustrate the different results when a command is invoked by a system administrator as opposed to an accounting administrator. A system administrator who intends to perform accounting administrator functions from his own process should be aware of the differences between these commands. He should consistently use only the system administrator version or only the accounting administrator version of each command to minimize the possibilities of error.

Error messages produced by command programs called from within master.ec are prefixed by the name of the command. If an error is detected by the exec_com portion of the environment, the error message is produced by a call to err.ec (also in the >udd>SysAdmin>lib directory) and has this general form:

```
--> ERROR User Jones is not registered
```

The error-handling exec_com then signals the program_interrupt condition to return control to the user. Since the system administrator normally does not have a handler for the program_interrupt condition, the signalling of this condition by the exec_com results in a new command level being established (ready messages indicate level 2 or higher). The correct response to this error sequence is to issue the release command before proceeding with further commands.

A lock is kept in the sys_admin_data segment (in >udd>SysAdmin>lib) to prevent more than one system administrator from editing the data bases at the same time. If the segment is locked, the error handling exec_com prints an error message and signals a program interrupt. This lock mechanism is another reason why the master.ec should be used whenever possible by system administrators (it is always used by the restricted environment of accounting administrators.)

Name: add_anon

SYNTAX AS A COMMAND

```
add_anon Project_id initproc homedir {password}
```

FUNCTION

adds an anonymous user to an undelegated project; the accounting administrator assigns this user a home directory and an initial procedure and can also assign a password if he wishes.

ARGUMENTS

Project_id
is the name of the project under which the anonymous user is to be registered.

initproc
is the name of the process overseer.

homedir
is the pathname of the anonymous user's home directory.

password
is the password assigned to the anonymous user.

EXAMPLES

To add an anonymous user (who does not have a password) to the Proj5 project, type:

```
!  add_anon Proj5 process_overseer_ >udd>Proj5
   Warning: anonymous user has no password
   r 15:57 1.372 102

   From Initializer.SysDaemon.z (install)  1602.0:
   installed Proj5.pdt for SA1.SysAdmin
```

To add an anonymous user, with the 3-character password cjt, to the Proj8 project, type:

```
!  add_anon Proj8 process_overseer_ >udd>Proj8 cjt
   r 15:57 1.372 102

   From Initializer.SysDaemon.z (install)  1557.0:
   install Proj8.pdt for SA1.SysAdmin
```

bill

Name: bill

SYNTAX AS A COMMAND

```
bill function {args}
```

FUNCTION

calls the biller.ec segment (in the directory >udd>SysAdmin>lib) to perform billing operations.

ARGUMENTS

function
is one of the following:

prepare
prepares the billing data bases for billing.

run
runs the billing programs

accept
prints the bills, cleans up the old billing data bases, and creates new one.

delete
deletes the current months bills from the storage system.

CONTROL ARGUMENTS

are arguments accepted by the particular billing function.

EXAMPLES

To prepare for billing, type:

```
! bill prepare
```

To run the bills, type:

```
! bill run MM DD YY
  or
! bill run MM DD YY arg
```

where MM DD YY is the date on which the billing is being run and arg is an argument accepted by a command to output billing in a site-dependent format.

To accept a bill, type:

```
! bill accept arg
```

where arg is a month, a Julian date, or any name the uniquely identifies the billing run.

To clean up after the bills are printed, type:

```
! bill delete
```

Name: chaddr

SYNTAX AS A COMMAND

```
chaddr Person_id {address}
```

FUNCTION

changes the address of one user.

ARGUMENTS

Person_id
: is the Person_id of the user whose address is to be changed.

address
: is the new address of the user. It must be enclosed in quotes if it contains blanks or other special characters. If this argument is omitted, the chaddr command prints the user's old address and then waits for the accounting administrator to either enter the new address (which should not be enlosed in quotes in this case) or enter a null line (to retain the old address).

EXAMPLES

To change the address for user Jones to MS 486, type:

```
! chaddr Jones "MS 486"
  r 1557 1.372 1.258 102
```

To make the same change, but first check on the old address, type:

```
! chaddr Jones
  MS 203
! MS 486
  r 1557 1.372 1.258 102
```

Name: chalias

SYNTAX AS A COMMAND

```
chalias Person_id {alias}
```

FUNCTION

gives a user a login alias or changes his existing alias. The alias can be a maximum of eight characters; the first character must be a lowercase letter. Only one alias per user is allowed. An alias is particularly useful for a user with a long or complex name.

ARGUMENTS

Person_id
: is the Person_id of the user whose login alias is to be changed.

alias
: is the new login alias of the user. If this argument is omitted, the chalias command prints the user's old alias and waits for the accounting administrator to either enter the new one or enter a null line (to retain the old alias).

EXAMPLES

To change user Jones' alias to "waj", type:

```
!  chalias Jones waj
   r 15:57 1.372 102
```

Name: change

SYNTAX AS A COMMAND

```
change {Person_id} {item}} {new_value}
```

FUNCTION

review and possibly modify user registration data in the URF and PNT. Each item is typed out. The administrator can leave the item unchanged by typing a carriage return, or he can replace the item by typing new data.

ARGUMENTS

Person_id
: is the Person_id of a user whose registration data is to be changed. If it is omitted, the change command asks for it.

item
is the item to be changed. If it is omitted, every item is displayed and a new value is requested. Item can only be given if Person_id is given. Item may be any one of the following keywords:

addr
User's mailing address

cpass
card input password

name
full name (Last First I.: title)

notes
any miscellaneous information

proj
default project

pass
login password

progn
programmer number

new_value
is the new value for item. If it is omitted, the old value is displayed and a new value is requested.

NOTES

The chaddr, chalias, chcpass, chdf_proj, chname, chpass, and chprog commands can be used to change some of the individual items in the PNT and avoid the need to type the item name or to be prompted for new values of all items.

The user's last name can be a maximum of 32 characters. The field for the user's first name and middle initial can be a maximum of 24 characters. The user's name should be given last name first, then a comma, the first name, a space, and then middle initial followed by a period. Names of the form "Smith, J. Alfred" are also allowed. If the user has a title (e.g., Prof., Capt., Dr.), the title (maximum of 8 characters) should follow the name and be separated from the name by a colon, for example:

```
Smith,John J.:Prof.
```

The address field can be a maximum of 32 characters; slashes can be used to separate lines of an address.

The programmer number and default project ID can be a maximum of 32 and 9 characters, respectively.

Notes can be a maximum of 32 characters. This field can hold miscellaneous information such as phone number, additional address information, position within the organization, etc.

The password and card input password can be from one through eight ASCII printing characters, including backspace, but excluding space and semicolon. "HELP", "help", "quit", and "?" are interpreted uniquely by the password processor and are therefore unacceptable as password specifications for an interactive login. A response of either a blank line or an asterisk ("*") will leave the password unchanged.

The accounting administrator can type "stop" at any time to abort all the changes he has made for a particular user and start over.

EXAMPLES

In the following example, address and default project are to be changed for user Smith. An exclamation point precedes entries typed by the administrator; an exclamation point followed by nothing indicates a carriage return.

```
!  change Smith
   Full name:                *Smith, John
   Full name                 !
   Address:                  MS 149
   Address                  ! MS 204
   Prog. number:             1234
   Prog. number              !
   Notes:           Tel.  324-9261
   Notes                      !
   Project ID:               Quark
   Project ID               ! Physics
   Password:                  !
   Card Input Password:      !

   r 10.07 1.486 34
```

The following example shows how the administrator would change a single item, in this case, the telephone number maintained in the notes item for user Smith. If "notes" were omitted, each item would be displayed, and a new value requested, as above.

```
!  change Smith notes
   Notes:                        Tel.  324-9261
   Notes                     !   Tel.  324-9200

   r 10:07 1.486 34
```

Name: charge

SYNTAX AS A COMMAND

```
charge
```

FUNCTION

enters miscellaneous charges into the miscfile. For each transaction, the Project_id, the amount, and an explanation are required. All three input items for a transaction can be put on the same line or they can be supplied one at a time. To exit from this command, type "x" instead of a Project_id. The miscfile segment is printed using the pmisc command. Credits and deletions are made to the miscfile using the credit and dmisc commands, respectively.

EXAMPLES

To charge the Alpha project for some manuals, type:

```
!  charge
   project
!  Alpha
   amt
!  10.55
   explanation
!  manuals ordered 6/23 Jones
   project
!  x
   r 1557 1.372 1.258 102
```

Name: chcpass

SYNTAX AS A COMMAND

```
chcpass Person_id
```

FUNCTION

changes the user's card input password in the PNT found in >sc1. Thus, even if the user has changed his password with the -cpw control argument on the ++PASSWORD card, the chcpass command overrides the user's change.

A card input password may consist of from one through eight ASCII printing characters, including backspace, but excluding space and semicolon.

where Person_id is the Person_id of the user whose card input password is to be changed.

EXAMPLES

To change the card input password for user Jones, type:

```
! chcpass Jones
  Password
! cardpass    (printer is turned off)
  Password again:
! cardpass    (printer is turned off)
  r 15:45 64:979 5097
```

Name: chdf_proj

SYNTAX AS A COMMAND

```
chdf_proj Person_id Project_id
```

FUNCTION

changes the default project for a single user. It changes the user's default project in the PNT found in >sc1. Thus, even if the user has changed his default project with the -change_default_project control argument to the login command, the chdf_proj command overrides the user's change.

ARGUMENTS

Person_id
is the Person_id of the user whose default project is to be changed.

Project_id
is the new default project for the user.

EXAMPLES

To give user Smith a default project of Proj3, type:

```
! chdf_proj Smith Proj3
  r 1557 1.372 1.258 102
```

Name: check_log

SYNTAX AS A COMMAND

```
check_log Person_id
```

FUNCTION

scans the log segment for initializer messages. Sometimes a user is unable to log in; if so, the initializer places a message in the log.

ARGUMENTS

Person_id
is the Person_id of the user whose log entry is to be checked.

LIST OF MESSAGES

The following is a list of the messages and explanations for refusing login.

bad_pass
bad password.

badpers
person not in PNT, either not added or mistyped.

no_name
no name given.

pwlocked
password locked by administrator.

bad_proj
Project_id typed by user does not exist in SAT.

no_pdt
project'pdt/s pdt ipdt/s not in >pdt/sc1>pdt; mepdt/spdt/sage with pdt/starpdt/s alpdt/so typed online.

not_pdt
user not in PDT for project; the user is not registered on the project.

anon_pw
bad anonymous-user password.

already
user already logged in and does not have the "multip" attribute in the PMF (absentee does not count).

absdate
user absolute cutoff date (set by project administrator) exceeded.

limitu
user dollar limit (set by project administrator) exceeded.

mnthlim
user per-month dollar limit (set by project administrator) exceeded

shiftlim
user per-shift dollar limit (set by project administrator) exceeded.

abslimit
user absolute cutoff dollar limit (set by project administrator) exceeded.

hd_make
user home directory does not exist and cannot be created.

sys_full
system full according to load units.

saturate
system full according to maximum number of users.

cant_bum
system full, user cannot find anyone to bump.

groupmax
load control group at absolute maximum.

no_group
load control group in SAT not in master_group_table segment (in the udd>SysAdmin>admin directory).

nf_nosec
system not full, group full, no secondary.

sysgrpfl
system full, group full, cannot bump.

projothr
project cut off for some other reason.

prog_err
programming error in answering service (lg_ctl_).

EXAMPLES

To see why user Jones cannot log in, type:

```
! check_log Jones
  LOGIN Denied  Jones.Multics  a.h013  int (bad_pass)
```

* This message says that the user could not log in because he gave the wrong password.

Name: chname

SYNTAX AS A COMMAND

```
chname Person_id {mailing_name}
```

FUNCTION

changes the mailing name for one user. The full name must be enclosed in quotes if it is given on the command line. If the new name is not given, the old name is displayed and a change is accepted.

ARGUMENTS

Person_id
: is the Person_id of the user whose mailing name is to be changed.

mailing_name
: is the new mailing name of the user. It must be enclosed in quotes. If this argument is omitted, the chname command prints the old mailing name and then waits for the accounting administrator to enter the user's new mailing name (which should not be enclosed in quotes in this case) or enter a null line (to retain the old value).

EXAMPLES

To change the mailing name for user Jones, type:

```
!  chname Jones "Jones,W. Alfred"
   r 1557 1.372 1.258 102
```

To make the same change, but first check the old mailing name, type:

```
!  chname Jones
   Jones, Alfred
!  Jones,W. Alfred
   r 1558 1.261 1.114 97
```

Name: chpass

SYNTAX AS A COMMAND

```
chpass Person_id
```

FUNCTION

changes the password for a single user. It changes the user's password in the PNT found in >sc1. Thus, even if the user has changed his password with the -change_password control argument to the login command, the chpass command overrides the user's change.

ARGUMENTS

Person_id
is the Person_id of the user whose password is to be changed.

NOTES

A password may consist of from one through eight ASCII printing characters including backspace, but excluding space and semicolon.

"HELP", "help", "quit", and "?" are interpreted uniquely by the password processor and are therefore unacceptable as password specifications for an interactive login. Entering "quit" terminates the login attempt, while "HELP", "help", or "?" results in an explanatory message and repeat of the password prompt.

EXAMPLES

To change the password for user Jones, type:

```
! chpass Jones
  Password
! comein        (printer is turned off)
  Password again:
! comein        (printer is turned off)
  r 15:45 64.979 5097
```

Name: chprog

SYNTAX AS A COMMAND

```
chprog Person_id {prog_no}
```

FUNCTION

changes the programmer number for one user. The programmer number must be all numeric. It can be up to sixteen characters long. (Some installations use the employee number assigned by the company in this field.) If the new programmer number is not supplied, the old value is displayed and a change is accepted.

ARGUMENTS

Person_id
: is the Person_id of the user whose programmer number is to be changed.

prog_no
: is the new programmer number. If this argument is omitted, the chprog command prints the old programmer number and then waits for the accounting administrator to either enter the new programmer number of a null line (to retain the old number).

EXAMPLES

To change the programmer number for user Smith, type:

```
!  chprog Smith 7399
   r 1557 1.372 1.258 102
```

To make the same change, but first print the old programmer number, type:

```
!  chprog Smith
   7299
!  7399
   r 1558 1.261 1.114 97
```

Name: credit

SYNTAX AS A COMMAND

```
credit
```

FUNCTION

enters miscellaneous credits into the miscfile. For each transaction, the Project_id, the amount, and an explanation are required. All three input items for a transaction can be put on the same line or they can be supplied one at a time. To exit from this command, type "x" instead of a Project_id. The miscfile segment is printed using the pmisc command. Charges and deletions are made to the miscfile using the charge and dmisc commands, respectively.

EXAMPLES

To credit the Alpha project for a crash, type:

```
!  credit
   project
!  Alpha
   amt
!  23.00
   explanation
!  system crash 6/23 Smith
   project
!  x
   r 1557 1.372 1.258 102
```

Name: cu

SYNTAX AS A COMMAND

```
cu Person_id Project_id
```

FUNCTION

creates a user's home directory. Home directories are created automatically when a PDT containing a new user is installed. In rare cases, it may be necessary to execute this function separately.

ARGUMENTS

Person_id
: is the Person_id of the user whose home directory is to be created.

Project_id
is the Project_id of the project under which the user's home directory is created.

EXAMPLES

To create the home directory >udd>Gamma>Jones, type:

```
! cu Jones Gamma
  r 1557 1.372 1.258 102
```

Name: day

SYNTAX AS A COMMAND

```
day
```

FUNCTION

prints the output from the crank on the terminal and asks whether or not to delete the output. (Every night, a self-rescheduling absentee job called the "crank" is run. It records the daily charges of users and checks for users who should be cut off.) Unless there was an error, the absentee output segment should be deleted.

EXAMPLES

The output from the crank has this general appearance:

```
! day

  Absentee user Accountant.SysAdmin logged in ...
  r 0330 1.372 1.258 102
  Begin charging for 7/31/84 2355.0 to 8/9/84 2345.1
  cut 3, warned 7, total charge $45678.90
  r 1557 1.372 1.258 102

  Absentee user Accountant.SysAdmin logged out ...
  Delete?
! yes
```

Name: delegate

SYNTAX AS A COMMAND

```
delegate Project_id path User_id
```

FUNCTION

gives a project administrator control over a project's PMF. Once this is done, the project administrator can add and delete users and set resource-usage limits on users in the project. There may be up to four project administrators for the project.

The delegate command edits the SAT to show that a project administrator can install PDTs for a particular project. It also sets access on the proj_admin_seg segment (in the >sc1 directory) and the >sc1>update directory so the project administrator can install his PDT.

A delegated project is under control of the project administrator only. The accounting administrator receives an error message if he attempts to execute the add_anon, dpmf, upmf, or pmf commands on a delegated project. The accounting administrator can use the undelegate command if he wishes to take a project back from a project administrator, in order to have the project under his control, or to delegate it to some new project administrator.

ARGUMENTS

Project_id
: is the Project_id of the project to be assigned a project administrator.

path
: is the pathname of the directory in which the PMF is placed. This pathname is usually the pathname of the project directory.

User_id
: is the User_id, specified as Person_id.Project_id, of the project administrator.

EXAMPLES

To give Jones.Gamma control of the Beta project (placing the PMF in >udd>Beta), type:

```
!  delegate Beta >udd>Beta Jones.Gamma
   archive: Beta appended to delegated_pmf.archive
   r 15:57 1.372 102

   From Initializer.SysDaemon.z (install)  1430.0:
   installed sat for SA1.SysAdmin
```

Name: disk_report, drp, disk_auto

SYNTAX AS A COMMAND

```
drp
```

FUNCTION

calculates disk usage and creates a disk usage report. The administrator can invoke the disk_report command to cause a manual disk usage calculation. Normally, though, disk usage is calculated automatically every night by the absentee job, dodrp.absin, that executes the disk_auto command.

NOTES

The sweep program is used to get quota for all directories into the disk_stat data segment (in >udd>SysAdmin>admin). The charge_disk program then charges these usage figures to projects in their "projfile" entries. A printable segment, called diskreport, is produced but not automatically printed. To print copies of this segment, type "rqm diskreport". See the rqm command.

This command examines a large fraction of all the directories in the hierarchy, placing a heavy load on the system. Its use during peak load hours should be avoided if possible.

EXAMPLES

To run a disk report, type:

```
!  disk_report
   $ Creating disk usage report.
   $ Following figure is total quota/current use
   75500/64432
   dir:  5500/4432
   seg: 70000/60000
   Charged 906 directories out of 910 to 108 projects
   r 1557 1.372 1.258 102
```

Name: dmisc

SYNTAX AS A COMMAND

```
dmisc
```

FUNCTION

deletes charges from the miscfile. For each transaction, the Project_id and the number of the miscfile entry are required. The number of the miscfile entry is printed using the pmisc command. Both input items for a transaction can be put on the same line or they can be supplied one at a time. To exit from this command, type "x" instead of a Project_id. Charges and credits are made to the miscfile using the charge and credit commands, respectively.

EXAMPLES

```
!  dmisc
   project
!  Alpha
   number
!  23
   project
!  x
   r 1557 1.372 1.258 102
```

Name: dpmf

SYNTAX AS A COMMAND

```
dpmf Project_id Person_id
```

FUNCTION

deletes a user from a PMF for an undelegated project that is managed by an accounting administrator and then installs the new PDT.

ARGUMENTS

Project_id
: is the Project_id of the project from which a user is to be deleted.

Person_id
: is the Person_id of the user to be deleted from the project.

EXAMPLES

To delete user Black from the Gamma project, type:

```
!  dpmf Gamma Black
   r 1557 1.372 1.258 102
```

Name: dproj

SYNTAX AS A COMMAND

```
dproj Project_id
```

FUNCTION

deletes a project. It edits the SAT to remove the project entry, sets the date off for the project, and deletes the project directory and all of its contents.

ARGUMENTS

Project_id
    is the Project_id of the project to be deleted.

EXAMPLES

To delete the Delta project, type:

```
!  dproj Delta

   quota   used   directory name
   1000     5      >udd>Delta
    250     2      >udd>Delta>Person_id1
              . . .
   1250     7      Total
   delete_dir: do you want to delete the directory >udd>Delta??
!  yes
   r 15:57 1.372 102

   From Initializer.SysDaemon.z (install)  1558.0:
   install sat for SA1.SysAdmin
```

If the accounting administrator answers "yes" to the question about deleting the directory, the project directory and all segments and directories inferior to the project directory are deleted. The project is charged for disk usage until the project directory is deleted.

If the accounting administrator answers "no" to this question, the project's directory and segments are not deleted, and the project continues to accumulate storage charges. The accounting administrator should not answer "no" unless there is some exceptional reason for doing so; if he does, the project directory has to be deleted later by a system administrator.

Name: edit_proj

SYNTAX AS A COMMAND

```
edit_proj Project
or
edit_proj project keyword
or
edit_proj project keyword old_value
or
edit_proj project keyword new_value
or
edit_proj project keyword old_value new_value
```

FUNCTION

changes project registration information.

The command can be invoked in several ways as described below.

1. The command can be invoked in the format "edit_proj project," in which case you are queried for changes to each keyword.

2. The command can be invoked in the format "edit_proj project keyword," in which case you are queried for a change only to the specified keyword.

3. The command can be invoked in the format "edit_proj project keyword old_value." In this case, "keyword" must be either "administrator" or "group" and "old_value" must be one of the multiple values associated with "administrator" or "group." This invocation permits you to be queried for a change to the specified value associated with the "administrator" or "group" keyword.

4. the command can be invoked in the format "edit_proj project keyword new_value," in which case the keyword value is changed to new_value.

5. The command can be invoked in the format "edit_proj project keyword old_value new_value," in which case old_value is changed to new_value. In this case, keyword must be either "administrator" or "group."

ARGUMENTS

project
is the project_id of the project whose registration data items are to be edited. If this is the only argument specified, edit_proj prints each data item one at a time and waits for a response from the accounting administrator before proceeding. The accounting administrator may respond with any one of the following:

carriage return
to leave the item unchanged.

a new value
to replace the printed value.

stop
to exit immediately from the edit_proj command without making any changes.

keyword
is the particular data item to be changed. The valid keywords are:

title	t
investigator	inv
investigator_address	inv_addr
supervisor	sup
supervisor_address	sup_addr
supervisor_phone	sup_phone
account	acct
requisition	req
requisition_amount	amt
cutoff_date	cutoff
billing_name	billto
billing_address	billat
group	grp
attributes	attr
grace	gr
administrator	admin
quota	q
dir_quota	dq
alias	aka
groups	grps
min_ring	min
max_ring	max
max_foreground	maxfg
max_background	maxbg
abs_foreground_cpu_limit	absfgcpulim
pdir_quota	pdq
rate_structure	rs
accounting_category	acct_cat
authorization	authorization
audit	audit

old_value
is one of the current values of the administrator keyword or the group keyword.

new_value
is the new value of the data item identified by the specified keyword.

NOTES

If the accounting administator requests a change to either the requisition or account number of the project, he is asked a question to which he must respond with one of the following:

drop
to cause the charges to the old account and requisition to be eliminated.

transfer
to cause the charges to be transferred from the old account and requisition to the new account and requisition.

bill
to cause the charges to be billed to the old account and requisition, and the new account and requisition to start off with a clean slate.

reset
to cause the changes to the old account to be reset.

For example:

```
!  edit_proj States req P05566-J
   What is the disposition of charges of
   $1233.79 to account 70906, req P03344-J?
!  bill
   r 19:39 2.661 197
```

Name: install

SYNTAX AS A COMMAND

```
install path
```

FUNCTION

installs a system table. Many of the other commands described in this manual that modify system tables also automatically install them. Therefore, this command should only be used after a command whose documentation indicates that a system table is modified but not installed or when an operation that usually installs a system table is interrupted by a system failure before it can do so.

ARGUMENTS

path
is the pathname of the system table to be installed or is one of the following keywords, sat or smf.cur.sat. These keywords invoke the daily_summary command before the SAT is installed.

If any other pathname is given, the specified system table is installed as described in the privileged Multics install command found in Section 2 of this manual.

EXAMPLES

To install the current SAT, type:

```
! install smf.cur.sat
  r 15:57 1.372 102

  From Initializer.SysDaemon.z (install)  1605.0:
  installed sat for SA1.SysAdmin
```

Name: ison

SYNTAX AS A COMMAND

```
ison name
```

FUNCTION

prints "true" or "false" depending on whether or not a person is registered in the PNT. It then lists all users in the URF who have a last name that matches the name argument.

ARGUMENTS

name
is the last name of the person who may be registered on the system.

EXAMPLES

To check whether user Jones is registered, type:

```
! ison Jones
  true
  Personid for "Jones, Herbert R." is "HJones"
  Personid for "Jones, Peter" is "Jones"
  Personid for "Jones, W. Alfred" is "AJones"
  Number of users with last name "Jones" is 3
  r 1557 1.372 1.258 102
```

Name: new_proj

SYNTAX AS A COMMAND

```
new_proj Project_id
```

FUNCTION

requests all project registration data, including the initial list of users, and installs the new PDT and SAT. The new_proj command exits immediately and stops whatever it is doing when the accounting administrator types "stop" in response to any question. At the end of input, the accounting administrator is asked if he wants to review the data;
* if he replies "yes", all of the project registration data are printed out.

ARGUMENTS

Project_id
: is the Project_id of the project to be registered. The Project_id must be from one to nine characters long, must begin with a capital letter or a digit, and must be unique at the site.

NOTES

If the system crashes during the new_proj command, the project may have been incompletely added to the system. If this happens, the accounting administrator MUST NOT try to add the project again from the beginning. The help of a system administrator is necessary to complete the adding of the project. All terminal output should be saved and no other project registrations attempted until a system administrator has checked and corrected the files.

EXAMPLES

To add a new project named Gamma, type:

```
! new_proj Gamma
```

First, the system asks for the project title. This field should be a short description of the purpose of the project. It can be up to 52 characters long.

```
   Title:
!  Gamma Ray Research
```

Next, the system asks for the name and address of the principal investigator. This is the individual in the management structure who is responsible for the project. Both name and address can be up to 32 characters long.

```
   Investigator:
!  Albert W.  Jones
   Address:
!  MS 310
```

The system next asks for the name, address, and telephone number of the project supervisor. This is the individual in direct contact with the project's day-to-day activities. The detailed user usage report, produced by monthly billing, is sent to this person. He can be the same person as the principal investigator. (Typing "=" in response to the request for the supervisor's name causes the principal investigator's name and address to be used.) Often, the supervisor is registered as a user or project administrator for the project. The name and address can each be up to 32 characters long. The telephone number can be up to 16 characters long.

```
   Supervisor:
!  Mr.  Melvin Fooch
   Address:
!  MS 350
   Phone:
!  x1234
```

At this point, the system asks for the external account number and the requisition number. Each of these "numbers" can be a maximum of 12 characters.

```
   Account:
!  11792x
   Requisition:
!  AB123456
```

The system next asks for the cutoff limits for the project. The funds limit is a dollar amount (enter "open" or "O" to show an open amount). The date cutoff limit is a date, expressed as mm/dd/yy. If the project exceeds either of these limits, all users on the project are prevented from logging in, but the project continues to incur disk and registration charges until the project is deleted. (The funds limit is stored internally as a floating-point value.)

```
   Amount:
!  open
   Cutoff date:
!  7/30/85
```

Next, the system asks for the name and address to which bills for the project's usage should be sent. The name and address can each be up to 32 characters long. Typing "=" in response to the request for the billing name causes the principal investigator's name and address to be used.

```
   Billing name:
!  Fiscal Office, L.  Spottswood
   Address:
!  MS 501
```

The administrator is next asked to supply an alternate name (usually a short name) for the project. For example, the alternate name of the Accounting project could be specified as accnt.

```
   Alias:
!  accnt
```

The accounting administrator is now asked if the project is to be delegated and, if so, to supply the name(s) of the project administrator (s).

```
   Is this project delegated?
!  yes
   Enter administrator IDs (Person_id.Project_id).  Type "." to exit.
   Administrator ID:
!  Benway.Gamma
   Administrator ID:
!  .
```

Next, the system asks for the absentee foreground cpu limit. This value is a decimal integer specifying (in seconds) a CPU time limit for foreground absentee jobs. A value of zero means no limit.

```
   Absentee foreground cpu limit:
!  0
```

The system then asks for a range of AIM authorizations. The range of authorizations must be in the form low_auth : high_auth and indicates the range of authorizations at which users on the project can log in. The default value is "system_low."

```
   Authorization:
!  system_low:L6,C1,C2,C3,C4,C5,C6
```

Next, the system asks for the audit flags for the project. The list of audit flags is supplied below. The default value is that no access auditing takes place.

priv_op
controls auditing of privileged operations performed by the process. A privileged operation is one performed through use of a privileged gate or under previously set AIM privileges. It is recommended that sites interested in auditing turn this flag on for all processes except perhaps the system daemons.

admin_op
controls auditing of administrative operations performed by the process. This includes such operations as registration of new users or projects. It is recommended that sites interested in auditing should turn this flag on for all processes.

fault
controls auditing of illegal procedure and access violation faults that can indicate an attempt to access protected data.

small_cc
controls auditing of covert channel activity that takes place over channels with a potential bandwidth of 1-10 bps.

moderate_cc
controls auditing of covert channel activity that takes place over channels with a potential bandwidth of 10-100 bps.

<object_type>=<grant_level>/<deny_level>
controls the auditing of specified operations on specified system objects. The values of <object_type> can be one of the following:

fsobj
specifies that operations to file system objects are to be audited.

fsattr
specifies that operations to file system attributes are to be audited.

rcp
specifies that operations to objects controlled by the Resource Control Package are to be audited.

admin
specifies that operations to administrative objects (e.g., the PNT) are to be audited.

special
specifies that operations to special objects are to be audited. (Currently, the only special objects are processes.)

other
specifies that operations to objects (e.g., mailboxes) controlled by ring 1 security related subsystems are to be audited.

The values that can be assigned to <grant_level> and <deny_level> are listed below.

N
specifies that no auditing is to take place.

MA
specifies that"modify access" operations are to be audited. Operations are audited that attempt to change the access attributes of the object.

M
specifies that "modify" operations are to be audited. Operations are audited that attempt to change the object or the attributes of the object. This level of auditing includes the "modify access" operations.

R
specifies that "read" operations are to be audited. Operations are audited that return information about the contents of the object or its attributes/properties. This level of auditing includes the "modify" and "modify access" operations.

The <grant_type>/<deny_type> values are a matched pair. The <grant_type> value specifies auditing of successful operations. The <deny_type> value specifies auditing of unsuccessful operations. For example, the audit flag "fsobj=N/M" specifies that there is to be no monitoring of successful operations on file system objects; however, all unsuccessful modify operations on file system objects will be audited.

Please note that modify access operations cannot be associated with file system objects (fsobj). Instead, modify access operations can be specified for file system attributes (fsattr).

Additional information on auditing, including a more detailed description of the operations that are audited on each object type, can be found in the *Multics System Administration Procedures* manual, Order No. AK50.

```
   Audit:
!  priv_op, small_cc
```

Next, the system request a segment quota value. The segment quota value is a decimal number specifying the number of records of segment quota to be allocated to this project:

```
   Segment quota:
!  100
```

The system next requests a directory quota value. The directory quota value is a decimal number representing the number of records of directory quota which should be allocated to the project directory. The default value is 10% of the segment quota.

```
   Directory quota:
!  10
```

The system next asks for the name of the rate structure to be used for the project. A rate structure is a table created by the system administrator that defines the prices to be charged for resource usage. The rate structure name must be supplied by the system administrator.

```
   Rate structure:
!  Structure1
```

Next, the system asks for the name of the default load control group. A load control group is a group of projects that share certain attributes.

```
   Default group:
!  GroupA
```

The system then asks for the name(s) of all eligible load control groups.

```
   Authorized Groups:
!  GroupA GroupB
```

The system next asks for the list of attributes to be applied to this project. The list of valid attributes is supplied below:

guaranteed_login
: users can use the -force argument to the login command to bypass load control

anonymous
: The project can have anonymous users

preempting or bumping
: permits users to log in by preempting others in the same load control group

brief
: suppresses messages associate with a successful login

nostartup
: users start_up.ec is not executed

nostartup no_warning
: suppresses urgent system warning and emergency messages from the operator

save_on_disconnect, save
: saves users' processes if the process is disconnected because of a communications line hangup or FNP crash

null, none
: can be used as the only value to turn off all default values

nobump
: users are not subject to preemption by anyone

nopreempt
users are not subject to preemption by others in the same load control group

nolist
users are not to be listed in whotab (others may nevertheless be able to deduce that a user having the nolist attribute is logged in)

dialok or dial
users may accept dial requests

multip or multilogin
users can log in more than one process

vinitproc or v_process_overseer
users may specify a process overseer or outer module on the login command line; users may also replace the process overseer, outer module, or other procedures by placing a copy in the home directory.

vhomedir, v_home_dir
user may specify home directory at login

no_secondary, no_sec
user may not have secondary status

no_primary, no_prime
user may not have primary status

op_login, daemon
user may be logged in by operator, via message coordinator

no_warning, _nowarn
user is permanently in no_warning mode (as if -no_warning had been given in login command) and never receives system warning messages

igroup
user is in an individual load control group

save_pdir
save process directory after fatal process error; used for debugging purposes at development sites

disconnect_ok
users may have saved disconnected processes, i.e., user may use the -save argument to the login command.

```
   Attributes:
!  bumping, brief, vinitproc, nostart
```

The system next requests the administrator to specify the grace period. The grace period is the number of minutes after login for which users will be protected from preemption by other users.

```
   Grace:
!  60
```

The system next asks that the administrator specify the minimum ring number (most privileged ring) in which the users can log in.

```
   Minimum login ring:
!  4
```

The system next asks for the maximum ring number (least privileged ring) in which users can log in.

```
   Maximum login ring:
!  5
```

Next, the system asks for the maximum number of records to be used for process directory storage for uers on this project. A value of zero indicates that the system default project directory quota value is used.

```
   Maximum pdir quota:
!  0
```

The system next asks for the maximum number of foreground processes that can be executed on behalf of this project. A foreground process is a process created in response to a user login request or a request in the foreground absentee queue.

```
   Maximum foreground processes:
!  0
```

The system next asks for the maximum number of background processses that can be executed on behalf of this project. A background process is an absentee process created in response to a request in one of the background absentee queues (queues 1 through 4).

```
   Maximum background processes:
!  0
```

If the project has been delegated to a project administrator, the system next asks for the name of the diretory to which the project master file is to be moved.

```
   PMF directory name:
!  >udd>Gamma
```

The system next requests the name of a logical volume on which segments subordinate to the project directory will reside. The default value is that segments will reside on the same volume as those under >udd.

```
   Project directory logical volume:
!  [carriage return for default]
```

If a non-null response was given for the previous prompt, the system then asks for the identity (Person_id.Project_id) of the owner of the master directory and the identity (Person_id.Project_id) of the quota account on the specified logical volume. The default value for the owner of the master directory is the Person_id.Project_id of the current user of the new_proj command. If the quota account is not specified, the system uses an account accessible to the master directory owner. (Type carriage return for default.)

```
   Master directory owner:
!  Smith.Pubs
   Master directory account:
!  [carriage return for default]
```

The system next requests the AIM access class of the project directory. The default value is the access class applied to the >udd directory (normally system_low).

```
   Project directory access class:
!  [carriage return for default]
```

* Finally, the accounting administrator is asked to enter the initial list of users. Notice when Benway is added to the project's user list a warning is returned stating that Benway is not registered in the PNT. Benway remains on the project's user list but is unable to log in until the accounting administrator registers him in the PNT using the register command.

```
*  Enter initial list of users.  Type "." to exit.
   Person:
   Benway
   WARNING: Benway is not registered.
   Person:
   Smith
   Person:
   Jones
   Person:


   Input for "Gamma" complete

   Do you wish to review?
   no
   r 19:37 6.732 669

   From Initializer.SysDaemon.z (install)  1937.0:
   installed sat for SA1.SysAdmin

   From Initializer.SysDaemon.z (install)  1937.0:
   installed Gamma.pdt for SA1.SysAdmin
```

Name: pmf

SYNTAX AS A COMMAND

```
pmf path
```

FUNCTION

edits a PMF (for an undelegated project) with the qedx editor, converts the edited PMF to a PDT, and installs the PDT. It first puts the administrator into the qedx editor after reading in the segment and allows him to make any changes he wants. After the administrator exits from qedx, the command converts the PMF to a PDT and signals the initializer to install the PDT. The command takes care of updating the pmf.archive segment (in the >udd>SysAdmin>admin directory) as well. See the *qedx Text Editor User's Guide* manual, Order No. CG40) for details on use of the qedx command.

ARGUMENTS

path
is the pathname of the PMF to be edited.

EXAMPLES

To edit the segment named Operator.pmf, type:

```
!  pmf Operator
   Edit.
!  (editing commands)
!  w
!  q
   r 15:57 1.372 102

   From Initializer.SysDaemon.z (install) 1558.0:
   installed Operator.pdt for SA1.SysAdmin
```

Name: pmisc

SYNTAX AS A COMMAND

```
pmisc
```

FUNCTION

prints the contents of the miscfile. It is useful as a check to be sure that all entries in the miscfile are correct before the bills are run. The accounting administrator can print the miscellaneous charges and credits for an individual project or he can print the entire miscfile. He can also print the charges and credits for specific dates. The dates must be in the form mm/dd/yy or mm/dd. If more than one date appears on the same line, they must be separated by spaces. Type "x" instead of a Project_id to exit from the command.

NOTES

The pmisc command first asks the accounting administrator to enter the name of the project whose miscfile entries he wants to review. If the accounting administrator types "all" in response to this request, the entries for all projects in the miscfile are printed.

The pmisc command then asks the accounting administrator to enter the dates on which the charges to be reviewed were incurred. If the accounting administrator types "all" in response to this request, all the entries in the miscfile are printed.

EXAMPLES

To print the entries for the Alpha project incurred on July 9 and July 12, type:

```
!  pmisc
   project
!  Alpha
   dates
!  7/9/84 7/12/84
   07/09/84  23  Alpha  4.50  manual
   07/12/84  32  Alpha  7.60  manual
   project
!  x
   r 1557 1.234 1.001 115
```

To print all the entries in the miscfile, type:

```
!  pmisc
   project
!  all
   dates
!  all
   07/01/84 1  Gamma  23.56  manuals
   .
   .
   .
   07/29/84  207  Beta  .50  news bulletin
   r 1557 1.234 1.001 115
```

To print all the entries for the Time project, type:

```
!  pmisc
   project
!  Time
   dates
!  all
   07/02/84  8  Time  9.00  text book
   .
   .
   .
   07/15/84  59  Time  5.00  fine
   project
! x
   r 1557 1.234 1.001.115
```

Name: proj_mtd

SYNTAX AS A COMMAND

```
proj_mtd Project_id
```

FUNCTION

types a month-to-date report for any project's usage. The report lists all users on the project and their dollar totals, as well as disk and miscellaneous charges. This command does not accept additional arguments as the project administrator's proj_usage_report command does. It prints a more complete summary of the project's usage charges because the accounting administrator has access to more accounting data than the project administrator does.

ARGUMENTS

Project_id
is the Project_id of the project whose usage report is to be printed.

EXAMPLES

To get a report of the usage for the SAIL project, type:

```
!  proj_mtd SAIL
   Month to date for proj SAIL

   Name                logins      charge

   White                 11     $ 133.41
   Brown                  0     $   0.00

   2 users               11     $ 133.41

   registration                 $  20.00
   misc                         $   0.00
   disk                         $ 176.08

   Total                        $ 329.49

   r 1557 1.372 1.258 102
```

Name: recov

SYNTAX AS A COMMAND

```
recov Project_id
```

FUNCTION

generates a new PMF from the system's binary PDT; it is used if a project's PMF is destroyed. It is also used to recover a project's PMF when the project is undelegated.

ARGUMENTS

Project_id
is the Project_id of the project whose PMF is to be recovered.

EXAMPLES

Suppose the Proj2 project loses its PMF. To get a new one, type:

```
! recov Proj2
  r 1557 1.372 1.258 102
```

Name: register

SYNTAX AS A COMMAND

```
register
```

FUNCTION

registers a new user. It enters his Person_id in both the URF and PNT segments. If a person is already a user of Multics, or if he was once registered and was not removed, this command should not be used since the user is still in the URF and PNT segments.

The accounting administrator can type "stop" at any time to abort the processing of the current user; for instance, he may want to abort if he misspells the user's last name or if the user is already registered.

EXAMPLES

Refer to the change command in this document for the list of rules about a user's name, password, alias, and programmer number to be followed when registering a new user. To register a user, type:

```
! register

   Enter full name (Last, First I.)
   Full name            !  Smith, Robert M.

   Enter mailing address
   Address              !  CISL

   Enter programmer number
   Prog number          !  2424

   Enter notes
   Notes                !  temporarily assigned to project

   Enter default project
   Project ID           !  Maint

   Password             !  letmein  (printer is turned off)

   Password again:      !  letmein  (printer is turned off)

   Card Input Pasword:  !  cardpass (printer is turned off)

   Password again:      !  cardpass (printer is turned off)
```

Programmer number, notes, default project, and card input password are optional and may be ommitted by typing a carriage return in response to the prompt. At this point, the system attempts to generate a unique Person_id for the person being registered by trying first his last name alone and then his last name prefixed by his initials. If either of these attempts can be used, the system makes a tentative assignment and asks if the Person_id is acceptable. If neither of these attempts can be used (because they are already Person_ids), or if the administrator rejects the system's tentative assignment, the register command asks for a Person_id and then checks to make sure that the identifier entered by the accounting administrator is unique. The administrator may use the ison command before registration to see all Person_ids for the persons last name.

```
   Person_id "Smith" is already used by "Smith, Frank X."
   Trying "RSmith" for Person_id.
   Person_id assigned is "RSmith"
   Is this ok?
!  no
   Please suggest a Person_id for "Smith, Robert M."
!  RMSmith
   Person_id assigned is "RMSmith"
   Is this ok?
!  yes
```

The accounting administrator can add more users at this time, or he can exit from the register command.

```
   More users to add?
!  no
   r 15:57 1.372 102
```

At this point, the user has been added to both the URF and PNT segments.

Name: rename_proj

SYNTAX AS A COMMAND

```
rename_proj Project_id1 Project_id2 {new_project_alias}
```

FUNCTION

is used to rename a project. It changes the project's name in the SAT and installs it; it renames the system copy of the project's PDT and changes the project name stored in that PDT; and it renames the project directory. If the project is not delegated, it renames and edits the PMF and replaces it in the pmf.archive segment, deleting the copy with the old name from the archive.

NOTE: The system administrator must set access on the segments contained in the renamed project directory so that users on the renamed project have access to them.

ARGUMENTS

Project_id1
: is the Project_id of the project whose name is to be changed.

Project_id2
: is the new Project_id of the project.

new_project_alias
: is the new project alias, if it is to be changed. If this argument is omitted, the old project alias, if any, is unchanged.

NOTES

To change the default project for users now on project AAA, the system administrator can use the chdf_proj command (once for each user). If the accounting administrator does not change the users' default projects, users with users' default projects AAA can change the default project to BBB when they log in (by using the -change_default_project control argument to the login command).

EXAMPLES

To rename project AAA to BBB, type:

```
! rename_proj AAA BBB
  r 15:57 1.372 102

  From Initializer.SysDaemon.z (install)  1557.0:
  installed sat for SA1.SysAdmin
```

Name: rqm

SYNTAX AS A COMMAND

```
rqm path
```

FUNCTION

sends printed reports to all administrators.

ARGUMENTS

path
: is the pathname of the segment to be dprinted.

EXAMPLES

To send a copy of the disk usage report to all administrators, type:

```
! rqm diskreport
  r 1557 1.372 1.258 102
```

Name: setcrank

SYNTAX AS A COMMAND

```
setcrank
```

FUNCTION

schedules the absentee job that performs the accounting segment update. Unless the absentee job crashes or the absentee job queues are lost, there is no need to execute this command. To check whether a job is scheduled, invoke the list_absentee_request command (described in the *Multics Commands and Active Functions Manual*, Order No. AG92).

EXAMPLES

If the accounting update absentee job is not scheduled, type:

```
! setcrank
  3 already requested
  r 1557 1.372 1.258 102
```

Name: setdisk

SYNTAX AS A COMMAND

```
setdisk
```

FUNCTION

schedules the absentee job that performs disk reporting. Unless the absentee job crashes or the absentee job queues are lost, there is no need to execute this command. To check whether a job is scheduled, invoke the list_absentee_request command (described in the *Multics Commands and Active Functions* manual, Order No. AG92).

EXAMPLES

If the disk reporting absentee job is not scheduled, type:

```
! setdisk
  4 already requested
  r 1557 1.372 1.258 102
```

Name: undelegate

SYNTAX AS A COMMAND

```
undelegate Project_id
```

FUNCTION

allows the accounting administrator to regain control of a project that was delegated. He can then either control the project himself or delegate it to some other user.

ARGUMENTS

Project_id
    is the Project_id of the project to be undelegated.

EXAMPLES

To have the Alpha project controlled by Jones.Alpha instead of Smith.Alpha, type:

```
!  undelegate Alpha
   archive: Alpha appended to pmf.archive
   r 15:57 1.372 102

   From Initializer.SysDaemon.z (install) 1557.0:
   installed sat for SA1.SysAdmin

!  delegate Alpha >udd>Alpha Jones.Alpha
   archive: Alpha appended to delegated_pmf.archive
   r 16:02 1.896 64

   From Initializer.SysDaemon.z (install)  1602.0:
   installed sat for SA1.SysAdmin
```

Name: upmf

SYNTAX AS A COMMAND

```
upmf Project_id Person_id
```

FUNCTION

adds a user to a project. It is used only on projects that are not delegated. The command extracts the project's PMF from the pmf.archive segment (in the >udd>SysAdmin>admin directory), edits the PMF to add the new entry, converts the PMF to a PDT, signals the initializer to install the PDT, and replaces the edited PMF in pmf.archive. The user is added with no individual attributes, just those of the project. To give individual attributes to the user, use the pmf command.

ARGUMENTS

Project_id
is the Project_id of the project on which a user is to be added.

Person_id
is the Person_id of the user to be added to the project.

EXAMPLES

To add user Jones to the Gamma project, type:

```
! upmf Gamma Jones
  r 15:57 1.372 102

  From Initializer.SysDaemon.z (install)  1557.0:
  installed Gamma.pdt for SA1.SysAdmin
```

Name: who_delg

SYNTAX AS A COMMAND

```
who_delg {Project_id}
```

FUNCTION

used to determine if a project has been delegated. All project administrators are listed for a delegated project.

ARGUMENTS

Project_id
: is the Project_id of the project in question. If this argument is omitted, a list of all delegated projects is printed.

EXAMPLES

To determine the project administrator(s) for project Proj8, type:

```
!  who_delg Proj8

   List of Delegated Projects as of 05/19/84 2223.7

         Project       Administrator

         Proj8         Jones.Proj8

   r 2103 1.005 1.356 112
```

SECTION 4

INITIALIZER COMMANDS

The commands in this section can be used only when communicating with the initializer process. The usual method of communicating with the initializer process is through an initializer terminal. The bootload console is used as the first initializer terminal. When Multics is first booted and the initializer process is active in ring 1, the initializer process uses the bootload console for input/output. Once the initializer process moves to ring 4, the administrator can select another terminal as the initializer terminal. (The administrator can also select multiple terminals to be used as initializer terminals.) If the administrator does not explicitly select an initializer terminal, the bootload console is used as the initializer terminal.

Some of the initializer commands can be used only when the initializer is operating in ring 1; other initializer commands can be used only when the initializer is operating in ring 4; a third group of initializer commands can be used at any time. An indication of when each command can be used is provided in the description of the individual command.

(See Section 1 for a description of how to use initializer commands when not at an initializer terminal; see also Section 1 for a description of how to submit a non-initializer command to the initializer process.)

abs

Name: abs

SYNTAX AS A COMMAND

```
abs keyword {args}
```

FUNCTION

controls the absentee facility. The absentee facility is designed to operate automatically, without any user intervention. Every use of the abs command overrides some parameter or control algorithm that would otherwise have been used automatically. Thus the abs command should only be used in unusual circumstances, or at the direction of the system administrator. This command can be used only in ring 4.

ARGUMENTS

keyword
: must be chosen from the keywords listed under "Keyword Summary" below.

args
: may be either job selection arguments, described under "Job Selection" below, or other arguments specific to each keyword, or both. Exact usage is described under individual keywords.

KEYWORD SUMMARY

The keywords accepted by the abs command are listed below in two groups: those pertaining to the absentee facility as a whole, and those pertaining to individual jobs. Detailed descriptions of the keywords and their arguments are given under "Keyword Descriptions" below.

Keywords pertaining to entire absentee facility:

cpu_limit
: do not run jobs with time limits higher than the specified values

maxq
: run only jobs from queues 1 through the specified queue

maxu
: run no more than the specified number of jobs at once

qres
: reserve the specified number of absentee slots for each queue

start
: start absentee facility

stop
: stop absentee facility

abs

Keywords pertaining to individual jobs:

bump
    log out running job(s); leave in queue (if restartable)

cancel
    delete job from queue; log out if running

defer
    dc not log job(s) in until they are released

list
    list specified job(s)

move
    move job(s) to another queue

next
    log job in next

release
    allow deferred job(s) to run

run
    log job in immediately

suspend
    cause running job(s) to stop running until released

terminate
    same as "bump", but job will appear to have taken a fatal process error

JOB SELECTION

Some of the keywords described below specify operations that can be performed on a single job only (indicated as JOB); others specify operations that can be performed on one or more jobs (indicated as JOBS). You can specify the job(s) by using the job selection arguments. Each argument can appear only once in the command. If the request operates on a single job, give enough job selection arguments so that only one job is found. If the request operates on multiple jobs, all jobs that match the specified job selection arguments are acted upon. To be considered a match, a job must fit all the job selection arguments specified. That is, the selection arguments make selection more specific. For example, "-queue 1" selects all jobs in absentee queue 1; "-queue 1 -user Smith" selects all Smith's jobs in absentee queue 1, but does not select any other user's jobs, nor does it select any of Smith's jobs that are in other queues. Job selection arguments are not treated as "match at least one" but rather "match all."

You can choose the job selection arguments from the following list. Specify each selection only once.

absN
where N is a decimal number or "*". It selects the running job in absentee slot N, or all running jobs if you give "*". Slot numbers are printed by the who initializer command or the as_who Multics command.

-entry STR, -et STR
where STR is the entryname of the absin segment. It can be a starname.

-foreground, -fg
is equivalent to "-queue fg."

-id ID, ID
where ID can be a long job ID or a substring of a job ID long enough to uniquely identify one job. A long job ID is a 19-digit number of the form

```
yymmddHHMMSS.UUUUUU
```

which is the time (GMT) at which the job was entered. You can give any substring of the ID that contains either the decimal point or the digit to its left. If you give no decimal point, it is assumed to be to the right of the last digit given. Any other job selection arguments you give must identify the same job, no matter what keyword you give it with. If the other job selection arguments do not match the uniquely specified job, the command is rejected. You can give both -user NAME and -id ID without their identifying control arguments if either one, or both, is the first arguments after the keyword.

path
is the absolute pathname of the absentee input (absin) segment. The pathname can end in a starname.

-queue STR, -q STR
where STR is the absentee queue to be searched for the job, and can be 1 through 4 or "foreground" (fg). If you omit -q, all queues are searched.

-sender STR
where STR is the name of the RJE station from which the job was entered. It can be a starname.

-user NAME
where NAME can be specified in one of the following forms:

```
Person
Person.Project
.Project
```

The star convention is allowed except when a single job is selected. Both -user NAME and -id ID may be given without their identifying control arguments if either one, or both, are the first arguments after the keyword.

KEYWORD DESCRIPTIONS

bump JOBS
bumps the specified running job or jobs. If they are restartable jobs, they remain in the queue and are restarted at some later time.

cancel JOB
deletes a single job from the absentee queues, and bumps it if it is running. Note that some form of the user name must always be given with the cancel keyword as protection against accidental cancellation of the wrong user's job.

cpu_limit, cpu_limit auto, cpu_limit s1,s2,s3,s4
either prints the CPU time limits, restores them to their automatic values (which are per-shift values set by the system administrator), or sets them to the specified values. Values, in seconds, must be given for queues 1 through 4, separated by commas. Omitted values (indicated by adjacent commas) leave the limits for the respective shifts unchanged. (Trailing commas are optional.) The limits remain at the specified values until the next Multics bootload, unless reset to their automatic values by specifying cpu_limit auto. Jobs with estimated times greater than the automatic or user-specified CPU time limits are not run until the limits are raised (either by the operator or by the arrival of a shift having higher limits). If no argument is given, the current CPU time limits are printed.

defer JOBS
places the specified jobs in the deferred state. Jobs in this state are deferred indefinitely, even across Multics bootloads -- that is, they are not run until released (by the abs command with the release keyword). Users can place their own jobs in the deferred state. This is useful if a job is not to be run until the occurrence of some event, such as the delivery of a reel of tape to the computer room. The list keyword (with suitable arguments) can be used to list all jobs that are deferred indefinitely.

list JOBS {-control_args}
lists the specified jobs. All of the control arguments accepted by the list_abs_requests command (described in the *Multics Commands and Active Functions* manual, Order No AG92) are accepted with the list keyword. These control arguments are summarized below:

-absolute_pathname, -absp
prints absolute pathname of selected jobs. (The default is that entry name is printed.)

-all, -a
prints totals for all nonempty queues. (The default is that nothing is printed for queues from which no jobs are selected.)

abs

-deferred_indefinitely, -dfi
selects only jobs that are deferred indefinitely.

-immediate, -im
selects only jobs that are neither deferred indefinitely nor deferred until a specified time.

-long, -lg
prints all information for each selected job. This argument produces many output lines for each job, and should not be used when a large number of jobs are being listed.

-long_id, -lgid
prints all 19 digits of IDs of selected jobs.

-position, -psn
prints the position in the queue of each selected job.

-resource {STR}, -rsc {STR}
selects only jobs with resource requirements, where STR is the name of a resource. If STR is given, selects only jobs requiring the specified resource (e.g., -rsc tape_drive).

-total, -tt
prints only the total number of jobs selected from each queue. (The default is that one line is printed for each selected job).

maxq, maxq auto, maxq N
either prints the highest numbered (lowest priority) queue being searched, restores it to its automatic value (which is a per-shift value set by the system administrator), or sets it to the specified value N. The maximum queue remains at the specified value until the next Multics bootload, unless reset to its automatic value by specifying maxq auto. Jobs from higher numbered queues are not logged in until the maximum queue is raised (either by the operator or by the arrival of a shift with a higher automatic maxq value). If no argument is given, the current maxq figure is printed.

maxu, maxu auto, maxu N
either prints the number of absentee slots (i.e., the maximum number of simultaneously running background absentee jobs) restores it to its automatic value (a per-shift value set by the system administrator that can optionally adjust itself automatically depending on the interactive load), or sets it to the specified value N. The number of slots remains at the specified value until the next Multics bootload, unless reset to its automatic value by specifying maxu auto. If no argument is given, the current abs maxu value is printed.

move JOBS -to_queue N, move JOBS -to_q N
: moves the specified job or jobs to the end of the specified queue (N). Any jobs already running are not moved. The abs move command operates the same as the move_abs_request command, described in the *Multics Commands and Active Functions* manual, Order No. AG92, and accepts the same arguments. See that manual for more information.

next JOB
: places the specified job ahead of all other queued jobs, so that it is the next job to log in. The job is actually moved to queue zero, which has higher priority than queue 1. The job is charged queue 1 rates when it runs. A series of abs next commands cause the specified jobs to accumulate in queue zero; they are run in the order in which they are specified.

qres, qres auto, qres R1 {R2 {R3 {R4}}}
: either prints the per-queue reserved slot values, restores them to their automatic per-shift values (specified by the system administrator), or changes them to the specified values. Values for up to 4 queues may be given; zero is the default for any values not given. To set all 4 values to zero, type "abs qres 0". If no arguments are given, the current values are printed. Values set by this command remain in effect until the next Multics bootload, unless restored to their automatic values by specifying qres auto.

  The slot reservation feature prevents jobs from lower priority queues from logging in if they would occupy slots reserved for higher priority queues. However, jobs from higher priority queues are permitted to log in and occupy slots reserved for lower priority queues.

release JOBS
: releases the specified job(s) from the deferred or suspended states, allowing them to log in, or to resume running, as the case may be.

run JOB
: forces the specified job to be logged in immediately, bypassing all absentee load control processing. A new temporary absentee slot is created, if necessary. Under certain circumstances, the user is warned that logging in the job might create problems, and is asked if the job should be logged in despite the potential problems. Note that some form of the user name must be given with the run keyword.

abs

start {M {Q}}, start queue Q
: the first form is used to start the absentee facility. (It is normally started at answering service startup time, making use of this command unnecessary.) The optional arguments are the maximum numbers of users (M) and queues (Q). (These are the same parameters that can be set by the maxu and maxq keywords.) If they are not specified, the automatic values set by the system administrator are used.

  The second form restarts a queue that was dropped due to errors, and it also wakes up the absentee facility, causing it to check whether any jobs waiting to log in could now do so due to changed conditions. If Q is omitted in the second form, the current error status of each queue is printed. If Q is "all" then all queues are restarted.

stop {now}, stop queue Q
: the first form is used to stop the absentee facility 30 minutes after the command line is issued. If "now" is specified, the absentee facility is immediately stopped without the 30 minute grace. (See "Notes" below.)

  The abs stop command can also be given before the answering service is started to prevent the absentee facility from being started automatically.

  The second form causes the specified queue to be dropped, as if it had gotten too many errors. The absentee facility continues to run, taking requests from the other queues. If Q is "all" then all queues are dropped. The queue(s) can be restarted using the abs start queue command.

suspend JOBS
: causes the specified running job(s) to enter the suspended state, in which they remain logged in but use no cpu or memory resources. The jobs remain in this state until released. A suspended job places no load on the system (although it does occupy an absentee slot and uses up some units from its load control group, possibly preventing other interactive or absentee users from logging in). Jobs can be suspended to reduce the load on the system, allowing the timely completion of some critical work in another process. Suspended jobs should be released as soon as the critical work is completed, since they are NOT preserved across shutdowns and crashes.

terminate JOBS
: causes the specified running job(s) to be bumped in a way that makes them appear to have failed because of a fatal process error. Certain debugging facilities treat such failures differently from normal bumps. This keyword should only be used at the request of the user.

NOTES

Some of the abs command keywords place the absentee facility, or one or more jobs, in an unusual state, where they will remain indefinitely. You must always remember to restore the absentee facility or the affected jobs to normal operation after the unusual circumstances have passed.

abs

The "abs stop" command line should be issued about 20 minutes before a stop command is invoked. If a down command has been issued, "abs stop" is automatically issued 20 minutes before the scheduled shutdown time.

After an "abs stop" command line is issued, the absentee facility does not let any more absentee users log in and waits for all current ones to log out. If all absentee users have not logged out in 30 minutes, the absentee facility automatically bumps the remaining absentee users.

The difference between the 30-minute time limit on absentee jobs and the 20 minutes between a scheduled shutdown and the automatic "abs stop" is intended to give absentee processes an additional 10 minutes to finish after all interactive users are bumped.

If all absentee users log out before 30 minutes, the message:

```
admin:  All absentee processes have run to completion.
```

is printed.

If not, then the message:

```
admin:  bumping all remaining absentee processes .
```

is printed.

If it is necessary to stop the absentee facility immediately with an "abs stop now" command line, all absentee users are bumped as if the 30-minute time limit had expired. Since this command line forcibly terminates user programs that may have difficulty restarting, the now argument should be used only when specifically authorized.

If a down command has scheduled an automatic shutdown, issuing an "abs start" automatically schedules an "abs stop" 20 minutes before the scheduled shutdown time.

The "abs bump" command line causes an absentee user to be bumped. If the absentee job has been declared restartable, the job is left in the queue and reexecuted later. This command can be used when shutting down the absentee facility quickly because of some system malfunction, usually at the request of a system programmer. If a user calls and asks to have his absentee job bumped, you should make certain that the "abs cancel" command is not intended instead.

The "abs cancel" command line can also cause an absentee user to be bumped. It differs from "abs bump" in that restartable absentee requests are removed from the queue, and not restarted, and jobs not yet started can be removed from the queue. This command line is used to cancel an absentee job that is in trouble (for instance, one that seems to be reissuing itself in a loop, or one that calls for a tape that does not exist) or that a user has requested be cancelled.

Name: accept

SYNTAX AS A COMMAND

```
accept channel_id {restrict} {target} {bclist}
```

FUNCTION

accepts a terminal device channel and connects it to the message coordinator's device complement. This command cannot be used on multiplexer channels. This command can be used only in ring 4.

ARGUMENTS

channel_id
: is the name of a communications channel that must be listed with "service: mc;" in the CMF, or be dialed up by a user who has issued the dial system or slave commands. (See the *Multics System Maintenance Procedures* manual, Order No. AM81, for information on channel names.)

restrict
: may be any of the following:

  full
  : the device is able to issue all initializer commands. This is the default.

  none
  : no commands allowed.

  reply
  : only reply is allowed.

  query
  : only who and hmu are allowed.

  daemon
  : only reply, intercom, and exec allowed.

target
: if specified, is a source name that is the only source name permitted for reply commands issued from channel_id. This parameter is used for terminals dedicated to the control of a single I/O daemon. The default name is *.

bclist
if you specify it, it is a "broadcast list." This list gives the channels that will have copies of input from channel_id, in the form

```
(input on channel_id) r cdl hello
```

bclist can be:

```
none                for no broadcasting
all                 to broadcast to all others
a.h111,a.h102       to broadcast to the given list
```

Output is never broadcast back to the inputting channel. If you specify no bclist, any broadcast list previously specified remains unchanged.

NOTES

Response: channel_id attached by system control.

Name: add_lv, alv

SYNTAX AS A COMMAND

```
alv {lv_name} {-control_arg}
```

FUNCTION

accepts a logical volume for paging. You can use it in ring 1 or 4.

ARGUMENTS

lv_name
is the name of a registered logical volume.

CONTROL ARGUMENTS

-all
mounts all incomplete logical volumes.

NOTES

If all physical volumes for the logical volume are already accepted, known, or assumed, alv ensures that all labels have been read and checked, and then accepts the logical volume for paging.

If one or more volumes are missing, alv assigns drives for the missing volumes and prints mount messages for each missing volume in the form

```
mount pv k202 on dska_07
```

and sets the assigned drives into the assumed state. You can mount the physical volumes where requested or use your own judgment. As each physical volume is mounted, issue an add_vol command to the system; the last add_vol command causes the logical volume to be accepted with a message of the form

```
lv cp22 mounted
private lv cp22 mounted
```

When you issue alv, all assumed drives should have the correct packs mounted; otherwise an error message is typed.

The list_disks command lists all outstanding alv commands and the del_lv command cancels an outstanding alv.

User processes can initiate alv commands for private logical volumes through the attach_lv Multics command. The system proceeds as though an alv command had been issued by the operator.

Unless the nodt and/or nolv keyword is included with the BCE boot command, the system attempts to mount during initialization all logical volumes that were mounted at the last shutdown. For each physical volume required, the system attempts to use the drive on which the volume was mounted at the last shutdown.

Name: add_pdir_volume

SYNTAX AS A COMMAND

```
add_pdir_volume lv_name
```

FUNCTION

adds a logical volume to the set of volumes available for process directory segments. You can use it only in ring 4.

ARGUMENTS

lv_name
is the name of the mounted public logical volume.

NOTES

Process directory segments are temporary segments associated with each interactive, absentee, and daemon process. For system efficiency, make available as many volumes as possible for process directory segments. You can't dismount a volume, however, while it is a pdir_volume, or while process directory segments reside on it.

See the del_pdir_volume, set_pdir_volumes, and vacate_pdir_volume commands.

Name: adddev, addd

SYNTAX AS A COMMAND

```
addd DEVNAME
```

FUNCTION

adds a tape or disk drive to the Multics configuration, and tells the system that it can use the specified drive. You can use this command only in ring 1.

ARGUMENTS

DEVNAME
is the device name of the drive to be added.

EXAMPLES

```
! addd tape_06

! addd dska_15
```

Name: add_vol, av

SYNTAX AS A COMMAND

```
av pv_name drive_name
```

FUNCTION

tells the system that a physical volume is on a disk drive. The system then reads and checks the volume label. You can use this command in ring 1 or 4.

ARGUMENTS

pv_name
is the name of a physical volume.

drive_name
has the form <subsys>_<number (nn)>{subvol (sv)}, e.g., dskb_00c for devices that are divided into subvolumes and dska_02 for devices that are not divided into subvolumes. It can be -all to cause the system to read and check the labels of all assumed physical volumes.

NOTES

The registration file for the volume name is consulted to find out the logical volume name and to validate the label. The volume label is read and checked. If the label is correct the drive is left in the known state.

If you have previously issued av for the logical volume containing this physical volume and if this invocation completes the logical volume, the logical volume is accepted for paging and a message is printed.

Name: admin

SYNTAX AS A COMMAND

```
admin
```

FUNCTION

causes the system to enter admin mode, allowing you to use the bootload console or an initializer terminal to execute normal Multics commands. You can't be in admin mode on more than one terminal at a time. Because the initializer has exceptional power and special limitations, only a system programmer, a system administrator, or an authorized operator should use admin mode. To enforce this, a password is required to enter admin mode. You can use this command only in ring 4.

EXAMPLES

When you type

```
admin
```

the system enters admin mode and responds with

```
Password:
```

and turns off the printer. Type the password (you need no password if the system administrator has set the admin mode password to "*"). The system responds with the normal ready message. Execute any desired commands, and then type the command

```
admin_mode_exit
```

or its short name

```
ame
```

to return to initializer command level.

Name: attach

SYNTAX AS A COMMAND

```
attach channel_id1 {channel_id2...channel_idN}
```

FUNCTION

controls communications channels for login, slave, autocall, or ftp service. Specifically, it directs the answering service to use, or make available, specified channels that have previously been marked inactive. A channel can be set inactive by the remove command, COLTS, or errors. You can't use attach on multiplexer channels. You can use it only in ring 4.

ARGUMENTS

channel_ids
: are channel names (see the *Multics System Maintenance Procedures Manual*, Order No. AM81). This argument can be either terminal channel names or network channel names.

Name: bce

SYNTAX AS A COMMAND

```
bce
```

FUNCTION

enters the bootload command environment (BCE). All Multics operation is suspended. When the system is in trouble, it is sometimes necessary to enter BCE to use the dump or probe commands. You can use it in ring 1 or 4.

NOTES

Use bce only with the system programming staff's permission, as it causes disruption of service to users. Under some circumstances, it can cause interactive users to be disconnected.

EXAMPLES

When you type

```
bce
```

the system enters BCE.

To restart Multics, type:

```
go
```

on the bootload console.

Name: bump

SYNTAX AS A COMMAND

```
bump target {mm} {message}
```

FUNCTION

logs out a user at a specified time by causing the initializer to signal itself to bump the user. If the initializer is operating improperly and cannot handle logins, bump probably will not operate properly either. You can use it only in ring 4.

ARGUMENTS

target
must be one of the following:

Person_id.Project_id
bumps the user with the specified name and project. You can use the star convention for either or both names. Users with the nobump privilege are sent a message informing them that an attempt to bump them has been made.

channel_id
bumps the user on the channel whose name is channel_id. These names can be either terminal or network channel names. This argument overrides the nobump attribute (see the *Multics System Maintenance Procedures Manual*, Order No. AM81).

mpx NAME
bumps all users whose terminals are connected through the multiplexer specified by NAME. NAME must be the name of a multiplexer or an FNP.

mm
is the number of minutes to elapse before the user is bumped. If omitted, the user is bumped immediately.

message
is a string that is typed on the user's terminal. If omitted, the user is not told why he is being bumped.

NOTES

Sometimes, the bump command appears to work--the message "Person_id.Project_id bumped" is printed--but the user is not logged out. If this happens, try the remove command on the user's terminal channel.

See the abs command for instructions on bumping absentee users. See the logout command for instructions on logging out daemon processes.

EXAMPLES

```
bump * * 3 Problems with reconfiguration.  Up in 10 minutes.

bump Marcos.Multics

bump a.h013
 bump mpx a
```

Name: cripple

SYNTAX AS A COMMAND

```
cripple
```

FUNCTION

prepares for a hierarchy reload operation that reloads the system library directories or the initializer's directory >system_control_1.

NOTES

To do such a hierarchy reload, bring up a special session and log in a daemon process to be used for reloading (e.g., Dumper.SysDaemon). The message coordinator should not be used during such a reload, since its programs and databases are likely to be replaced by the reload, too. (See the no_start command.) You can use cripple only in ring 4.

After the reloader is logged in, execute cripple to shut off the answering service so that no unexpected faults occur if one of the answering service databases or programs is deleted by the reloader.

When the reload is done, do not attempt to shut down, as this is likely to cause a fault. Instead, reenter BCE with the bce command or from the processor panel. (A reset command resets the effect of cripple, but is not recommended.)

Name: debug

SYNTAX AS A COMMAND

```
debug
```

FUNCTION

preserves the prior stack history, should a fault occur, for later use for debugging. It is the opposite of the release command. You can use it in ring 1 only.

Name: define

SYNTAX AS A COMMAND

```
define vcons type dest
```

FUNCTION

defines a virtual console or adds a destination for it to the destination list of a previously defined virtual console. You can use it in ring 4 only.

ARGUMENTS

vcons
is the name of the virtual console to which output is to be routed.

type
is the type of destination. It has the following effects on the dest argument:

If type is tty, dest must be a channel_id that has been accepted previously.

If type is log, dest is the name of the log file in >sc1 to which messages are added as they are sent to the virtual console.

If type is sink, dest can be any name; output sent to a sink vanishes.

dest
is the destination for the virtual console. A virtual console can have up to eight destinations.

Name: del_lv, dlv

SYNTAX AS A COMMAND

```
dlv lv_name
```

FUNCTION

forces the demounting of a logical volume. You can use it in ring 1 or 4.

ARGUMENTS

lv_name
specifies the name of a logical volume.

NOTES

If the logical volume is being mounted, dlv cancels the request.

If the logical volume is already mounted, dlv makes the segments on the volume unavailable to all users, and shuts down and unloads all physical volumes in the logical volume with a message of the form

```
demounted dska_02
```

for each physical volume.

Name: del_pdir_volume

SYNTAX AS A COMMAND

```
del_pdir_volume lv_name
```

FUNCTION

removes a logical volume from the set of logical volumes available for process directory segments. You can use it only in ring 4.

ARGUMENTS

lv_name
is the name of a logical volume currently available for process directory segments.

NOTES

To demount a process directory volume, you must delete it from the set of process directory volumes and wait for existing process directories whose segmemts are on it to be deleted when the processes are destroyed (by bump, logout, or new_proc).

See the add_pdir_volume, set_pdir_volumes, and vacate_pdir_volume commands.

Name: del_vol, dv

SYNTAX AS A COMMAND

```
dv drive_name
```

FUNCTION

tells the system that a physical volume is no longer on a drive. You can use it in ring 1 or 4.

ARGUMENTS

drive_name
has the form <subsys>_<number (nn)>{subvol (sv)}, e.g., dskb_00c for devices that are divided into subvolumes and dska_02 for devices that are not divided into subvolumes.

NOTES

The drive specified cannot contain part of an accepted logical volume. (See the del_lv command for demounting logical volumes that are in use.)

You can use dv to cancel the assumed or known states of any drives.

Name: deldev, deld

SYNTAX AS A COMMAND

```
deld DEVNAME
```

FUNCTION

removes a tape or disk drive from the Multics configuration, and tells the system that it can't use the specified drive. You can use this command only in ring 1.

ARGUMENTS

DEVNAME
is the device name of the drive to be removed.

Name: deroute

SYNTAX AS A COMMAND

```
deroute source stream old_vcons
```

FUNCTION

removes a virtual console from a specified source. You can use it only in ring 4.

ARGUMENTS

source
is the name of the source of output.

stream
is the name of the stream on which output is being carried.

old_vcons
is the name of the virtual console to be removed from the output list.

NOTES

If the stream is left with no virtual consoles, output is sent to the default virtual console, which is usually defined to be the bootload console.

Name: detach

SYNTAX AS A COMMAND

```
detach target {message}
```

FUNCTION

prevents the answering service from listening for calls on a channel. Any user logged in on the channel is bumped. This is done by signaling, as with the bump command. You can use detach only in ring 4.

ARGUMENTS

target
must be one of the following:

Person_id.Project_id
detaches the user with the specified name and project. You can use the star convention for either or both names.

channel_id
detaches the user on the channel whose name is channel_id. These names can be either terminal or network channel names. See the *Multics System Maintenance Procedures* manual, Order No. AM81.

mpx name
detaches all users whose terminals are connected through the multiplexer specified by name. The name argument must be the name of a multiplexer or an FNP.

message
is the string that is typed on the user's terminal. If you omit message, the user is not told what is happening.

NOTES

If the message "channel_id detached" is printed, then the detach has been signaled. If the user on the channel does not log out shortly, try the remove command.

Name: disconnect

SYNTAX AS A COMMAND

```
disconnect target
```

FUNCTION

hangs up a login-service communications channel.

ARGUMENTS

target
must be one of the following:

Person_id.Project_id
disconnects the login channel for the user with the specified name and project. You can use the star convention for either or both names.

channel_id
disconnects the user on the channel whose name is channel_id. These names can be either terminal or network channel names. See the *Multics System Maintenance Procedures* manual, Order No. AM81.

mpx name
disconnects all users whose terminals are connected through the multiplexer specified by name. The name argument must be the name of a multiplexer or an FNP.

NOTES

The user process gets disconnected, which allows the user to log in again and reconnect to the process. If the user is not allowed to have disconnected processes, or has disabled disconnection service for the process, the process is logged out. You can use disconnect when part of the communications network has broken down without informing Multics; for example, if a foreign terminal concentrator crashes, it is possible for Multics to still treat the communication lines as dialed-up.

This command causes the initializer to send itself a signal to hang up the specified channel. If the initializer or Multics communications software is malfunctioning, disconnect may operate improperly. You can use this command only in ring 4.

EXAMPLES

```
disconnect Marx.Multics

disconnect a.h013

disconnect mpx a
```

Name: display_volume_log

SYNTAX AS A COMMAND

```
display_volume_log vlname {-control_args}
```

FUNCTION

displays a summary of the dump volumes that contain information dumped from a specified physical volume. You can use it in ring 1 only.

ARGUMENTS

vlname
is the name of the volume log. If you specify no volog suffix, it is assumed.

CONTROL ARGUMENTS

-complete, -comp
displayes information only about complete mode dump volumes.

-consolidated, -cons
displayes information only about consolidated mode dump volumes.

-incremental, -incr
displayes information only about incremental mode dump volumes.

-no_header, -nhe
does not print the output header. (Default: to print the output header)

-volname STR
prints only the information pertinent to the dump volume named STR.

NOTES

If you give no -comp, -cons, or -incr, information about all modes is displayed.

Name: down

SYNTAX AS A COMMAND

```
down {time} {back} {reason}
```

FUNCTION

schedules an automatic shutdown for a specified time and an automatic bump of users for N minutes before the time of the shutdown. At the time of the bump, all interactive users are warned that they will be logged out in N minutes. You can use it only in ring 4.

ARGUMENTS

time
is the time to schedule an automatic shutdown. It is a clock reading acceptable to convert_date_to_binary_. Enclose it in quotes if it contains spaces. If you omit it, the command displays the current down settings.

back
is the time announced to users when the system will come back up. Give it in the same form as the time argument.

reason
is a message that tells users the reason for the shutdown.

NOTES

N is figured using the value of warning_time, a parameter in the installation_parms segment. The warning_time parameter is the number of real-time seconds between warning of an automatic logout and the actual logout see the *Multics System Administration Procedures* manual, Order, No. AK50.) N equals warning_time divided by 60, rounded up to the next whole minute; for example, if the installation_parms segment shows a warning_time value of 6000 seconds, N will be 100 minutes.

This page intentionally left blank.

When the system is started up, a check is made to see if a down command has been issued that has not yet taken effect. If one is found and more than 30 minutes remain before it takes effect, the down command is reissued automatically.

If absentee is up when the down command is issued an automatic abs stop is set up for 20 minutes before the time specified by the time argument. If absentee is not up when the down command is issued, but it is brought up later, the automatic abs stop is still set up for 20 minutes before shutdown. If absentee is already being shut down when the down command is issued, the shutdown of absentee continues.

EXAMPLES

To schedule a shutdown at 5:45 a.m., and to tell the users that the system is coming back at 9:00 a.m., type:

```
! down 05:45 09:00 Regularly scheduled shutdown.
```

To cancel a scheduled shutdown, type:

```
! down 0
```

To display the time of the next shutdown, type the down command with no arguments.

Name: drop

SYNTAX AS A COMMAND

```
drop channel_id
```

FUNCTION

causes a communications channel to be dropped from message coordinator use. Any pending output for the channel is lost. If the channel was dialed to the initializer, it is disconnected. This command can be used in ring 4 only.

ARGUMENTS

channel_id
: is the name of the communications channel to be dropped (see the *Multics System Maintenance Procedures* manual, Order No. AM81, for information on channel names.) This command cannot be used on multiplexer channels.

EXAMPLES

When you issue the drop command:

```
drop
```

The system responds with:

```
please reissue dial command (only if channel was dialed)
```

Name: dump_mpx

SYNTAX AS A COMMAND

```
dump_mpx mpx_name
```

FUNCTION

forces an immediate dump of the specified multiplexer. This causes the multiplexer to crash, hanging up any users logged in over its channels. Therefore, if there are any such users, the number of users is printed, and you are asked if the dump should still be taken. Any reply other than "yes" causes the dump not to be taken, and the multiplexer to continue running. All the channels in the CDT that match the starname a.* are hung up by this command. This command can be used in ring 4 only.

ARGUMENTS

mpx_name
specifies the name of the multiplexer to be dumped.

NOTES

Individual multiplexers respond to this command in whatever way is appropriate to their implementation. For FNPs a message is printed, giving the pathname of the segment containing the dump of the FNP. This is not a printable segment; its contents can be examined and interpreted by a system programmer, using the online_dump_fnp (od_fnp) or debug_fnp command.

This command should only be used at the direction of the system programming staff. The answering service automatically reloads a multiplexer if it crashes.

Name: exec, x

SYNTAX AS A COMMAND

```
x exec_command {arguments}
```

FUNCTION

executes operator commands implemented in the segment >sc1>admin.ec. You can use it in ring 4 only.

ARGUMENTS

exec_command
is one of the commands listed below:

```
attended, attend
auth
auto
cat
complete_volume, vcomp
consolidated_volume, vcons
copy_dump
delete_dump, dd
deny
echoplex
end_dump
inc
incremental_volume, vinc
io
iol
meter
print_queues, pq
punch, puna
punch_end, end_punch
punch_restart
read_cards, cards, rc
repair
reprint, rep
reset_phcs_access, rpa
reset_tabs, tabs, rt
scav
set_fdump_number, set_fdump, sfdn
set_phcs_access, spa
unattended, unattend
wakeup_dump
```

arguments
are any arguments for the chosen exec_command.

NOTES

The version of admin.ec distributed with the system implements the above exec commands, which are generally useful. They are described in Section 5.

You site may wish to modify these commands, or to implement new operator commands, to meet its needs. You can do this by making changes or additions to >sc1>admin.ec. Document these changes and additions, however, by adding or replacing pages in Section 5 in the copies of this manual used by operations personnel.

The effect of exec is as if you had typed the command

```
exec_com admin exec_command arguments
```

in admin mode.

Name: force_reset

SYNTAX AS A COMMAND

```
force_reset
```

FUNCTION

forces the answering service to reset itself. It is similar to the reset command but does additional resetting. If administrators are unable to install system tables, this function can sometimes clear the jam. This command can be used in ring 4 only.

Name: go

SYNTAX AS A COMMAND

```
go
```

FUNCTION

initializes all channels attached to the answering service so that they answer the phone. It is the second half of startup. Typing multics and then go is similar to startup except that the login word is set to a random number, so only users who know the special word can log in. The go command initializes all lines in the channel definition table (CDT). Parts two and three of the system_start_up.ec are executed in response to a go command. Part two (the logging in of the daemons) is done before user lines are conditioned for answering; part three (commands executed after the phone lines are conditioned for answering) is done afterward. This command can be used in ring 4 only.

NOTES

A check is made to see if a down command has previously been issued. If one is found, and more than 30 minutes remain before it takes effect, the down command is reissued automatically.

Name: help

SYNTAX AS A COMMAND

```
help {command_name}
```

FUNCTION

in ring 1, prints a list on the bootload console of all ring 1 initializer commands. The argument command name cannot be used in ring 1.

In ring 4, prints a description of the specified command.

ARGUMENTS

command_name
identifies the command for which information is desired. The suffix "info" is assumed.

Name: hmu

SYNTAX AS A COMMAND

```
hmu
```

FUNCTION

prints how many users are logged in. Since it also prints the system ID, it can be used to check the success of a sysid command. This command can be used in ring 4 only.

EXAMPLES

Type hmu to get a message in the following form:

```
Multics SYSID; INSTALLATION
Load = XX.X out of YY.Y units:  users = ZZ
Absentee users = W; Maximum absentee users = K
```

Name: init_vol

SYNTAX AS A COMMAND

```
init_vol volume_name drive_name {-control_arg}
```

FUNCTION

writes the label of a new physical volume and sets up its VTOC and volume map. This operation destroys any previous contents of the physical volume. This command can be used in ring 1 or ring 4. It is also automatically entered when a cold boot is done to initialize the RPV.

ARGUMENTS

volume_name
is the name of a physical volume.

drive_name
has the form <subsys>_<nn>, e.g., dska_02.

CONTROL ARGUMENTS

-special
enters dialogue with the user. The -special control argument is recommended for all RLV volumes.

-copy
initializes a copy of an already accepted physical volume (-copy also enters dialogue with the user).

-rlv
specifies that the physical volume is to be registered as part of the root logical volume. this control argument can be used only during a cold boot of the root physical volume at ring-1 initializer command level.

SPECIAL MODE

Special mode is entered when -special is given on an init_vol command line, when the rebuild_disk command is used, or when a cold boot of the RPV is done. In special mode, the user may specify various parameters of the volume being initialized as other than their default values. Special mode uses a request loop subsystem to allow entry of these parameters. This includes the ability to specify the locations and extents of partitions.

Defaults for volume initialized by init_vol or rebuild_disk command:

- no partitions
- VTOC size constrained by average segment length of 5.0

Defaults for RPV initialized by cold boot sequence:

- partitions laid out on pack as follows:

```
partition          partition          low/high
  name        size (in records)      end of pack

HC                    2500            low
CONF                     4            low
<paging region>     <all space not used by partitions>
FILE                   255            high
BCE                   2200            high
LOG                    256            high
DUMP                  2000            high
BOS                    270            high
ALT                as needed          high
```

- VTOC size constrained by average segment length of 2.0

For non-RPV packs belonging to the RLV, it is recommended that special mode be used to set an average segment length of 4.0 since directories, which reside on the RLV, are usually smaller than other segments. A smaller average segment length increases the number of VTOC entries (VTOCES) on the pack. Since it is easier to make the VTOC bigger than to make it smaller, the recommended procedure is to start with 4.0 as an average segment length. If more VTOCES are needed later, the rebuild_disk command can be used to define them.

When using volume backup to recover the contents of a disk pack, make sure there are at least as many available records and total VTOC entries as there were on the physical volume before it was damaged.

REQUEST LOOP OF INIT_VOL

The following request lines may be typed when init_vol is invoked with the -special control argument, when the rebuild_disk command is being used, or within the cold boot sequence:

asl FFF.FF, avg FFF.FF
: specifies the average segment length, which is used to determine the VTOC size. As partitions are defined, the VTOC size is adjusted to maintain a VTOC entry to free page ratio producing this average segment length. When operating in this mode (the default, with average segment length = 5.0), the parameters are said to be constrained by average segment length.

default
: causes all parameters to be reset to their cold boot defaults, including the list of partitions. This request is only valid within the cold boot sequence.

end
causes the command init_vol or rebuild_disk to proceed using the parameters as they stand at this point.

help
lists all requests available within the init_vol request loop.

lace N
specifies physical address assignment interlace; i.e., specifies that the system should try to place the pages of a segment N disk records apart on the target disk. The default value for N is 2. This request is only valid in the rebuild_disk command.

list
causes the current parameters to be listed, including partition data, VTOC size, number of VTOC entries and effective average segment length. Issuing other requests causes these parameters to change. When the request loop is entered, a list request is performed automatically.

nvtoce N
specifies the VTOC size by the number of VTOC entries to be created. As five VTOC entries occupy each page, this number is rounded up to the next five before use.

This constrains parameters by VTOC size.

part name hilow size
where name is a 4-character or fewer partition name, hilow is either "high" or "low", and size is the partition size in records, defines a partition to be allocated on the pack. The hilow argument specifies which end of the pack, with respect to device address, the partition shall be allocated. Successive requests for "high" partitions receive successively lower addresses, and successive requests for "low" partitions receive successively higher addresses. Partitions may not be redefined without issuing the "startover" request.

quit
causes the command init_vol or rebuild_disk to be aborted, without initializing or rebuilding any disks. This request is not valid within the cold boot sequence.

startover
causes all parameters to be reset to their defaults (init_vol command defaults, not cold boot defaults). In particular, no partitions are defined, even in the cold boot environment.

vtoc N
specifies the total number of pages to be allocated for the volume header and volume table of contents (VTOC). All other space on the pack is available for paging and partitions. When the "vtoc" or "nvtoce" request is given, the parameters are said to be constrained by VTOC size, and average segment length varies as partitions are defined.

NOTES

The init_vol command queries the user before destroying the label of any pack that appears to be a validly labeled pack. A message giving the pack's physical volume name and time of last use is displayed.

The init_vol command takes about 90 seconds for a model M400 disk.

Name: intercom

SYNTAX AS A COMMAND

```
intercom channel_id message
```

FUNCTION

sends a message string from one terminal controlled by the message coordinator to another. This command can be used in ring 4 only.

ARGUMENTS

channel_id
: is the name of the communications channel to which the message string is sent. (See the *Multics System Maintenance Procedures* manual, Order No. AM81, for information on channel names.)

message
: is the message string.

Name: list_disks, ld

SYNTAX AS A COMMAND

```
ld {drive_name} {-control_args}
```

FUNCTION

lists the usage of storage system disk drives and all pending add_lv commands. This command can be used in ring 1 or ring 4.

ARGUMENTS

drive_name
is the name of a drive to be listed. If you give no drive_name, all drives configured on the system are listed.

CONTROL ARGUMENTS

-lv lvname
list only drives pertaining to lvname.

-pv pvname
list only drives pertaining to pvname.

-brief, -bf
list only nonempty drives.

-mounts, -mt
list only pending ad_lv commands.

NOTES

If a physical volume is listed with the logical volume name in parentheses and one star at the end of the line, that volume is assumed to be present, as a result of a previous bootload, demount, or outstanding add_lv. The system has not read the volume label, but does so in response to an add_vol command.

If a physical volume is listed with the logical volume name in parentheses and three stars at the end of the line, that volume is known; its label has been read by an add_vol command. You must issue an add_lv command to place the volume in use.

EXAMPLES

```
list_disks
 dska_01 root2      root
 dska_02 new        (blue)      *
 dska_03 (io drive)
 dska_04
 dska_05 (deleted)
 dska_06 true       (blue)      ***
 dska_07
 dska_08
 dska_09
 dska_10 (deleted io drive)
 dska_11
 dska_12 rpv        root
 dska_13
```

```
dskb_01
dskb_02
dskb_03
dskb_04
dskb_05
public lv blue mount in progress
```

Name: load_mpx

SYNTAX AS A COMMAND

```
load_mpx name {-control_args}
```

FUNCTION

forces an immediate reload of the specified multiplexer. You can use it in ring 4 only.

ARGUMENTS

name
designates the multiplexer to be loaded.

CONTROL ARGUMENTS

-check, -ck
checks consistency between channels listed in the CDT and channels actually configured on the multiplexer, reporting inconsistencies. Use it after any change in channel configuration. Not all multiplexers may implement it. For FNPs, inconsistencies are reported on the FNP console.

-force
reloads the multiplexer even if some of its channels are in use, bumping users logged in on the channel.

-no_start, -ns
does not listen for calls on the channels of this multiplexer when the load completes (see the start_mpx command). (Default: to listen to the channels)

NOTES

Reloading a running multiplexer hangs up any users logged in over its channels; therefore if you give no -force, the number of any such users is printed and the reload is refused. The multiplexer must be stopped and the users bumped before the reload takes place (see the stop_mpx and bump commands).

Consult the system programming staff before using load_mpx. The answering service loads multiplexers automatically under normal circumstances.

If channels of the multiplexer have been added or deleted since the last time the multiplexer was loaded, these additions and deletions take effect with the next load_mpx command.

Name: lock_mca

SYNTAX AS A COMMAND

```
lock_mca
```

FUNCTION

locks (disables) input to all maintenance channel adapters (MCAs) from the console. You can use it only in ring 4.

Name: log

SYNTAX AS A COMMAND

```
log message
```

FUNCTION

leaves a message for system programmers by entering a line in the system log. You can use it in ring 4 only.

ARGUMENTS

message
    is the message to be entered in the log.

Name: login, logi

SYNTAX AS A COMMAND

```
logi daemon_user_id source {-control_args}
```

FUNCTION

logs in a daemon process at your request. You can use it in ring 4 only.

ARGUMENTS

daemon_user_id
logs in the user whose name is specified by daemon_user_id as a daemon. The daemon user's name takes this form:

```
Person_id.Project_id
```

*

The user whose name is specified by daemon_user_id must have the daemon attribute.

source
is the name of the input source, used in the reply command to direct terminal input to this process.

CONTROL ARGUMENTS

You can choose them from those available with the Multics login access request (see the *Multics Commands and Active Functions* manual, Order No. AG92). Appropriate in this context are the following:

-arguments STR, -ag STR
supplies arguments to the process, where STR can be one or more arguments. All arguments following -ag on the command line are taken as arguments to the process; therefore give -ag last on the login command line. The process can determine the number and value of each argument with the user active function, described in the Commands manual.

-authorization STR, -auth STR
sets the authorization of the process to that specified by STR, where STR is a character string composed of level names and category names for the desired authorization, separated by commas. STR cannot contain any embedded blank or tab characters. (The short names for each level and category are guaranteed not to contain any blanks or tabs, and you can use them whenever the corresponding long names do contain blanks or tabs.) STR must represent an authorization that is less than, or equal to, the maximum authorization of Person_id on the project Project_id. If you omit -auth, the daemon's registered default login authorization is used (see Access Control in the *Multics Programmer's Reference Manual*, Order No. AG91).

-brief, -bf
suppresses messages associated with a successful login. If the standard process overseer is being used, the message of the day is not printed.

-change_default_auth, -cda
changes the daemon's registered default login authorization to the authorization specified by -auth. If the authorization you give is valid, the default authorization is changed for subsequent logins, and the message "default authorization changed" is printed at the terminal. If you give -cda without -auth, an error message is printed.

-home_dir path, -hd path
sets the daemon's home directory to the path specified, if the daemon's project administrator allows this choice.

-no_start_up, -ns
instructs the standard process overseer not to execute the daemon's start_up.ec segment, if one exists, and if the project administrator allows this choice.

-outer_module path, -om path
attaches the daemon's terminal through the outer module named path, rather than through the daemon's registered outer module, if the daemon is allowed this choice.

-process_overseer path, -po path
sets the daemon's process overseer to the procedure given by the path specified, if the daemon's project administrator allows this choice. If path ends in the characters ",direct", the specified procedure is called directly during process initialization, rather than by the standard procedure provided by the system, which means that the program specified by path must perform the tasks that would have been performed by the standard procedure.

-ring N, -rg N
sets the daemon's initial ring to be ring N, if this ring number is greater than, or equal to, the daemon's registered initial ring and less than the daemon's registered maximum ring.

NOTES

Most daemon logins don't need any control arguments; the admin.ec or the system_start_up.ec often take care of them.

This command logs in a daemon process without its own terminal. That process sends all its output through the message routing DIM to the message coordinator for output on one of the message coordinator's output destinations. Use the reply command to do all input to such a process.

Name: logout, logo

SYNTAX AS A COMMAND

```
logo daemon_user_id {source}
```

FUNCTION

logs out a daemon process at your request. You can use it in ring 4 only.

ARGUMENTS

daemon_user_id
: logs out the user whose name is specified by daemon_user_id as a daemon. The daemon user's name takes these forms:

```
Person_id.Project_id
Person_id
```

source
: is the name of the input source.

NOTES

If Person_id, Project_id, or source is "*", all daemons that match are logged out. If you omit source and Project_id, or source, they are assumed to be "*".

EXAMPLES

To log out all daemon processes, type:

```
logout * * *
```

Name: maxunits, maxu

SYNTAX AS A COMMAND

```
maxu {N}
```

FUNCTION

alters the maximum number of load units accommodated by the system. If it is set to below the current number of units, no users are bumped, but only those users with guaranteed login privilege can log in. This command can be used in ring 4 only.

ARGUMENTS

N
is ten times the maximum number of load units (see "Examples" below). N must be greater than zero. If no argument is given, the current value is printed.

NOTES

To cause the system to set the standard value of maxunits from the configuration table set by the system administrator in installation_parms, type:

```
! maxu auto
```

The maxu auto command is normally issued after every reconfiguration. The number of units remains at the specified value until the next Multics bootload, unless this command is issued.

EXAMPLES

To set the maximum load to 41.3 load units, type:

```
! maxu 413
```

Name: mc_list

SYNTAX AS A COMMAND

```
mc_list {args}
```

FUNCTION

lists the message coordinator tables, in the form of a list of define and route commands that can be issued by the user or the system_start_up.ec to create the current message routing tables. This command can be used in ring 4 only.

ARGUMENTS

args
can be the names of up to ten virtual consoles or source_ids.

Name: message, motd

SYNTAX AS A COMMAND

```
motd
```

FUNCTION

invokes the Multics qedx editor to edit the file message_of_the_day, which most (but not all) users print out automatically when they log in. This command can be used in ring 4 only.

NOTES

When you type:

```
! motd
```

the system puts you in qedx edit mode, where you may enter editing requests. Some of the valid requests are:

```
a  to enter input mode
d  to delete a line
s  to change a line
p  to print the current line
\f to exit from input mode and return to edit mode
w  to write the edited message
q  to exit from qedx and return to command level
```

There are many other requests, and options for these requests. See the description of the qedx command in the *Multics Commands and Active Functions* manual, Order No. AG92, for complete information. Note that the e request is not permitted. An error message is typed if the e request is used.

EXAMPLES

To insert a whole new message, the following sequence of commands and edit requests might be used:

```
! message
! 1,$d
! a
! Special shutdown at 04:45 for PM.
! \f
! s/PM/Preventive Maintenance/
! p
  Special shutdown at 04:45 for Preventive Maintenance.
! w
! q
  R
```

It is a good idea to keep the message brief, but not cryptic, and to use lines shorter than 80 characters and with a minimum of tabs and spaces for the convenience of users at terminals that have short carriages.

Name: multics, mult

SYNTAX AS A COMMAND

```
mult
```

FUNCTION

initializes the answering service but does not accept logins. It is the first half of a startup. A random login word is made up, which the users must use to log in on this special session. This command can be used in ring 1 or ring 4.

NOTES

Typing the multics command initializes the answering service and generates a random login word. To cause the lines to answer, type go. To revert to a normal session, use the word command. Part one of the system_start_up.ec is executed before the answering service is initialized.

When this command is issued in ring 1, it causes the entire RLV to be mounted, if not already accepted (when issued in ring 4, the RLV is already accepted).

Name: no_start, ns

SYNTAX AS A COMMAND

```
ns
```

FUNCTION

prevents subsequent execution of the system_start_up.ec segment until the next system bootload. See the *Multics System Maintenance Procedures* manual, Order No. AM81, for more details on the use of this segment. This command can be used in ring 4 only.

NOTES

This command is useful only if executed in ring 4 before a startup, multics, or go command. The no_start command should not be used by operators unless they have been specifically instructed to do so by the system programming staff.

Name: preload

SYNTAX AS A COMMAND

```
preload device_name volume_name -control_arg
```

FUNCTION

informs the system that a tape volume is to be preloaded on a specified device for a certain user. This allows you to load tapes in advance when the needs of a user are known. This command can be used in ring 4 only.

ARGUMENTS

device_name
is the name of the device on which the volume is to be preloaded.

volume_name
is the name of the volume to be preloaded.

CONTROL ARGUMENTS

-user STR
where STR is the name of the user for whom this preload is being performed. This is a string of the form Person_id.Project_id.tag.

NOTES

This command only informs the system of your intention to preload a particular volume on a specified device for a user. It does not check the loading of the proper volume on the device. This label checking is done at attach time.

Use of the preload command does not preserve the preloaded device. The preloaded volume may be automatically unloaded if another user of the system requests a mount of a different volume on the preloaded device.

The user name specified may contain fields that match any string. These are specified by the use of an asterisk (*) in the appropriate field. For example, the user name "*.Proj.*" will match any user on the project Proj. Also, missing trailing fields are assumed to be *. For example, "*.Proj" also matches any user in the Proj project.

Name: quit

SYNTAX AS A COMMAND

```
quit source_id
```

FUNCTION

simulates the action of a BREAK or QUIT key on the specified virtual consoles. This command can be used in ring 4 only.

ARGUMENTS

source_id
is the name of the source process to which a quit signal is to be sent.

Name: rcp

SYNTAX AS A COMMAND

```
rcp keyword {-control_args}
```

FUNCTION

lists resources controlled by the resource control package (RCP), or cancels a reservation. This command can be used in ring 4 only.

rcp

ARGUMENTS

keyword
must be one of the following:

list
lists resources controlled by RCP. This keyword may be used with any control arguments except -id. If used without control arguments, all resources are listed.

cancel
cancels the specified reservation. This keyword must be used with both the -id and the -user control arguments.

CONTROL ARGUMENTS

-device STR, -dv STR
specifies the name of one device to be listed.

-id STR
specifies the ID of the reservation to be cancelled. This control argument must be used in conjunction with the cancel keyword.

-long, -lg
specifies that all of the information known about each device listed is printed. If this argument is not specified, only the state of the device, the time the device was put into that state, the process group ID of the process that has the device assigned, and any volume mounted on the device are printed. If this argument is specified, then all of the characteristics of each device are printed. Also, for each device type listed all of the per device type information is printed.

-mounts, -mnt
lists pending mount requests for the device or devices specified. If the -long control argument is given with -mounts, the process name and write-protect status of the mount requests listed are listed as well.

-reserved, -resv
lists information for all reservations known at this time, including the reservation ID, for the devices specified.

-type STR, -tp STR
specifies the type of resource to be listed. Currently, only device types are allowed. The following device type names are valid: tape_drive, disk_drive, console, printer, punch, reader, and special.

-unattached, -unat
lists names of all loaded, unattached volumes.

-user STR
specifies the group ID of the user whose reservation is to be canceled. Use -user in conjunction with cancel.

Name: rebuild_disk

SYNTAX AS A COMMAND

```
rebuild_disk volume_name drive_name1 -copy drive_name2 {-control_args}
```

FUNCTION

rebuilds a disk by copying all information onto another disk and makes changes to partitions and/or to the VTOC as directed. You can use it only in ring 1.

ARGUMENTS

volume_name
is the name of a mounted physical volume.

drive_name1
is the source pack's drive in the form <subsys>_<nn>.

drive_name2
is the target pack's drive to be used for the copy, in the form <subsys>_<nn>.

CONTROL ARGUMENTS

-console
sends output to the bootload console instead of to the printer.

-copy
copies all information from the source pack's drive onto the target pack's drive. (Required)

-debug
is intended for system programmer use.

-dump
dumps damaged objects.

NOTES

The disk being copied must be in a logical volume that is in service (has been added to the system with the add_lv command).

This command prints out the current extent and location of partitions on the source pack, VTOC size, number of VTOC entries, and average segment length. The request loop of the init_vol command is then entered to specify the partition extents and locations of the target pack and VTOC size.

You can change some of the parameters of a volume, such as VTOC size, paging region size, and partitions, using rebuild_disk. Please note:

1. You can increase the VTOC size, as long as there is room to do so.

2. You can decrease the VTOC size, provided that there are no active VTOCEs in the truncated portion of the old VTOC. The command validates this condition and terminates with an error message if there are active VTOCEs in the portion of the old VTOC that is to be truncated. You can use the sweep_pv command with -move and -from to vacate a portion of the VTOC prior to running rebuild_disk.

The disk rebuild operation takes about seven minutes per thousand VTOC entries in use and one minute per thousand VTOC entries not in use. A progress message is reported to the bootload console as every thousand VTOC entries are processed.

The init_vol request loop assumes the same starting parameters whether it is being used for a volume initialization or a disk rebuild; you must not assume that the starting parameters of the target pack are in any way derived from the source pack.

This command queries you before destroying the label of any pack that appears to be a validly labeled pack. A message giving the pack's physical volume name and time of last use is displayed.

Name: reconfigure, rcf

SYNTAX AS A COMMAND

```
rcf function type name {-control_args}
```

FUNCTION

manipulates selected reconfigurable entities in the current configuration. You can use it only in ring 4.

ARGUMENTS

function
specifies the function to be performed. Can be one of the following values:

add
adds selected reconfigurable entities to the current configuration, making them available for use.

delete
deletes selected reconfigurable entities from the current configuration, making them unavailable for use.

type
is one of the reconfigurable entities listed below under "List of Reconfigurable Entities."

name
is the name of the item being reconfigured. Examples of names are given under "List of Reconfigurable Entities."

CONTROL ARGUMENTS

-add_all_attachments
causes all reconfigurable entities which are newly accessible to be added. This control argument can be used with the add function only.

-brief, -bf
does not print out a list of every item which is manipulated. This is the default.

-delete_all_attachments
causes all reconfigurable entities which will become inaccessible to be deleted. This control argument can be used with the delete function only.

-long, -lg
prints out a list of every item which is manipulated.

LIST OF RECONFIGURABLE ENTITIES

channel, chan, chnl
a logical channel. A channel's name is the name by which IOI knows it (e.g., a9, b23).

cpu
A Central Processing Unit. A processor's name is its tag as it appears on a cpu card in the config deck (e.g., a, b).

device, dv prph
a peripheral device. A device's name is the name by which RCP knows it (e.g., tapa_03, dskb_13, fnpc opca).

iom
an Input/Ouput Multiplexer. An IOM's name is its tag as it appears on an iom card in the config deck (e.g., a.b).

link_adapter, la
a link adapter or physical channel. This is a shorthand way of specifying a collection of logical channels. A link adapter's name is the name of its lowest channel (e.g., b28).

mpc
a Microprogrammed Peripheral Controller. This is a shorthand way of specifying a collection of link adapters (and thus, a collection of logical channels). An MPC's name is its name as it appears on an mpc card in the config deck (e.g., mspa, mtpb).

page
a page of memory. A page's name is its number. Pages are numbered starting at 0. Numbers may be given in any form acceptable to the cv_interger_string_ function. A range of pages may be specified with an expression of the form <low>:<high>. Pages to be removed must reside within a single system controller. When you delete all of an SCU's pages, the SCU itself is not deleted.

scu, mem
a System Control Unit. An SCU's name is its tag as it appears on a mem card in the config deck (e.g., a, b). When you delete all of an SCU's pages, the SCU itself is not deleted.

NOTES

For more details on dynamic reconfiguration, see the *Multics System Maintenance Procedures* manual, Order No. AM81, and the *Operator's Guide to Multics*, Order No. GB61.

EXAMPLES

The examples below assume the following config deck fragment:

```
iom b 1 nsa on
mpc mspd 607. b 28. 4
prph dskd b 28. 4 0 16. 501. 16.   <dskd_17 - dskd_32 are 501s>
mem b 2048. on                     <contains pages 0-2047>
mem a 1024. on                     <contains pages 2048-3071>
cpu c 5 on dps8 70. 32.
```

```
! rcf delete cpu c
```

deletes CPU c from the configuration.

```
! rcf delete mem a -delete_all_attachments
```

deletes SCU a and pages 2048 through 3071 from the configuration.

```
! rcf delete iom b -delete_all_attachments
```

deletes IOM b from the configuration. MPC mspd is also deleted, and if any of dskd_17 through dskd_32 were previously added, they are deleted as well.

```
! rcf add cpu c
```

adds CPU c to the configuration.

```
! rcf add mem a -add_all_attachments
```

adds SCU a and pages 2048 through 3071 to the configuration.

```
! rcf add iom b -add_all_attachments
```

adds IOM b to the configuration. MPC mspd is also added, and if any of dskd_17 through dskd_32 were previously deleted, they are added as well.

Name: recover_volume_log

SYNTAX AS A COMMAND

```
recover_volume_log pvnames {-control_args}
```

recover_volume_log

FUNCTION

recovers volume logs from dump volumes. It should be invoked only if volume logs for physical volumes to be reloaded cannot be found. Its input is a list of the latest dump volumes for the physical volumes in question as specified by the caller. This command can be used in ring 1 or ring 4.

ARGUMENTS

pvnames
: are the names of the physical volumes whose volume logs are to be recovered. All pvnames supplied must be valid physical volume names (i.e., the physical volumes must be valid members of a registered logical volume).

CONTROL ARGUMENTS

-input_volume_desc STR
: where STR is the attach description used to replace the default attach description "tape_mult_ ^a -system". The dump volume name is inserted in the attach description at the first occurrence of the string "^a" in the attach description.

-working_dir, -wd
: specifies that the volume backup databases are to be recovered relative to the working directory. The default is to recover them relative to the >ddd>volume_backup directory. This control argument can be used to recover the volume logs for physical volumes that are not part of the currently mounted storage system. This control argument is optional.

NOTES

No announcement of the recovery of any volume logs is provided until all volume logs have been recovered or the query for the dump volume name is answered with a period (".").

After a recover_volume_log operation is executed, it may be necessary to run a merge_volume_log operation.

Name: redefine

SYNTAX AS A COMMAND

```
redefine vcons old_dest new_type new_dest
```

FUNCTION

removes one destination from a virtual console and adds another. Do not confuse this command with the substty or reroute commands. This command can be used in ring 4 only.

ARGUMENTS

vcons
is the name of the virtual console for which a destination is to be changed.

old_dest
is the destination that is to be changed, i.e., removed. If old_dest is a device channel that currently has output queued for it, no more output is queued but all the queued output is printed.

new_type
is the new type of destination that is to be added. Refer to the define command.

new_dest
is the destination that is to be added.

Name: release

SYNTAX AS A COMMAND

```
release
```

FUNCTION

releases a preserved stack frame history of a fault. This command is the opposite of the debug command. This command can be used in ring 1 only.

Name: reload

SYNTAX AS A COMMAND

```
reload {-control_args}
```

FUNCTION

is used for recovery, when the intent is to make the storage system hierarchy look indentical to that which was dumped onto the tape being reloaded. It deletes segments and directories from the hierarchy that are not present on the tape, and replaces existing segments and directories with their counterparts from the tape. However, to avoid destroying useful information, it does not delete directories, and it does not replace segments and directories in the hierarchy that were modified after the time at which they were put on the tape. To ensure that ring 1 segments (such as mailboxes) are recovered correctly, the process performing the reload must be logged in at ring 1. To ensure that AIM restrictions are enforced correctly, the process performing the reload must have segment and directory privileges turned on. The reload command calls the backup_load command to do the actual reloading. This command can be used in ring 4 only.

The reload command creates the directory >reload_dir (to which the process must have sma access), and places its maps there. It doesn't automatically dprint them. The reload command causes the entire RLV to be accepted, if it isn't already accepted. If issued in ring 1, this command causes all logical volumes that were mounted at the time of the last shutdown to be automatically mounted. Quota on the reloaded directories is force-set to that specified on the tape.

The reload command is one of the commands used for hierarchy reloading and retrieving of storage system segments and directories. The other commands are:

```
backup_load
reload (Multics command)
reload_system_release
retrieve
```

Do not confuse this reload command, which is an initializer command, with the reload Multics command.

You should note that argument processing for all of the hierarchy backup commands is performed by a common argument processing procedure. The values of all arguments are remembered in static storage and remain in effect for the life of the process, unless changed by arguments given in subsequent invocations of backup commands. It should also be noted that the dumping commands and the reloading/retrieving commands are all part of the same hierarchy backup system, and argument values set by the dumping commands remain in effect for the reloading/retrieving commands and vice versa, unless overridden. However, dumping and reloading cannot be done in the same process; use the new_proc command between dumping and reloading. See "Notes on Default Arguments" below.

CONTROL ARGUMENTS

-all
causes segments to be retrieved from the tape regardless of their date/time dumped. This control argument overrides a previously given DATE argument. This is the default.

-brief_map, -bfmap
creates a map file that lists the processed entries.

-debug
disables those hphcs_ calls that set quotas and transparency switches.

-destination STR, -ds STR
specifies a destination for printing maps and error file. The default is "incremental" for maps and "error file" for error files.

-error_of
writes error messages into a file rather than printing them. The name of the error file is printed when the first error is encountered. This is the default.

-error_on
writes error messages on the user's terminal.

-first
prevents searching a tape for additional copies of a requested segment or subtree after the first copy has been retrieved.

-header STR, -he STR
specifies a heading for printing maps and error files.

-last
indicates that the last copy of a given segment or subtree on a tape or set of tapes is to be retrieved. This is the default.

-map
writes a list of the segments and directories processed into a file. This is the default.

-nodebug
enables hphcs_ calls to set quotas and the transparency switches. This is the default.

-nomap
inhibits listing of the names of processed segments and directories.

-noprimary, -npri
uses each pathname as given. The default is -primary.

-noqcheck
causes the hierarchy reload to be done with quota checking suspended. Access to hphcs_ is required. This is the default.

-noquota
inhibits resetting of quotas. See -quota. This is the default.

-noreload
inhibits actual hierarchy reloading of segments into the hierarchy. This control argument can be used with -map to create a table of contents of the tape. The -noreload control argument also causes the names that would have been reloaded to be put into the map.

-nosetlvid
inhibits the setting of the logical volume identifiers for each directory to be reloaded.

-notrim
inhibits deletion of entries in a directory. Entries can only be added or modified.

-operator STR
indicates that STR is the user's name or initials (up to 16 characters in length).

-primary, -pri
replaces all directory names in each pathname with the primary names. This is the default.

-pvname STR
indicates that segments and directories may only be reloaded onto the physical volume specified by STR.

-qcheck
causes quota restrictions to be enforced during the reload.

-quota
causes the quotas on directories being reloaded to be set to the values they had when the directories were dumped. Access to hphcs_ is required.

-reload
enables actual reloading of segments into the hierarchy. This is the default.

-request_type STR, -rqt STR
specifies an output request type for printing maps and error files. Available request types can be listed by using the print_request_types command (described in the *Multics Commands and Active Functions* manual, Order No. AG92). The default is "printer".

-setlvid
enables setting of the logical volume identifier for reloaded entries inferior to each directory reloaded. This is the default.

-trim
enables deletion of all entries in a directory not found in the copy of that directory being reloaded. This causes entries deleted from an earlier version of the directory to be deleted when a later version is reloaded. It has effect only in the case of a directory that is both on the tape and in the hierarchy. This is the default.

DATE
an argument beginning with a character other than "-", or ">" is assumed to be a date in a format acceptable to the convert_date_to_binary_ subroutine. If it can be converted successfully, then the hierarchy retriever only retrieves segments and directories dumped at or after the given date/time.

NOTES ON DEFAULT ARGUMENTS

The values of arguments given to any of the hierarchy backup commands are remembered in static storage and remain in effect for the life of the process, unless explicitly changed during the invocation of a subsequent backup command.

The following defaults are in effect for the reloader and retriever before any backup commands are given; they are not, however, reset to these values at the start of each backup command, except as noted below.

```
-all
-error_of
-map
-nodebug
-nohold
-noquota
-primary
-reload
-setlvid
-trim
```

The following defaults are set automatically at the time the respective commands are executed:

```
reload (initializer command), reload (Multics command),
  reload_system_release:
    -quota
    -trim

retrieve:
    -all
    -noquota
    -notrim

All of the above commands:
    -map
```

Name: reload_system_release

SYNTAX AS A COMMAND

```
reload_system_release {-control_args}
```

FUNCTION

loads new release tapes into the hierarchy. It should not be used for recovery. It deletes segments and directories from the hierarchy that are not present on the tape. and replaces existing segments and directories with their counterparts from the tape. This command overrides the checks employed by the reload Multics command. In other words, it allows directories to be deleted, and it allows segments and directories to be replaced in the hierarchy even if they were modified after the time at which they were put on the tape. Since it turns off the checks employed by the reload Multics command and this is remembered for the life of the process, and since there are no control arguments to turn the checks back on, use of any other backup commands in the same process as the reload_system_release command is not recommended. The reload_system_release command calls the backup_load command to do the actual reloading. These command can be used in ring 4 only.

The reload_system_release command places its map in the directory >reload_dir (to which the process must have sma access), and automatically dprints it. The reload_system_release command causes the entire RLV to be accepted, if it isn't already accepted. Quota on the reloaded directories is force-set to that specified on the tape.

The reload_system_release command is one of the commands used for hierarchy reloading and retrieving of storage system segments and directories. The other commands are:

```
backup_load
reload (initializer command)
reload (Multics command)
retrieve
```

You should note that argument processing for all of the hierarchy backup commands is performed by a common argument processing procedure. The values of all arguments are remembered in static storage and remain in effect for the life of the process, unless changed by arguments given in subsequent invocations of backup commands. It should also be noted that the dumping commands and the reloading/retrieving commands are all part of the same hierarchy backup system, and argument values set by the dumping commands remain in effect for the reloading/retrieving commands and vice versa, unless overridden. However, dumping and reloading cannot be done in the same process; use the new_proc command between dumping and reloading. See "Notes on Default Arguments" below.

CONTROL ARGUMENTS

-all
causes segments to be retrieved from the tape regardless of their date/time dumped. This control argument overrides a previously given DATE argument. This is the default.

-brief_map, -bfmap
creates a map file that lists the processed entries.

-debug
disables those hphcs_ calls that set quotas and transparency switches.

-destination STR, -ds STR
specifies a destination for printing maps and error file. The default is "incremental" for maps and "error file" for error files.

-error_of
writes error messages into a file rather than printing them. The name of the error file is printed when the first error is encountered. This is the default.

-error_on
writes error messages on the user's terminal.

-first
prevents searching a tape for additional copies of a requested segment or subtree after the first copy has been retrieved.

-header STR, -he STR
specifies a heading for printing maps and error files.

-last
indicates that the last copy of a given segment or subtree on a tape or set of tapes is to be retrieved. This is the default.

-map
writes a list of the segments and directories processed into a file. This is the default.

-nodebug
enables hphcs_ calls to set quotas and the transparency switches. This is the default.

-nomap
inhibits listing of the names of processed segments and directories.

-noprimary, -npri
uses each pathname as given. The default is -primary.

-noqcheck
causes the hierarchy reload to be done with quota checking suspended. Access to hphcs_ is required. This is the default.

-noquota
inhibits resetting of quotas. See -quota. This is the default.

-noreload
inhibits actual hierarchy reloading of segments into the hierarchy. This control argument can be used with -map to create a table of contents of the tape. The -noreload control argument also causes the names that would have been reloaded to be put into the map.

-nosetlvid
inhibits the setting of the logical volume identifiers for each directory to be reloaded.

-notrim
inhibits deletion of entries in a directory. Entries can only be added or modified.

-operator STR
indicates that STR is the user's name or initials (up to 16 characters in length).

-primary, -pri
replaces all directory names in each pathname with the primary names. This is the default.

-pvname STR
indicates that segments and directories may only be reloaded onto the physical volume specified by STR.

-qcheck
causes quota restrictions to be enforced during the reload.

-quota
causes the quotas on directories being reloaded to be set to the values they had when the directories were dumped. Access to hphcs_ is required. This is the default for the reload command.

-reload
enables actual reloading of segments into the hierarchy. This is the default.

-request_type STR, -rqt STR
specifies an output request type for printing maps and error files. Available request types can be listed by using the print_request_types command (described in the *Multics Commands and Active Functions* manual, Order No. AG92). The default is "printer".

-setlvid
enables setting of the logical volume identifier for reloaded entries inferior to each directory reloaded. This is the default.

-trim
enables deletion of all entries in a directory not found in the copy of that directory being reloaded. This causes entries deleted from an earlier version of the directory to be deleted when a later version is reloaded. It has effect only in the case of a directory that is both on the tape and in the hierarchy. This is the default.

DATE
an argument beginning with a character other than "-", or ">" is assumed to be a date in a format acceptable to the convert_date_to_binary_ subroutine. If it can be converted successfully, then the hierarchy retriever only retrieves segments and directories dumped at or after the given date/time.

NOTES ON DEFAULT ARGUMENTS

The values of arguments given to any of the hierarchy backup commands are remembered in static storage and remain in effect for the life of the process, unless explicitly changed during the invocation of a subsequent backup command.

The following defaults are in effect for the reloader and retriever before any backup commands are given; they are not, however, reset to these values at the start of each backup command, except as noted below.

```
-all
-error_of
-map
-nodebug
-nohold
-noquota
-primary
-reload
-setlvid
-trim
```

The following defaults are set automatically at the time the respective commands are executed:

```
reload (initializer command), reload (Multics command),
  reload_system_release:
    -quota
    -trim

retrieve:
    -all
    -noquota
    -notrim

All the above commands:
    -map
```

Name: reload_volume

SYNTAX AS A COMMAND

```
reload_volume -control_args
```

FUNCTION

reconstructs the contents of physical volumes using the dump volumes produced by the volume dumper facility. You can use it in ring 1 only.

CONTROL ARGUMENTS

-disk_model STR
where STR is the type of disk being reloaded. STR must be one of the following:

3380
3381
d400
d402
d451
d500
d501

If not specified, the registration information for the physical volume to be reloaded is used. This control argument is not allowed when more than one physical volume is being reloaded.

-error_on
specifies that error messages are written to the error_output switch as well as the rldr_err.mm/dd/yy.hhmm.s segment. (Default: off)

-input_volume_desc STR
where STR is an attach description for the dump volumes the reloader reads. The dump volume name is inserted in the specified attach description at the first occurrence of the string "^a" within the attach description. The default attach description is

```
tape_mult_ ^a -system
```

-manual
specifies that the dump volumes are requested by the reloader, rather than being automatically determined. If there are no more dump volumes to read, type a period.

-no_detach
specifies that, at the completion of the reload, neither the dump volume nor the physical volume are detached. This control argument is not allowed when more than one physical volume is being reloaded. (Default: off)

-no_object
specifies that segments and directories are not read from the dump volumes and thus not written to the physical volume. In this case only the VTOC is reloaded. (Default: segments and directories are read)

-operator STR
specifies the name of the user doing the reload. (Required)

-output_volume_desc STR
where STR is an attach description for the physical volume the reloader writes. The physical volume name is inserted at the first occurrence of the string "^a" in the attach description, and the type at the second occurrence. The default attach description is

```
rdisk_ ^a ^a -write -system
```

-pvname STR1 STR2...STRs
specifies the name(s) of the physical volume(s) to be reloaded. You can give it more than once. (Required)

-pvname_device STRP1 STRD1...STRPs STRDs
-pvdv STRP1 STRD1...STRPs STRDs
specifies the name(s) of the physical volume(s) to be reloaded and what device(s) the volume(s) will be on. STRPs and STRDs make up an ordered pair list of pvname (STRPs) followed by the device name (STRDs) that will contain the physical volume (e.g., -pvdv pub01 dska_00a pub02 dska_00b). This control argument is useful when reloading devices that have fixed media, and is the only way to reload a physical volume to a subvolume of a device. You can do this only with the default output attach description. You must set the device usage for "ip" by using the set_drive_usage command. If you use -pvdv, there is no need to use the Multics assign_resource command.

-restart
specifies that the reloader is restarted using control information contained in the control segment in the working directory; the suffix "control" is assumed if not specified. Use -restart only if a system failure occurs during a reload sequence. (Default: off)

*
-working_dir, -wd
specifies that the volume backup databases are to be searched relative to the working directory. You can use -wd to cause reloading of physical volumes that do not belong to the currently mounted storage system. All specified physical volumes must "belong" to the same RPV. (Default: to search relative to the >ddd>volume_backup directory)

NOTES

When you are doing a volume reload of a single physical volume that resides on an MSU 500/501 disk drive, you must use the assign_resource command to assign the disk_drive resource on which you want the physical volume to reside after the reload. For this assignment to succeed, the operator must have set both this disk_drive resource and its partner on the shared spindle to "io" with the set_drive_usage command. Since assign_resource is only available in ring 4, this restriction does not apply to ring 1 reloads.

The volume reloader can use a preinitialized disk pack (initialized by the init_vol command) as an output medium.

Name: remove

SYNTAX AS A COMMAND

```
remove channel_id1 {channel_id2 ...  channel_idN}
```

FUNCTION

controls communications channels for login, slave, autocall, or ftp service. Specifically, it directs the answering service to stop using specified channels or making them available. Any user logged in on one of these channels is logged out immediately with no message. This command can be used in ring 4 only.

ARGUMENTS

channel_idi
: are the names of the channels to be removed. (See the *Multics System Maintenance Procedures* manual, Order No. AM81, for information on channel names.) The channel_id arguments may be either terminal channel names or network channel names.

NOTES

The remove command sometimes succeeds where bump and detach fail, because it uses different mechanisms for destroying the user's process. This command should not be used unless the user complains of channel trouble and cannot be bumped. Once a channel is removed, the telephone associated with the channel will not answer and the channel cannot be used for logins again until an attach command reattaches it to the answering service.

Name: reply, r

SYNTAX AS A COMMAND

```
r source rest_of_line
```

FUNCTION

sends an input line to a specified source and sends a wakeup to that source. This command can be used in ring 4 only.

ARGUMENTS

source
is the source to which the input line is to be sent.

rest_of_line
is the input line to be sent.

Name: reregister

SYNTAX AS A COMMAND

```
reregister volume_name drive_name
```

FUNCTION

regenerates volume registration information. This command can be used in ring 1 only.

ARGUMENTS

volume_name
is the name of a physical volume.

drive_name
has the form <subsys>_<nn>, e.g., dska_02.

NOTES

This command reads the label of the volume on the specified drive name. If it is a valid label for the volume whose name is given, the registration data is refabricated from the label and the drive is left in the known state.

This command should only be used to re-create logical volume registration information if it has been lost because of a system crash or other mishap.

The reregister command can re-create the logical volume registration, physical volume registration, master directory control segment, and the link to the access control segment. If the registration data was damaged, the system administrator should be informed so that he can run the register_mdir command or correct any errors in the volume registration.

Volumes that are accepted by bootload (the volumes listed on the root config card) are registered automatically and cannot be registered manually.

Name: reroute

SYNTAX AS A COMMAND

```
reroute source stream old_vcons new_vcons
```

FUNCTION

reroutes output from a source to a new virtual console. Do not confuse this command with the redefine and substty commands. This command can be used in ring 4 only.

ARGUMENTS

source
is the name of the source of output.

stream
is the name of the stream on which output is being carried.

old_vcons
is the name of the virtual console that was receiving output.

new_vcons
is the name of the virtual console to which output is now to be routed.

NOTES

This command changes the routing table so that the output from "source" on stream "stream" is sent to the virtual console "new_vcons" instead of "old_vcons." The reroute command is equivalent to typing the sequence:

```
deroute source stream old_vcons

route source stream new_vcons
```

This command can not be used to reroute RCP messages and other syserr traffic from the bootload console to an initializer terminal.

Name: reset

SYNTAX AS A COMMAND

```
reset
```

FUNCTION

resets the answering service. If an unexpected fault happens, it is possible that the initializer will not react correctly. The reset command is therefore issued to reset the initializer process and restart all message coordinator terminals. This command can be used in ring 4 only.

Name: route

SYNTAX AS A COMMAND

```
route source stream vcons
```

FUNCTION

sends the output from a specific source to a designated virtual console. This command can be used in ring 4 only.

ARGUMENTS

source
is the name of the source of output.

stream
is the name of the stream on which output is being carried.

vcons
is the name of the virtual console to which output is to be routed.

NOTES

If no entry for source or for stream under source exists in the message routing table (MRT), one is created. There may be up to 16 sources. Each source may have up to eight streams, and each stream may have up to eight virtual consoles. Vcons must have been previously defined. It is added to the virtual console list for stream.

Name: salvage_dirs, salv

SYNTAX AS A COMMAND

```
salv
```

FUNCTION

salvages the directory hierarchy. The salvaging is performed in the initializer process. This command can be issued in ring 1 only. All physical volumes of the RLV must be accepted before this command can operate. If invoked, it attempts to mount all RLV volumes. This command should be used only during critical storage system recovery operation.

Salvaging is normally performed automatically by the system when needed. If optional salvaging is desired (e.g., for directory compaction) the x repair exec command should be used.

Name: salvage_vol

SYNTAX AS A COMMAND

```
salvage_vol volume_name drive_name {-control_args}
or
salvage_vol -all {-control_args}
```

FUNCTION

volume salvages a physical volume. This command can be used in ring 1 or ring 4 and should only be used at the direction of the Site Administrator for storage system recovery. (See the *Multics System Maintenance Procedures* manual, Order No. AM81, for more information.)

ARGUMENTS

volume_name
: is the name of a physical volume.

drive_name
: has the form <subsys>_<nn>.

CONTROL ARGUMENTS

-all
all known and assumed physical volumes are salvaged.

-console
outputs to the syserr console instead of to the printer.

-copy drive
salvages an inactive copy of an active volume on a drive.

-debug
for system programmer use only.

-dump
dumps damaged objects.

-noconsole
overrides the salv configuration card. This card can supply some default options.

-nodebug
overrides the salv configuration card.

-nodump

Name: set_drive_usage, sdu

SYNTAX AS A COMMAND

```
sdu drive_name usage_type
```

FUNCTION

converts a disk drive from storage system use to user input/output use, or vice versa, while the system is running. You can use it in ring 1 or 4.

ARGUMENTS

drive_name
is the name of the drive whose use is to be changed. It has the form <subsys>_<nn>, e.g., dska_05.

usage_type
can be one of the following:

io
converts the specified drive to user input/output use. This drive must be a storage system drive. The drive cannot have a known or accepted physical volume on it; if one is present, use the del_lv and del_vol commands to remove it before issuing io. Once converted, this drive remains a user input/output drive until you issue sdu again.

storage_system, ss
converts the specified drive to storage system use. This drive must be a user input/output drive. The drive cannot be assigned to any process; you can use the rcp command to find out. You can use the "x deny" command to forcibly remove the drive from any user process to which it may be attached. Once converted, this drive remains a storage system drive until you issue sdu again.

NOTES

Drives in use by the storage system appear as "storage system" to rcp, which means that they are unavailable for users. Drives assigned to user input/output use appear as "(io drive)" to the list_disks command.

Name: set_pdir_volumes

SYNTAX AS A COMMAND

```
set_pdir_volumes lv_name1...lv_nameN
```

FUNCTION

defines the set of logical volumes available for process directory segments. You can use it in ring 4 only.

ARGUMENTS

lv_nameN
is the name of one or more mounted public logical volumes to be used for process directory segments.

NOTES

Process directories are assigned to logical volumes in proportion to the number of physical volumes in each logical volume. Volumes specified by this command must have adequate space available to hold that proportion of process directory segments.

Any existing set of process directory volumes is replaced by this command. Only mounted public logical volumes can be made available for process directory segments. If none of the specified volumes is suitable, the existing set of process directory volumes is not replaced. Invoke this command at each bootload operation; it is, however, automatically invoked at each bootload if your site is running the system-supplied system_start_up.ec.

See the add_pdir_volume, del_pdir_volume, and vacate_pdir_volume commands.

Name: shift

SYNTAX AS A COMMAND

```
shift
or
shift {shift_number} {time}
or
shift auto
```

FUNCTION

sets and prints the shift and the time of the next scheduled shift change, thus overriding the values in the shift table. See the system administrator before using it. You can use it in ring 4 only.

ARGUMENTS

shift_number
: is the number of the shift to change to. This argument must be from 0 to 7, and must also be one of the shifts normally used at your site, for which billing rates are defined in installation_parms.

time
: is the time of the next shift change. It is a clock reading, acceptable to convert_date_to_binary_. Enclose it in quotes if it contains spaces. This argument cannot be more than one week in the future.

auto
: is a literal string that causes the system to use the value of the shift and the shift change time specified in the shift table.

NOTES

When the system is started up, a shift command from a previous bootload remains in effect until the specified time.

To change the shift without changing the time, type:

```
! shift shift_number
```

To change the shift and the shift change time to the regularly scheduled values (specified in the shift table), type:

```
! shift auto
```

To cause the shift and the next shift change time to be displayed, type:

```
! shift
```

The message:

```
shift shift_number until time
```

is printed in response to all three forms of the shift command. If the shift or the time have been changed, their new values are displayed.

The best time to execute the shift command is shortly (about one minute) before a regularly scheduled shift change. If the shift command must be executed in the middle of a shift, then notice should be given by executing it at the beginning of that shift, using the regular shift number and the time of the anticipated unscheduled shift change.

EXAMPLES

If shift 1 runs from 0800 to 1800 every day, and you want to run shift 3 from 1800 Sunday to 0800 Tuesday because of a Monday holiday, then at 1759 Sunday, type:

```
! shift 3 "Tuesday 0800.0"
```

However, if the change to shift 3 is to occur at 1500 Sunday, then at 0759 Sunday, type:

```
! shift 1 1500.0
```

and then at 1459 Sunday, type:

```
! shift 3 "Tuesday 0800.0"
```

The reason for the above procedures is that some users' programs keep track of the shift and the time it is to change, and update their shift information at each change (that they know about). Using the shift command a minute before a scheduled change causes these programs to operate with incorrect information for less than a minute, while using it just after a scheduled change leaves these programs with incorrect information for the remainder of the shift.

Name: shutdown, shut

SYNTAX AS A COMMAND

```
shut
```

FUNCTION

is used after all users are logged out to make a normal exit from Multics. If any users are still logged in, the system asks if you really want to shut down. Usually, you should not. Answering "yes" causes the system to shut down. Answering "no" returns the system to initializer command level so that the logged in users can be bumped. This command can be used in ring 1 or ring 4.

NOTES

The shutdown command should be issued only after a stop command, unless this is the end of a special session in which a startup or a multics command was never issued.

Name: sign_off

SYNTAX AS A COMMAND

```
sign_off
```

FUNCTION

used by operators to sign off as operator on a message coordinator terminal. Further commands on this terminal are not accepted until the terminal is signed on again with the sign_on command. Use it at the terminal from which you previously entered a sign_on command. You can use it only in ring 4.

Name: sign_on

SYNTAX AS A COMMAND

```
sign_on {Person_id}
```

FUNCTION

used by operators to sign on as an operator on a message coordinator terminal. It provides an identification and authentication function for persons logging in as operators. You can use it only in ring 4.

ARGUMENTS

Person_id
: is the user's registered personal identifier. The command prompts for this argument if you don't give it.

NOTES

The command prompts for a password. You should respond with your assigned password. Typing a password of "quit" terminates the sign_on attempt.

Only those users with the "operator" attribute enabled in the PNT may sign on as operators.

Name: standard, stan

SYNTAX AS A COMMAND

```
stan
```

FUNCTION

causes the initializer to cross to the user ring from ring 1 so that additional commands may then be issued. This command causes the entire RLV to be mounted, if not already accepted. This command can be used in ring 1 only.

Name: start_mpx

SYNTAX AS A COMMAND

```
start_mpx name
```

FUNCTION

causes a multiplexer that is already initialized (loaded) to be made active, and all channels configured on it to be listened to. This command can be used after a load_mpx command that was issued with the -no_start argument, or after a stop_mpx command, to reverse the effect of the latter command. For a multiplexer that is not initialized, the load_mpx command must be issued before the start_mpx command. This command can be used in ring 4 only.

ARGUMENTS

name
: designates the multiplexer that is to be activated. All the channels in the CDT that match the starname name.** will be listened to.

Name: startup, star

SYNTAX AS A COMMAND

```
star
```

FUNCTION

causes all logical volumes that were mounted at the time of the last shutdown to be automatically mounted. It also initializes the answering service. This command is usually the first thing you type when you are bringing up Multics for a normal user session. This command can be used in ring 1 or ring 4.

All parts of the system_start_up.ec are executed in response to the startup command. Part 1 (those commands executed before the answering service is started) is executed before answering service initialization. Part 2 (those commands executed after the answering service is ready but before logins are accepted) is then executed; then all terminal channels are instructed to answer. After the terminal channels have been started, part 3 (those commands executed after the telephone lines are answered) is executed.

Name: stop

SYNTAX AS A COMMAND

```
stop
```

FUNCTION

begins the shutdown process. The stop command also executes a "word shutdown" command, so that no more users may log in. The system automatically executes a "bump * * 3" to bump all users, giving them a three minute warning. Users with the nobump privilege are not bumped. This command can be used in ring 4 only.

EXAMPLES

Typing the stop command causes all users to get a message of the form:

```
**********
From Operator:  Multics will shut down in 3 minutes
**********
```

The next step in the shutdown procedure is to let the absentee users finish up and log out, and then log the daemons out. Then, type the shutdown command.

Issuing the stop command again will not cause the bump to be signalled again or the message to be resent. If all users are logged out, the stop command responds:

```
admin:  all users are out.  You may shut down.
```

If you type stop and then change your mind, type the following:

```
! unbump * * "shutdown cancelled"

! word login
```

When you set a down time, an automatic abs stop is scheduled for 20 minutes before that time. Absentee jobs are given an additional 10 minutes to complete after all interactive users have been bumped. In explaining initializer messages elsewhere, it is usually stated that the time for an automatic abs stop is 30 minutes before the system is to shut down.

Name: stop_mpx

SYNTAX AS A COMMAND

```
stop_mpx name
```

FUNCTION

causes the specified multiplexer to be made inactive. This makes the multiplexer stop listening for further calls on its dialup channels if it is up and running; it inhibits listening to the channels after a bootload, if one is in progress; it prevents automatic reloading of the multiplexer if it crashes (or is crashed by the dump_mpx command); and if issued before startup, it prevents the loading of the multiplexer during startup. Users currently logged in over the multiplexer channels are not affected. This command can be used in ring 4 only.

ARGUMENTS

name
: specifies the multiplexer that is to be made inactive.

NOTES

To bump all users logged in over channels of this multiplexer (see the bump command for more information), type:

```
! stop_mpx name
! bump mpx name {mm} {message}
```

This sequence of the two commands stop_mpx and bump mpx is the recommended method of taking a multiplexer out of service while Multics remains up.

Hardwired channels normally appear to be dialed up at all times; such channels are not affected by the stop_mpx command.

The effect of the stop_mpx command can be reversed by issuing the start_mpx command, provided that the multiplexer remains running during the time it is out of service. If it does not remain running, the load_mpx command must be used to return it to service.

Name: substty

SYNTAX AS A COMMAND

```
substty channel_id1 channel_id2
```

FUNCTION

substitutes one message coordinator terminal channel for another. Do not confuse this command with the redefine or reroute commands. This command can be used in ring 4 only.

ARGUMENTS

channel_id1
: is the name of the communication channel to be dropped.

channel_id2
: is the name of the channel to be attached (see the *Multics System Maintenance Procedures* manual, Order No. AM81, for information on channel names). All ouput queued for channel_id1 is placed in the queue for channel_id2. This command cannot be used on multiplexer channels.

EXAMPLES

If the terminal being used is connected to channel_id a.h100, and its printing mechanism jams, you can take the following steps. To switch all output to another device, e.g., a.h102, dial the terminal connected to channel_id a.h102 to the initializer (if it is controlled by the answering service) and then type:

```
! substty a.h100 a.h102
```

If the bootload console is being used as one of the message coordinator terminals and it breaks, you can switch its output to another terminal, as in the above example, using "otw_" for channel_id1:

```
! substty otw_ a.h102
```

Switch the bootload console's output to another terminal only if there is no alternate bootload console.

Name: sysid

SYNTAX AS A COMMAND

```
sysid name
```

FUNCTION

changes the system ID typed out by the who command and when users dial up. You can use it in ring 4 only.

ARGUMENTS

name
    is the new system ID.

Name: terminate, term

SYNTAX AS A COMMAND

```
term target {message}
```

FUNCTION

causes the initializer to destroy a user's process and create a new one. It has the effect of a new_proc command given in the user's process. If the user's process is not currently connected to a terminal, it is logged out. This command can be used in ring 4 only.

ARGUMENTS

target
must be one of the following:

Person_id.Project_id, Person_id Project_id
terminates the user with the specified name and project. The star convention may be used for either or both names.

channel_id
terminates the user on the channel whose name is channel_id. These names can be either terminal channel names or network channel names. (See the *Multics System Maintenance Procedures* manual, Order No. AM81, for information on channel names.)

mpx name
terminates all users whose terminals are connected through the multiplexer specified by name. The name argument must be the name of a multiplexer or an FNP.

message
is the message string that is printed on the user's terminal.

NOTES

The terminate command is only to be used if the user requests that his process be terminated. This command can be used to bump a user whose process is not currently connected to a terminal, even if the user's process has the "nobump" attribute.

EXAMPLES

To give a new process to Vleck.Van, type:

```
! terminate Vleck.Van
```

Name: unbump

SYNTAX AS A COMMAND

```
unbump target {message}
```

FUNCTION

cancels a request to bump a user. It cannot be used unless the user has been given some grace time in which to clean up and log out. This command can be used in ring 4 only.

ARGUMENTS

target
must be one of the following:

Person_id.Project_id, Person_id Project_id
unbumps the user with the specified name and project. The star convention may be used for either or both names.

channel_id
unbumps the user on the channel whose name is channel_id. These names can be either terminal channel names or network channel names. (See the *Multics System Maintenance Procedures* manual, Order No. AM81, for information on channel names.)

mpx NAME
unbumps all users whose terminals are connected through the multiplexer specified by name. The name argument must be the name of a multiplexer or an FNP.

message
is a message string that is to be printed on the user's terminal.

Name: undefine

SYNTAX AS A COMMAND

```
undefine vcons old_dest
```

FUNCTION

removes a destination from a virtual console. This command can be used in ring 4 only.

ARGUMENTS

vcons
is the name of the virtual console from which a destination is to be removed.

old_dest
is the name of the destination to be removed from the specified virtual console.

NOTES

If vcons is left with no destinations as a result of the undefine command, and output is routed to this virtual console, then the output is printed on the bootload console.

Name: unload

SYNTAX AS A COMMAND

```
unload volume_type volume_name
or
unload volume_type -control_arg
```

FUNCTION

unloads volumes that are loaded and not in use. You may specify which inactive volume is to be unloaded or that all inactive volumes are to be unloaded. This command can be used in ring 4 only.

ARGUMENTS

volume_type
is "tape_vol".

volume_name
is the name of the tape volume to be unloaded.

CONTROL ARGUMENTS

-all
specifies that all currently inactive tape volumes are to be unloaded.

NOTES

The unload command does not allow you to unload a volume that is in use.

Name: unlock_mca

SYNTAX AS A COMMAND

```
unlock_mca mca_number
```

FUNCTION

unlocks (enables) input to the maintenance channel adapter (MCA) specified by the argument. You can use it only in ring 4.

ARGUMENTS

mca_number
is the decimal number of the MCA to be unlocked.

Name: vacate_pdir_volume

SYNTAX AS A COMMAND

```
vacate_pdir_volume lv_1 lv_2...lv_N {-control_args}
```

FUNCTION

removes per-process segments (segments in process directories) of the specified logical volumes. You can use it in ring 4 only.

ARGUMENTS

lv_s
specifies the names of the logical volumes to be vacated.

CONTROL ARGUMENTS

-long, -lg
prints a message on the bootload console for each process directory and for each segment within that directory that are processed. If you give no -lg, the messages appear in the answering service log only.

NOTES

Per-process segments are spread evenly over the remaining volumes in the PDIR volume set. Before this operation begins, all volumes specified to this command are inhibited from having any more per-process segments created on them. Once the operation is complete, you can delete the logical volumes that have been vacated.

This command continues to process logical volumes as long as at least one logical volume remains in the PDIR volume set.

For every process directory whose segments are moved, the command writes a message in the answering service log, giving the pathname of the process directory, the User_id of the user to whom the per-process segments belong, and the name of the logical volume from which the process directory is being moved.

See the add_pdir_volume, del_pdir_volume, and set_pdir_volumes commands.

EXAMPLES

```
vacate_pdir_volume Temp1 Temp2
```

Name: warn, w

SYNTAX AS A COMMAND

```
w User_id message
```

FUNCTION

outputs a message onto a user's terminal, interrupting whatever other output is being printed. You can use it in ring 4 only.

ARGUMENTS

User_id
: specifies the user to whom the message should be sent. You can use the star convention. Give User_id in this form:

```
Person_id.Project_id
```

message
: is a string containing the text of the message to be sent to the user.

NOTES

If a user to be warned has specified -no_warning (-nw) at login, no warning message is sent to him and you are told that the user could not be warned.

Don't use warn for casual communication, since the message is forced into anything the user is typing, damaging output. Use it when the system is coming down shortly, when is likely to crash, or when you have an urgent message for a user you can't reach by phone.

EXAMPLES

To send a message to Bach.Multics, type:

```
warn Bach.Multics Can't find your tape.  Call X7739
```

Name: who

SYNTAX AS A COMMAND

```
who {-control_args} {User_ids}
```

FUNCTION

lists the users who are logged in, along with information. You can use it in ring 4 only.

CONTROL ARGUMENTS

-absentee, -as
: prints the ratio of absentee users logged in to the number of absentee slots currently available, and then lists the absentee users.

-channel chn_id, -chn chn_id
: lists only interactive users whose channel name matches chn_id, or daemon users whose source name (e.g., prta, vinc) matches chn_id, or absentee users whose absentee name (e.g., abs1) matches chn_id. The chn_id argument can be a starname, to cause several users to be listed.

-daemon, -dmn
: prints the number of currently active daemon processes and then lists them.

-group {name}, -gp {name}
: prints a list of users that fall under the specified load control group (see "Notes").

-interactive, -ia
: prints a list of all users having current interactive processes.

-name, -nm
: sorts the users by name.

-no_header, -nhe
: suppresses column headings and load control heading from the printed output.

-pdir_volume {lv_name}, -pdv {lv_name}
either includes in the output the name of the logical volume containing the user's process directory segments (if you give no lv_name) or prints information about only the users whose process directory segments are on the volume specified by lv_name.

-process_id, -pid
includes the process_ids for the listed processes.

-project, -pj
sorts the users by project.

-secondary, -sc
prints a list of all users having currently active secondary user processes.

User_ids
are access control names in one of the following forms:

Person_id.Project_id
lists all users logged in with the specified Person_id and Project_id.

Person_id
lists all users logged in with the specified Person_id.

.Project_id
lists all users logged in with the specified Project_id.

NOTES

The list of users shows Person_id, time of login, terminal ID, weight, channel_id, load control group, and flags. Anonymous users are flagged with a "*" before the user name.

Users who are not listed by a who command issued by "normal" users are flagged with an "N". Users with the nobump privilege are flagged with "+". Users who may be bumped by others in their project but whose "grace" has not run out are flagged with ">". Users who have been bumped but still have some of their warning time left before automatic logout are flagged with "X". Disconnected or suspended processes (interactive or absentee) are indicated by "DS". "S" indicates that the user has secondary status. The absence of a flag indicates a user with primary status.

Absentee users are listed after interactive users. Each absentee is flagged with "A", the user_id is followed by the name of the absin file in parentheses, and the queue and slot number are indicated instead of terminal and channel names. Daemon users are listed last; each daemon is flagged with "D" and its source identifier shown.

EXAMPLES

The following is a sample of the output produced when you invoke who without arguments:

```
    Login at      TTY    Load  Chan    Group    PNDS  User ID

10/20/81 1206.8   none   1.0   c.h118  System         Opr.Operator
10/21/81 0957.3   H19    1.0   a.1002  SysProg > DS   SHawkins.Multics
         1007.9   none   1.0   c.h003  Fourth   S     CBrinkley.MNA
         1020.6   none   1.0   a.h022  SysAdm   +N    Aulin.SysAdmin
         1021.2   none   0.5   a.h100                 *Student43.Class
         1132.7   Q 3    1.0   abs1    Third    A     Blackbird.STY
         1133.6   cord   0.5   cord    Dummy    D     IO.SysDaemon
         2030.3   bk     1.5   bk      System   D     Backup.SysDaemon
```

Name: word

SYNTAX AS A COMMAND

```
word {login_word} {message}
```

FUNCTION

changes the login word, or the dialup buffer typed out when a user dials up, or both. You can use it only in ring 4.

ARGUMENTS

login_word
: is a new login word that is set. If you set the login word to shutdown, no users are allowed to log in; if a user dials up, he is told that the system is shutting down and his terminal is hung up immediately. If you give no login_word, the current one is displayed along with the message if you gave one.

message
: is a message string to which the dialup buffer is set. If you give it, and if login_word is "login," the message buffer is reset; if login_word is "shutdown," the message buffer is set to "Multics is shutting down"; if login_word is anything else, the message buffer is set to "Special session in progress."

EXAMPLES

To leave the login word as normal, but to type a message to all users at login, type:

```
! word login Only one CPU until 1300
```

To set up a special session, type:

```
! word secret Test session until 0945
```

To see the current values of the word and the message, type:

```
! word
```

SECTION 5

INITIALIZER EXEC COMMANDS

The commands in this section can be used only when communicating with the initializer process. They are defined within an exec_com (admin.ec) and require that you type "exec" or "x" prior to typing the command. The usual method of communicating with the initializer process is through an initializer terminal. The bootload console is used as the first initializer terminal. When Multics is first booted and the initializer process is active in ring 1, the initializer process uses the bootload console for input/output. Once the initializer process moves to ring 4, the administrator can select another terminal as the inititalizer terminal. (The administrator can also select multiple terminals to be used as initializer terminals.) If the administrator does not explicitly select an initializer terminal, the bootload console is used as the initializer terminal.

It is important to note that the commands in this section can only be used when the initializer is operating in ring 4.

(See Section 1 for a description of how to use the initializer exec commands when not at an initializer terminal; see also Section 1 for a description of how to submit a non-initializer command to the initializer process.)

Name: attended, attend

SYNTAX AS A COMMAND

```
x attend
```

FUNCTION

reverses the effect of the unattended command. It adds the tape drives, turns off the unattended flag in the BCE flagbox, and clears the "Unattended service" greeting message.

Name: auth

SYNTAX AS A COMMAND

```
x auth device authentication_characters
```

FUNCTION

authenticates a tape or disk volume, i.e., verifies that the correct volume has been mounted. The system asks you to authenticate the volume whenever the volume name in the volume label does not match the volume name that was requested.

ARGUMENTS

device
: is the name of the device on which the volume is mounted.

authentication_characters
: is an encoded form of the volume name, consisting of three alphabetic characters for tape drives, or the characters described in "Notes" below for disks.

NOTES

Use of the volume authentication feature requires that labels displaying the volume name and corresponding authentication characters be pasted on each tape volume in use at the site. See the make_volume_labels command. The effect of typing the correct authentication characters is to indicate to the system that you have mounted a volume whose name matches the requested name, and that it is safe to ignore the volume name on the label.

The authentication characters for disk volumes correspond to the label type specified in the authentication request message. The allowable characters are: "ss", "io", "urd", and "urg" corresponding to label types of Storage System, IO, UnReaDable, or UnReGistered respectively. If you wish to deny authentication you may use the authentication characters "no".

If authentication is requested for a tape volume that does not have authentication characters displayed on it, an authentication character string consisting of three asterisks (***) may be given, to force authentication to be bypassed for that volume. Routine use of this feature should be discouraged because it lessens the security of the system. If none of the tape volumes at the site have authentication characters displayed on them, the authentication feature can be disabled by the system administrator until the tape volumes have been properly labeled.

EXAMPLES

```
RCP:  Mount reel inc-9 with ring on tape_02 for Backup.SysDaemon.
RCP:  Authenticate tape_02.  It has Multics label inc_9.

!  x auth tape_02 CVQ
```

In this example, Backup.SysDaemon has called for a tape named inc-9. The previous user of that tape spelled its name (incorrectly) with an underscore (inc_9) rather than a hyphen, and that name was written in the volume label. The "real" name of the tape is inc-9 (with a hyphen) and the authentication characters, CVQ, correspond to that name. By supplying the authentication characters using the x auth command, the operator has verified that the tape that was mounted is really inc-9 (with a hyphen) even though the volume label claims it is inc_9 (with an underscore).

```
RCP:  Mount Pack k202 without protect on dska_02 for Joe.Multics.a.
RCP:  Authenticate dska_02 for Joe.Multics.a.
      It has UnReaDable label.  (User requested volume k202).

!  x auth dska_02 urd
```

In this example Joe.Multics.a asked for disk pack k202 to be mounted on dksa_02. The pack that was mounted had an unreadable label so RCP asked for authentication. By supplying the authentication "urd" the operator has verified that the mounted disk pack is really k202 and may be used as an io disk by Joe.Multics.a.

Operator authentication is necessary for all I/O disk mounts. The authentication message will specify the label type of the disk pack as either:

```
Storage System
copy of Storage System
IO
UnReaDable
UnReGistered
```

You must use the authentication code which corresponds to the capitalized letters in the label type, although you may type them in lower case. Thus, the authentication codes are:

```
ss
io
urd
urg
```

You may deny authentication if an incorrect pack was mounted (or for any other valid reason) by using the authentication code "no".

```
RCP: Authenticate dska_01 for Johnson.SysAdmin.a
     It has UnReGistered label k202.

!  x auth dska_01 urg
```

Name: auto

SYNTAX AS A COMMAND

```
x auto mode
```

FUNCTION

turns the automatic reboot mode on or off. When auto reboot mode is on, the system automatically takes a dump and (if possible) reboots itself after a crash. When auto reboot mode is off, the system pauses in BCE awaiting operator action after a crash.

ARGUMENTS

mode
can be either "on" or "off".

EXAMPLES

```
x auto on
```

turns on automatic reboot mode.

Name: cat

SYNTAX AS A COMMAND

```
x cat initials tape {date time}
```

FUNCTION

causes the hierarchy dumper to perform a catchup dump. If the incremental hierarchy dumper is running, it must be ended (r bk end_dump) before the catchup dump can begin. When the catchup dump is completed, the incremental dump cycle is automatically started. The hierarchy subtrees dumped are the same ones that are dumped by the incremental dumper; the same dump control file is used for both.

ARGUMENTS

initials
are the user's name or initials (up to 16 characters).

tape
is the name of the first tape to be used.

date time
is the optional date and time from which the catchup dump is to be made. The default is to dump all objects modified since midnight of the day preceding the current day (i.e., between 24 and 48 hours ago).

EXAMPLES

```
x cat ebw cat-1a 12/14/77 1930.0
```

performs a catchup dump of all segments (in the specified subtrees) modified since 12/14/77 at 19:30. The first tape to be written will be cat-1a. The user's initials are ebw.

Name: complete_volume, vcomp

SYNTAX AS A COMMAND

```
x vcomp initials
```

FUNCTION

causes the volume dumper to perform a complete volume dump. A new Volume_Dumper.Daemon process is logged in to perform the dump; thus, it can be run concurrently with other volume dumper operations.

ARGUMENTS

initials
    are the user's name or initials.

NOTES

The tapes to be used for the dump are selected automatically by the volume dumper. The oldest available tapes, in a pool reserved for the use of the volume dumper, are selected. If no tapes in the pool are available, you are asked for a new tape and it is added to the pool.

EXAMPLES

```
x vcomp ebw
```

where the user's initials are ebw.

Name: consolidated_volume, vcons

SYNTAX AS A COMMAND

```
x vcons initials
```

FUNCTION

causes the volume dumper to perform a consolidated volume dump. A new Volume_Dumper.Daemon process is logged in to perform the dump; thus it can be run concurrently with other volume dumper operations. See the note under the complete_volume command describing the volume dumper tape pool.

ARGUMENTS

initials
    are the user's name or initials.

EXAMPLES

```
x vcons ebw
```

where the user's initials are ebw.

Name: copy_dump

SYNTAX AS A COMMAND

```
x copy_dump
```

FUNCTION

copies all dumps from the DUMP partition into the hierarchy (into the directory >dumps). Use of this command is usually unnecessary, since a dump, if present in the DUMP partition, is copied into >dumps automatically at startup time. However, there are some circumstances (e.g., a crash just before a scheduled shutdown, or a crash during startup) where use of this command may be advisable.

This command operates properly before startup, provided that the standard command has been issued. The DUMP partition has limited space for dumps. Therefore, dumps should be copied out as soon as they are produced to make room for new dumps.

Name: delete_dump, dd

SYNTAX AS A COMMAND

```
x dd number
```

FUNCTION

deletes a dump from >dumps. It should be used as directed by the system administrator and system programmers, to delete dumps that have been printed or analyzed by online dump analysis tools.

ARGUMENTS

number
is the number of the dump to be deleted.

EXAMPLES

```
x dd 55
```

deletes dump number 55 from >dumps.

Name: deny

SYNTAX AS A COMMAND

```
x deny device
```

FUNCTION

forcibly unassigns an I/O device from a user process. It should be used when the device is not operating properly or when it is not possible to mount the requested volume on the device. This causes the process to be signaled, allowing the program using the device to take some appropriate action rather then waiting indefinitely for the device to respond.

ARGUMENTS

device
is the name of the device to be unassigned.

EXAMPLES

```
x deny tape_01
```

Name: echoplex

SYNTAX AS A COMMAND

```
x echoplex {}
```

FUNCTION

puts an initializer terminal in or out of echoplex (full duplex) mode. It cannot be used on the bootload console.

ARGUMENTS

" "
causes an initializer terminal to be taken out of echoplex mode.

EXAMPLES

```
x echoplex
x echoplex
```

put an initializer terminal into, and out of, echoplex mode, respectively.

Name: end_dump

SYNTAX AS A COMMAND

```
x end_dump source_id
```

FUNCTION

terminates the specified volume or hierarchy incremental dumper daemon, detaching the tape and cleaning up.

ARGUMENTS

source_id
: is the source_id of the Backup.SysDaemon (usually "bk") or the Volume_Dumper.Daemon (usually vinc).

Name: inc

SYNTAX AS A COMMAND

```
x inc initials tape
```

FUNCTION

causes the hierarchy dumper to perform an incremental dump and then set a timer that causes incremental dumps to be performed every 60 minutes. The subtrees to be dumped are those listed in the segment sys_dirs.dump in Backup.SysDaemon's working directory.

ARGUMENTS

initials
: are the user's name or initials.

tape
: is the name of the first tape to be used.

EXAMPLES

```
x inc ebw inc-9
```

causes the incremental dumper to start. The first tape used will be inc-9. The user's initials are ebw.

Name: incremental_volume, vinc

SYNTAX AS A COMMAND

```
x vinc initials
```

FUNCTION

causes the volume dumper to perform an incremental volume dump and then set a timer that causes incremental volume dumps to be performed every 60 minutes. See the note under the complete_volume command describing the volume dumper tape pool.

ARGUMENTS

initials
are the user's name or initials.

NOTES

The command "r vinc end_volume_dump" must be given to terminate an incremental volume dump properly, before the dumper process is logged out. Failure to do this causes unpredictable errors to occur when the dumper process is logged out.

EXAMPLES

```
x vinc ebw
```

where the user's initials are ebw.

Name: io

SYNTAX AS A COMMAND

```
x io
```

FUNCTION

causes the printer daemons to start operating, by initializing the I/O coordinator process and a driver process for prta.

Name: io1

SYNTAX AS A COMMAND

```
x io1
```

FUNCTION

causes the printer and punch daemons to start operating, by initializing the I/O coordinator and driver processes for prta and puna. *

Name: meter

SYNTAX AS A COMMAND

```
x meter
```

FUNCTION

sets up the system in preparation for a performance metering run. It should only be used at the direction of the system administrator or system programmers. *

Name: print_queues, pq

SYNTAX AS A COMMAND

```
x pq {ldr_args}
```

FUNCTION

causes the number of requests in each of the dprint queues to be displayed.

ARGUMENTS

ldr_args
: are any of the arguments acceptable to the list_daemon_requests command, described in the *Multics Commands and Active Functions* manual, Order No. AG92.

Name: punch, puna

SYNTAX AS A COMMAND

```
x puna
```

FUNCTION

causes the punch daemon to start operating, by initializing a driver process for puna. It is assumed that the I/O coordinator has already been started. See the io and io1 exec commands.

Name: punch_end, end_punch

SYNTAX AS A COMMAND

```
x punch_end
```

FUNCTION

halts the puna driver and detaches the card punch device.

Name: punch_restart

SYNTAX AS A COMMAND

```
x punch_restart
```

FUNCTION

attaches the card punch device and starts the puna driver (i.e., it reverses the effect of the end_punch command).

Name: read_cards, cards, rc

SYNTAX AS A COMMAND

```
x rc
```

FUNCTION

logs in Card_Input.Daemon and causes that daemon to attach the card reader and start reading cards.

Name: repair

SYNTAX AS A COMMAND

```
x repair key dir nprocs {salvager_arguments}
```

FUNCTION

starts (or stops) a multiprocess hierarchy repair. The repair can consist of an online directory salvage, a quota_used correction, or both. The repair may be performed over the entire hierarchy or any subtree. Up to 36 Salvager.SysDaemon processes may participate in the repair, so that it is completed as rapidly as possible. This command should only be used at the direction of the system administrator or the system programmers.

ARGUMENTS

key
can be one of the following:

salv
performs a directory salvage

quota
performs a quota_used correction

salvquota
performs both

stop
stops a repair that is in progress

dir
is the directory at the top of the subtree to be repaired.

nprocs
is the number of processes to be used (1 to 36).

salvager_arguments
are arguments acceptable to the salvag_dir command. The default is the single argument, -compact.

NOTES

Any of these operations automatically collates and sorts all output, and dprints the result to "SALV-OUTPUT" and "SALV-ONLINE", being salvager output and online error messages and quota_corrections, respectively. The outputs are kept in the directory >udd>SysDaemon>Salvager, and automatically deleted when they become two weeks old.

Name: reprint, rep

SYNTAX AS A COMMAND

```
x rep segment user
```

FUNCTION

reenters a dprint request. It can be used to reprint a user's request if the original copy was damaged in some way during printing and the damage was not noticed in time to have the I/O daemon restart the request.

ARGUMENTS

segment
is the full pathname of the segment to be printed.

user
is the user's name and project; the first 12 letters are printed in large letters on the head sheet.

EXAMPLES

```
x rep >udd>Operator>ebw>system_usage Winslow.Operator
```

causes a copy of the specified segment to be reprinted for Winslow.Operator.

Name: reset_phcs_access, rpa

SYNTAX AS A COMMAND

```
x rpa Person_id.Project_id
```

FUNCTION

removes a user from the ACL of phcs_. It should be used at the direction of the system administrator, when a user who has been given access to phcs_ temporarily no longer needs it. See the description of the set_phcs_access exec command.

ARGUMENTS

Person_id.Project_id
is the User_id of the user to be removed from the ACL of phcs_.

EXAMPLES

```
x rpa Greenberg.Multics
```

removes Greenberg.Multics from the ACL of phcs_.

Name: reset_tabs, tabs, rt

SYNTAX AS A COMMAND

```
x rt
```

FUNCTION

sets the terminal type of an initializer terminal to TN300 and sets the tabs. It should only be used, of course, if the terminal being used as an initializer terminal is a TermiNet 300. Typing on a TermiNet initializer terminal without waiting for the OPER: message sometimes causes the tabs to be cleared, requiring use of this command to reset them.

Name: scav

SYNTAX AS A COMMAND

```
x scav {scavenger_arguments}
```

FUNCTION

starts a physical volume scavenge.

ARGUMENTS

scavenger_arguments
are arguments acceptable to the scavenge_vol command.

EXAMPLES

To log in a daemon to scavenge the root physical volume (rpv) during a period of heavy load:

```
x scav rpv -nopt
```

To log in a daemon to scavenge all physical volumes of the mounted logical volume Public during a period of heavy load:

```
x scav -lv Public -nopt
```

To log in a daemon to scavenge all mounted volumes with volume inconsistencies during a period of light load:

```
x scav -all -auto
```

Name: set_fdump_number, set_fdump, sfdn

SYNTAX AS A COMMAND

```
x sfdn number
```

FUNCTION

sets the next available dump number.

ARGUMENTS

number
is the number to be assigned to the next dump.

EXAMPLES

```
x sfdn 56
```

sets the next dump number to 56.

Name: set_phcs_access, spa

SYNTAX AS A COMMAND

```
x spa Person_id.Project_id
```

FUNCTION

gives a user re access to the phcs_ gate. This command should only be used at the direction of the system administrator.

ARGUMENTS

Person_id.Project_id
is the User_id of the user to be given phcs_ access.

EXAMPLES

```
x spa Greenberg.Multics
```

gives Greenberg.Multics re access to phcs_.

Name: unattended, unattend

SYNTAX AS A COMMAND

```
x unattend
```

FUNCTION

sets up the system for unattended service. It deletes all the tape drives, turns on the unattended and auto-reboot flags in the BCE flagbox, and sets the greeting message to "Unattended service".

Name: wakeup_dump

SYNTAX AS A COMMAND

```
x wakeup_dump source_id
```

FUNCTION

invokes the specified volume or hierarchy incremental dumper daemon.

ARGUMENTS

source_id
: is the source_id of the Backup.SysDaemon (usually "bk") or the Volume_Dumper.Daemon (usually vinc).

SECTION 6

I/O DAEMON COMMANDS

The commands in this section can be used only when communicating with the I/O daemon (i.e., the I/O daemon coordinator and drivers). The I/O daemon is logged in either automatically at system startup time (via the system_startup.ec) or by the operator using the daemon user_id and the coordinator or driver source_id.

At most sites, the I/O daemon is logged in by means of the initializer terminal. Since the initializer terminal can control multiple processes (via the message coordinator), this method avoids the need for a terminal to be dedicated to the I/O daemon. Alternatively, you can log in the I/O daemon from any remote terminal. However, in this case, that terminal is dedicated to communication with the I/O daemon process and cannot be used for other purposes.

The I/O daemon is composed of a "coordinator" and one or more "drivers." The *Operator's Guide to Multics*, Order No. GB61 describes the most common procedures for communicating with the coordinator and the driver(s). This section contains a description of both coordinator commands and driver commands. There are only eight coordinator commands, as follows:

```
logout
list
print_devices
wait_status
term
restart_status
help
start
```

The coordinator commands are identified as such in the individual command descriptions. All other commands are driver commands. You should note that in the case of logout and help there is a coordinator command and a driver command of the same name.

(See Section 1 for a description of how to submit an I/O daemon command when in the initializer command environment.)

Name: auto_queue

SYNTAX AS A COMMAND

```
auto_queue <switch_value>
```

FUNCTION

controls whether output files received by this driver are (1) automatically printed or punched locally or (2) scanned for Multics control records and placed in system pool storage for online perusal.

ARGUMENTS

switch_value
must be chosen from:

yes
automatically queue the files for printing/punching; do not scan for control records, or

no
scan the output files for Multics control records and store them in system pool storage for online perusal; do not automatically queue files for printing/punching.

Name: auto_start_delay

SYNTAX AS A COMMAND

```
auto_start_delay {N}
```

FUNCTION

displays or sets the length of time the driver will wait to issue the start command automatically after receiving a quit signal. An automatic start is cancelled if command input is received.

ARGUMENTS

N
is the desired delay time in seconds. N must be at least 30 seconds. The default delay time is 60 seconds. When no argument is given, the current delay time is displayed.

Name: banner_bars

SYNTAX AS A COMMAND

```
banner_bars {minor_device} {arg}
```

FUNCTION

used by printer drivers to establish how the separator bars at the bottom of the head sheet are to be printed. Printers that can overstrike should use "double" (this is the default). Other printers should use single.

ARGUMENTS

minor_device
: is a minor device name (as shown by the status command) and is required if there is more than one printer minor device; arg can be one of the following separator types:

double
: overstrikes each separator line.

single
: single strikes each separator line.

none
: suppresses separator lines.

-print
: if arg is not given, or if a single arg "-print", is given, the current value is printed.

Name: banner_type

SYNTAX AS A COMMAND

```
banner_type {minor_device} {key}
```

FUNCTION

used by printer drivers to change the information printed on the front and back of each copy of a request.

ARGUMENTS

minor_device
is a minor device name (as shown by the status command) and is required if there is more than one printer minor device; key must be one of the following:

standard
prints the normal head and tail sheets

none
prints nothing except the separator bars, if required (according to the banner_bars command)

brief
prints a short version of the head and tail sheets

-print
if arg is not given, or if a single arg "-print", is given, the current value is printed.

Name: cancel

SYNTAX AS A COMMAND

cancel

FUNCTION

terminates the request that the driver is currently processing. The request is not placed in the coordinator's saved list and thus cannot be restarted later. This command is only valid after a quit signal, or at request command level.

NOTES

After completing the command, the driver looks for another request to process. (In step mode, it returns to command level.)

Name: clean_pool

SYNTAX AS A COMMAND

```
clean_pool N
```

FUNCTION

applies to drivers that can read user card decks. It allows the operator to delete all segments in the system card pool that have been there more than a specified number of days. This command is available for the master terminal only.

ARGUMENTS

N
: is the maximum length of time in days for segments to be retained in the system card pool. All segments that have been in the card pool more than that number of days are deleted. N must be a decimal number greater than zero.

Name: copy

SYNTAX AS A COMMAND

```
copy N
```

FUNCTION

allows the operator to set the copy number of the next copy of the current request to the value specified. This command is used only at request command level.

ARGUMENTS

N
: is a decimal integer between 1 and the number of copies requested by the user.

Name: ctl_term

SYNTAX AS A COMMAND

```
ctl_term arg
```

FUNCTION

applies only to a control terminal (if attached). It allows the operator to specify the format of printed output.

One of the primary functions of the control terminal is to print information about each request processed, to aid in separating the output, and to ensure proper accountability of output generated by the driver. It is possible for the site to use preprinted forms for this purpose. (for information on preprinted forms, see the *Multics System Maintenance Procedures* manual, Order No. AM81.) In this case, alignment of the data on the form is very important. Generally a terminal that supports vertical tab and form feed control characters is used to ensure alignment. However, this command allows the operator to request that the software simulate the action of form feed control characters if the terminal does not provide this hardware support.

ARGUMENTS

where arg falls into one of two classes: general control or simulation control (some arguments require an additional value to define the requested action):

general control

form_type STR
: specifies the format program to be used to format the data printed on the control terminal. If STR is "default", the form_type is set to the default format.

detach
: discontinues the use of the control terminal. This argument is restricted to the master terminal and is not reversible unless the reinit command is given.

simulation control

simulate
: sets the driver to simulate form feeds by software. (This argument is not reversible even by the reinit command.)

page_length N
: sets the number of lines per logical page to N. This controls the forward spacing needed to go to the top of the form.

aligned
: indicates that the forms are aligned for the purpose of form feed control. (A sample form can be printed by the sample_form command.)

Name: defer

SYNTAX AS A COMMAND

```
defer
```

FUNCTION

sends the current request back to its queue marked as deferred. It is only issued from quit command level or request command level.

NOTES

Requests are automatically deferred when the requested line length of the device exceeds the physical line length, or when the estimated processing time of a request exceeds the operator-defined limit (see the defer_time command below).

A deferred request will be reprocessed when the driver is given the restart_q command or when the coordinator is next initialized.

Name: defer_time

SYNTAX AS A COMMAND

```
defer_time {minor_device} {N}
```

FUNCTION

sets or displays the current time limit for automatically deferring requests.

ARGUMENTS

minor_device
: is the name of the minor device for which the time should be set or displayed. It is optional for drivers that have only one minor device (e.g., the central site printer). If specified, this argument must be the first argument.

N
: sets a new defer time in minutes, with a precision of tenths (e.g., 1.5 is one minute, 30 seconds). A time of zero indicates that infinite time is allowed. If N is not given, the current defer time and driver output rate are displayed.

Name: go

SYNTAX AS A COMMAND

```
go {N}
```

FUNCTION

specifies that the driver is to search for requests to process. If no requests are currently available, the driver asks the coordinator for a request for each "ready" device. These requests are processed as soon as they are provided by the coordinator. (This command may not be used at request command level or immediately following a quit signal.)

ARGUMENTS

N
: is the number of requests processed before the driver returns to command level. If N is not specified, the driver will continue to process requests and will not return to command level until requested by the operator.

Name: halt

SYNTAX AS A COMMAND

```
halt devl ... devn {control_arg}
```

FUNCTION

provides the reverse function of the ready command. It places the device or each of the specified minor devices in the inactive state. The driver does not ask the coordinator for any further requests for a halted device. However, the coordinator may have already supplied a "pending request" for the halted device. In this case, any pending request is processed immediately after the device has been halted (except when the command has been issued following a quit signal).

ARGUMENTS

devi
: is the name of a device, or minor device in the case of a multifunction device, that is to be placed in the inactive state. The device names that can be used are those printed out by the status command.

CONTROL ARGUMENTS

-all, -a
specifies that all devices are to be halted. If the -all control argument is used, no device names need be specified. If there is only one device neither the device name nor -all need be specified.

Name: hasp_host_operators_console, hhoc

SYNTAX AS A COMMAND

```
hhoc tty_channel {control_arguments} {attach_arguments}
```

FUNCTION

is used to simulate the operation of the operator's console of a HASP workstation. The operator's console is used to identify a workstation to a remote system, to issue commands governing the operation of the workstation, and to receive status information from the remote system.

ARGUMENTS

tty_channel
is the name of the terminal channel to be attached as the operator's console. This channel must be configured as the console sub-channel of a HASP multiplexer channel (eg: a.h014.op). See the *Multics Communications Administrator's Manual*, Order No. CC75, for a further description of the HASP multiplexer.

CONTROL ARGUMENTS

-signon STR
specifies that the remote host requires a SIGNON record to be transmitted before data transmission may occur. STR is the text of the control record; it may be up to 80 characters in length. Before transmission it is translated to uppercase and the remote system's character is set.

-no_signon
specifies that the remote host does not require a SIGNON record. (Default)

attach_arguments
are options acceptable to the hasp_host_ I/O module. This command supplies the -comm, -tty, and -device options automatically; these options need not be given on the command line. (See the *Multics Subroutines and I/O Modules* manual, Order No. AG93 for a description of the hasp_host_ I/O module.)

NOTES

If the remote system requires a SIGNON, the -signon option should be supplied on the command line specifying the exact SIGNON record to be transmitted.

EXAMPLES

For example, the command line:

```
hhoc a.h014.opr -signon "/*SIGNON REMOTE7"
```

may be used to attach the channel a.h014.opr as the operator's console of a remote IBM system expecting a connection from the workstation named REMOTE7.

After attaching the channel specified on the command line, hasp_host_operators_console prompts the user for terminal input with the string "Input:".

Input from the terminal is transmitted directly to the remote system unless the line begins with the request character, an exclamation mark ( ); lines beginning with the request character are interpreted by this command. The valid requests are described below.

Any text received from the remote system is displayed directly on the terminal without any interpretation by hasp_host_operators_console.

HASP_HOST_OPERATORS_CONSOLE REQUESTS

The following requests are recognized by hasp_host_operators_console when given at the beginning of a line of terminal input:

.. <REST_OF_LINE>
the rest of the line is passed to the Multics command processor for execution as ordinary commands.

prints a message of the form:

```
hasp_host_operators_console N.N; connected to channel NAME.
```

where N.N is the current version of this program and NAME identifies the channel connected as a console to the remote system.

quit
causes the command to hangup the operator's console channel and return to Multics command level.

Name: help (coordinator command)

SYNTAX AS A COMMAND

```
help
```

FUNCTION

lists commands acceptable to the I/O daemon coordinator.

Name: help

SYNTAX AS A COMMAND

```
help
```

FUNCTION

prints the name of each command that may be executed by the driver. A short description of the arguments is provided with each command name. At request command level, the list of commands is limited to those unique to that command level.

Name: hold

SYNTAX AS A COMMAND

```
hold
```

FUNCTION

is used to hold the driver at command level.

NOTES

When the hold command is issued from the master terminal, the slave terminal is unable to issue any command that would cause the driver to leave command level until the master terminal has issued a go command (or a start command following a quit signal). This command should always be used following a quit signal if the automatic start is to be canceled.

Name: inactive_limit

SYNTAX AS A COMMAND

```
inactive_limit {N}
```

FUNCTION

allows the I/O Daemon to log out automatically after a specified period of inactivity.

ARGUMENTS

N
: is the number of minutes of inactivity allowed. N may be from zero to 200 minutes. Zero indicates no automatic logout; this is the default. The current inactivity limit is displayed if N is not given.

NOTES

The inactivity time counter is reset when a request or command is received or a quit is signalled. as well as when the driver processes a new request. A driver at command level is considered active.

An inactivity logout will reinitialize a remote driver so that another station can log in and use the line.

Name: kill

SYNTAX AS A COMMAND

```
kill
```

FUNCTION

terminates the request that the driver is currently processing. The request is passed back to the coordinator and placed in the saved list where it may be restarted if desired (within the limits of the coordinator save time).

After completing the command, the driver looks for another request to process. (In step mode, it returns to command level.)

Name: list (coordinator command)

SYNTAX AS A COMMAND

```
list
```

FUNCTION

causes the coordinator to print a list of active devices, i.e., devices currently assigned to drivers. The request type and current request number are printed for each active device.

Name: listen

SYNTAX AS A COMMAND

```
listen
```

FUNCTION

waits for a remote station to dial the line.

Name: logout (coordinator command)

SYNTAX AS A COMMAND

```
logout
```

FUNCTION

logs out the coordinator. Normally, all driver processes should be logged out before the coordinator. If driver processes are not logged out, however, they automatically detect the fact that the coordinator has been logged out. The drivers reinitialize and wait for a new coordinator to be logged in.

Name: logout

SYNTAX AS A COMMAND

```
logout
```

FUNCTION

terminates the driver process (like the standard Multics logout command).

NOTES

When the logout command is given from a remote station, the remote driver reinitializes and gets ready to accept a new station.

Name: master

SYNTAX AS A COMMAND

```
master mesage
```

FUNCTION

is the reverse of the slave command. It allows the operator of the slave terminal to communicate with the operator of the master terminal by sending a message.

ARGUMENTS

message
is any arbitrary one-line message containing no more than 120 characters.

Name: new_device

SYNTAX AS A COMMAND

```
new_device
```

FUNCTION

terminates the current device. The driver then asks the operator to enter a new "command or device/request_type" as described under "Driver Initialization with a Control Terminal" above.

The coordinator is notified of the termination of the current device and the device is detached by the process. If a control terminal has been attached, it also is detached.

The new_device command may only be issued from the master terminal.

Name: next

SYNTAX AS A COMMAND

```
next -control_args
```

FUNCTION

specifies which request is to be taken from the queues next. This allows the operator to specify priority requests and the order in which they are to be run.

CONTROL ARGUMENTS

-user Person_id.Project_id
specifies the submitter of the request by user_id. The full person and project names must be given. This control argument is required. At least one other argument must be chosen from the request identifiers -entry, -path, and -id.

-entry STR, -et STR
specifies the entryname of the request. Starnames are not allowed. This control argument may not be used with the -path control argument.

-id ID
specifies the match id of the request.

-path path, -pn path
specifies the full pathname of the request. Relative pathnames and starnames are not allowed. This control argument may not be used with the -entry control argument.

-device STR, -dev STR
specifies which of the driver's minor devices the command is being given for. This control argument is optional for drivers with a single minor device, but is required for drivers with multiple minor devices. It serves to identify which request type the coordinator will search to find the request.

-queue N, -q N
specifies that only queue N of the request type should be searched to find a matching request. This argument is optional; if not given, all queues will be searched.

NOTES

All requests to be run by the next command will be charged as though they came from queue 1.

Requests chosen to run next will be run after any restarted requests (see the restart command in this section).

This command may be given several times before a go command, to specify the exact order that requests in the queues are processed.

Name: paper_info

SYNTAX AS A COMMAND

```
paper_info {minor_device} {-control_args}
```

FUNCTION

defines the physical characteristics of the paper as used by the printer software.

ARGUMENTS

minor_device
is a minor device name (as shown by the status command) and is required if there is more than one printer minor device

CONTROL ARGUMENTS

-print
print the current values. If this is given, it must be the only control arg.

-ll N
sets the line length to N, where N is a decimal integer from 10 to 200.

-pl N
sets the page length to N, where N is a decimal integer from 10 to 127.

-lpi N
sets the number of lines per inch to N, where N is either 6 or 8.

If no control arguments are given, the current values are printed.

NOTES

If the printer uses a firmware VFC image, a new image is loaded (which causes the printer to go into an unsynchronized state). Otherwise, the operator is told to mount a new VFU tape.

Name: pause_time

SYNTAX AS A COMMAND

```
pause_time {N}
```

FUNCTION

allows a remote device driver to accept commands between requests by pausing a few seconds to allow the line to turn around.

ARGUMENTS

N
: is the number of seconds that the driver must pause between requests. N must be between 0 and 30 seconds. If N is not given, a value of 10 is assumed.

Name: print_devices (coordinator command)

SYNTAX AS A COMMAND

```
print_devices
```

FUNCTION

causes the coordinator to print a list of all devices managed by the I/O daemon. The devices are grouped according to the request types they service. An asterisk (*) appearing before a device indicates that the associated request type is the default for the device. The driver access name and the driver authorization (if any) are given for each request type.

Name: print_devices

SYNTAX AS A COMMAND

```
print_devices {-control_args}
```

FUNCTION

prints a list of devices for each request type handled by the I/O daemon. Also, the driver access name and driver authorization (if any) for each request type are printed. An asterisk (*) immediately preceding a device name indicates that the associated request type is the default for the device.

CONTROL ARGUMENTS

-access_name STR, -an STR
lists only devices for those request types having a driver access name of STR (STR should be of the form Person_id.Project_id).

-brief, -bf
suppresses printing of a heading line.

-dir path
specifies the absolute pathname of the directory containing the iod_working_tables segment. If not given, the directory >ddd>idd is assumed.

-request_type STR, -rqt STR
lists only devices for the request type specified by STR (e.g., printer, punch).

Name: print_line_ids

SYNTAX AS A COMMAND

```
print_line_ids {-control_args}
```

FUNCTION

prints a list of logical line_ids and their associated communications channel from data in the iod_working_tables segment.

CONTROL ARGUMENTS

-brief, -bf
suppresses printing of a heading line.

-dir path
specifies the absolute pathname of the directory containing the iod_working_tables segment. If not given, the directory >ddd>idd is assumed.

Name: prt_control

SYNTAX AS A COMMAND

```
prt_control {minor_device} {args}
```

FUNCTION

sets the driver request processing modes. Each key may be preceded by the circumflex character (^) to set the value to off.

ARGUMENTS

minor_device
: is a minor device name (as shown by the status command) This argument is optional if there is only one printer minor device, but it required otherwise.

-print
: if arg is not given, or if a single arg "-print" is given, the current modes are printed.

auto_print, ^auto_print
: This mode causes the driver to start printing each request as soon as it is received from the coordinator (after a go command has been given). This is the normal mode of operation. When ^auto_print is turned off, the driver goes to request command level immediately after printing the log message. This allows the operator to align the paper, change the paper, print sample pages, and issue all other commands allowed at request command level (including the kill command).

force_esc, ^force_esc
: This mode turns on the esc mode of the printer DIM during the processing of each request. This mode must be on if the slew-to-channel functions are to operate. Normally, the force_esc mode is set by data in the request type info (rqti) segment.

force_nep, ^force_nep
: This mode sets the noendpage (nep) mode of the printer DIM during the processing of each request, whether the user has requested that mode or not. It is normally set from data in the rqti segment. This mode is used for request types that require preprinted or preformatted paper (e.g., gummed labels, invoice forms).

force_ctl_char
: This sets the ctl_char mode of the printer DIM during the processing of each request, which allows an I/O daemon to send control sequences directly to a remote printer instead of discarding the characters or printing their octal equivalents. Setting this mode enables users who prepare print files through Compose to activate special printer features such as superscripting or multiple fonts. This mode is honored only by the remote printer driver module, remote_driver_.

NOTES

If no arguments are given, the current modes are printed.

Name: punch

SYNTAX AS A COMMAND

```
punch
```

FUNCTION

is used by remote punch drivers at request command level to proceed with the punching of the requested segment.

Name: pun_control

SYNTAX AS A COMMAND

```
pun_control {minor_device} {control_mode} {-control_arg}
```

FUNCTION

is used by remote drivers at normal command level to set the punch control modes. It does not apply to the central site punch driver.

ARGUMENTS

minor_device
: is the name of the punch minor device that the command is addressing. (Required if there are two or more punch minor devices)

control_mode
: specifies the modes to be set. You can precede the mode name by "^" to reset the mode. If you don't give it, the current modes for the specified minor device are printed. This mode is currently defined:

  autopunch
  : allows the driver to process punch requests continuously without operator intervention. When this mode is not set (i.e., ^autopunch), the driver comes to request command level after printing the log message and waits for the operator to give the "punch" command before continuing.

CONTROL ARGUMENTS

-print
if you give it alone, or if you give no control_mode, the modes are printed.

NOTES

The ^autopunch mode is normally used by a remote operator to allow the output to be directed to a particular device based on information in the log message. Once the proper device has been assigned, the operator must type "punch" for the driver to continue with the user's request.

Name: read_cards

SYNTAX AS A COMMAND

```
read_cards
```

FUNCTION

applies to device drivers that can read user card decks. It allows the operator to input card decks from a remote station or local device.

NOTES

The card codes accepted by various card readers may vary from one card reader to another. The operator should be familiar with the card codes used with the card reader at the remote station.

Name: ready

SYNTAX AS A COMMAND

```
ready dev1...devN {-control_arg}
```

FUNCTION

places the device and the specified minor devices in the active, or "ready," state. The driver requests service only from the coordinator for a ready device. This command performs the reverse function of the halt command.

ARGUMENTS

devN
is the name of a device, or minor device in the case of a multifunction device, to be placed in the ready state.

CONTROL ARGUMENTS

-all, -a
makes all devices ready. If you use it, no device names need be given. If there is only one device, -a is not required; in this case, ready is executed automatically during driver initialization. If there are multiple minor devices, the operator must specify either the ones to be made ready or -a.

Name: receive

SYNTAX AS A COMMAND

```
receive
```

FUNCTION

causes the driver to wait for output files to be transmitted from the remote system. A message is issued at the start and end of each file received. If automatic queueing of output files is enabled for this simulated device, output files are locally printed or punched after they have been successfully received; otherwise, the output files are placed into the system pool storage as specified by the ++IDENT control records, which must be present in the files.

Name: reinit

SYNTAX AS A COMMAND

```
reinit
```

FUNCTION

reinitializes the driver. The same device(s) and request type(s) are used without requesting operator input. Remote stations, however, have to reissue the station command and any new default request types. If a control terminal is attached to the driver, its attachment, form simulation mode, and form type are retained over the reinitialization. Each device and request type is again requested from the coordinator.

This command is equivalent to the Multics new_proc command.

Name: release

SYNTAX AS A COMMAND

`release`

FUNCTION

returns the driver to normal command level. It is primarily used following a quit signal. If a request was in progress, it is restarted.

This page intentionally left blank.

Name: req_status

SYNTAX AS A COMMAND

```
req_status {-control_arg}
```

FUNCTION

gives the operator information about the current request. This command may only be used at request command level.

CONTROL ARGUMENTS

-long, -lg
for printers only. Gives the operator the following information:

```
number of multisegment file components
number of characters in file
current page number
current copy number
current line count
current multisegment file component
char offset in current component
char offset from start of file
printer DIM modes
printer DIM position
```

If the control argument is omitted, only the first four items in the above list are printed. In this case, the information looks like:

```
Request 10001: >print_files>invoices>Station_A.invoices
file components:  2,   char count:  4732865
page no:        1006   current copy no: 2
```

There is no control_arg defined for punches. The following three items are printed:

```
current copy number
current request number
current pathname
```

In this case, the information looks like:

```
Request 20001 >punch_files>invoices>Station_A.invoices
current copy no:  2
```

Name: request_type, rqt

SYNTAX AS A COMMAND

```
rqt <rqt_name>
```

FUNCTION

is used to specify the request type to be used for the automatic queuing of output files received by this device.

ARGUMENTS

rqt_name
: is the name of the request type to be used for automatic queuing. The generic type of this request type must agree with the type of device being simulated ("printer" for simulated line printers, etc). This parameter is optional; the default value is the request type specified in the iod_tables definition of this driver.

Name: restart

SYNTAX AS A COMMAND

```
restart {arg}
```

FUNCTION

used either to restart processing of the current request after a device malfunction or to reprocess requests in the coordinator's saved list.

ARGUMENTS

N
: is the number of the request to be restarted. The coordinator searches its saved list for a matching request. If found, the request will be re-processed ahead of any other requests, including those from the "next" command. If the request had been saved in the middle of a copy (suspended), the request will be restarted beginning at the top of the following page; a punch request will start at the beginning of that copy.

-from N
: specifies that all requests in the series beginning with request N are to be restarted. This is an implicit save of all requests in the series.

NOTES

When the restart command is issued directly after a quit signal, with no arguments, the driver's current request is restarted. For print requests, the current page number, minus 5, and copy number are displayed and the driver goes to request command level. For punch requests, the number of copies completed (if more than one) is displayed and the operator is asked to note how many were good.

The user is charged for the requested number of copies only, regardless of how many copies were produced by this command.

If the request number series of a restarted request is still active, the driver will be switched to another series. Each restarted request is assigned a new request number, and any subsequent restart must be based on the new request number.

Name: restart_q

SYNTAX AS A COMMAND

```
restart_q {minor_device}
```

FUNCTION

signals the coordinator to start taking requests from the beginning of the queue again. This allows any deferred requests to be run if the operator has changed the deferring criteria (see the defer_time command in this section).

ARGUMENTS

minor_device
: is the name of one of the minor devices being run by the driver. It identifies the request type queues to be restarted. It is optional for drivers with a single minor device.

NOTES

When several drivers are running from a single device class, and several requests in the queues are still in progress, it is possible that some requests will be repeated.

Name: restart_status (coordinator command)

SYNTAX AS A COMMAND

```
restart_status
```

FUNCTION

causes the coordinator to print the number of restartable requests for each different request series and to dientify those request series for which a restart cycle is in progress.

Name: runout_spacing

SYNTAX AS A COMMAND

```
runout_spacing N
```

FUNCTION

sets the number of lines to advance the paper after requesting a command from a remote multifunction slave terminal.

ARGUMENTS

N
: is the number of lines the driver advances the paper after requesting a command from the slave. N may be from zero to 60.

NOTES

The runout spacing is normally set in the attach description from the iod_tables. This command allows the operator to change the spacing so that driver command requests may be seen clearly above the platen.

Name: sample

SYNTAX AS A COMMAND

```
sample {N}
```

FUNCTION

used by printer drivers at request command level to print a sample page of the file for paper alignment or to verify the starting position in the file. The current position of a new request is always page 1. The same page may be printed as often as needed.

ARGUMENTS

N
: is the page number that the driver prints. If N is omitted, the driver prints the current page in the file.

NOTES

If N is preceded by a "+" or "-", the number is relative to the current page of the file. For example, "sample +3" skips forward three pages and prints the page; "sample -8" skips backward eight pages and prints the page. Similarly, "sample 500" skips to page number 500 and prints it.

If the page number specified is beyond the end of the file, an error message is printed similar to:

```
End-of-File record encountered. EOF at page 2000, line 10.
Unable to skip to starting page.
Enter command(request):

and a new command is requested.
```

The sample command prints a page with separator bars as an aid to the operator in indicating the sample pages so they can be discarded.

Name: sample_form

SYNTAX AS A COMMAND

```
sample_form
```

FUNCTION

prints a sample of the data used to record request processing on the control terminal. The primary function of this command is to verify the alignment of the forms on the control terminal. The data is formatted by the program that is called for each copy of each request. (See the ctl_term command.)

NOTES

If form feed simulation is being used, the command checks to see if alignment has been set. If not, it is set before the sample form is printed.

The sample_form command applies to all drivers that use a control terminal.

Name: sample_hs

SYNTAX AS A COMMAND

```
sample_hs {minor_device}
```

FUNCTION

prints a sample head sheet to align the paper before starting to print or after loading more paper. This command should not be used in the middle of a request (e.g., after a quit) unless the request is restarted using the restart command. Otherwise, the page restart feature of the printer driver is placed out of synchronization.

ARGUMENTS

minor_device
: is a minor device name (as shown by the status command) and is required if there is more than one printer minor device.

Name: save

SYNTAX AS A COMMAND

```
save {arg}
```

FUNCTION

tells the coordinator that one or a series of requests are to be retained beyond the normal holding time. The action is limited to requests in the specified request number series. The save command allows requests to be saved for possible restarting until the coordinator is logged out.

ARGUMENTS

N
specifies the request number in the coordinator's saved list. The coordinator searches its list of finished requests and marks the matching request number as saved for later restarting. The request remains in the saved list until the request is restarted by the restart command or until the coordinator is next initialized.

-from N
specifies that all requests in the series beginning with request N are retained in the saved list.

If no argument is given, the current request will be returned to the coordinator and saved for later restarting. For printers, the request will be processed to the bottom of the next even page and a normal tail sheet will be printed, showing a charge of zero. When the request is later restarted, printing will begin at the top of the next odd page.

NOTES

Once a saved request is restarted, it is not saved any longer than the normal retention time. The coordinator never deletes the user's segment while the request is being saved.

Name: sep_cards

SYNTAX AS A COMMAND

```
sep_cards {minor_device} {arg}
```

FUNCTION

is used by a remote punch driver at normal command level to control the punching of separator cards between each output deck. If separator cards are not punched, the operator should run the driver in step mode (see the step mode command) and remove the cards from the punch as each request is completed.

ARGUMENTS

minor_device
is the name of the punch minor device which is being addressed. This argument is optional if there is only one punch minor device, but is required otherwise.

CONTROL ARGUMENTS

standard
the standard separator cards are to be punched (default).

none
no separator cards are to be punched.

-print

NOTES

if arg is not given, or if a single arg "-print", is given, the current value is printed.

Name: single

SYNTAX AS A COMMAND

```
single
```

FUNCTION

sets the single mode of the printer DIM so that form feed and vertical tab characters are treated as newline characters for the current request. It also cancels any additional requested copies that have not been processed by the driver. The single command is used after a quit to stop runaway paper feeding caused, for example, by the printing of a non-ASCII segment. The command applies only to drivers that operate a printer.

Name: slave

SYNTAX AS A COMMAND

```
slave message
```

FUNCTION

allows the master terminal operator to communicate with the operator of the slave terminal by sending a message. The slave command is the reverse of the master command.

ARGUMENTS

message
is any arbitrary one-line message containing no more than 120 characters.

Name: slave_term

SYNTAX AS A COMMAND

```
slave_term key
```

FUNCTION

controls the ability of the slave terminal to enter commands, issue quit signals, and receive log or error messages. The slave terminal must be active for the command to be effective. The commands, no_commands, quits, and no_quits keys are restricted to the master terminal.

ARGUMENTS

where key may be selected from the following:

commands
: commands can be sent from the slave terminal to the driver. (Restricted to master terminal.)

echo
: echoes each command line typed from the slave. (Input from the exec_com used by the "x" command will not be echoed.)

errors
: error messages are routed to the slave terminal.

log
: log messages are routed to the slave terminal.

modes STRs
: sets the slave terminal modes to those specified by STRs.

no_commands
: no commands can be sent from the slave terminal to the driver. (Restricted to master terminal.)

no_echo
: suppresses echoing of the slave commands (default).

no_errors
: no error messages are routed to the slave terminal.

no_log
: no log messages are routed to the slave terminal.

no_quits
: no quit signals can be sent from the slave terminal to the driver. (Restricted to master terminal.)

quits
quit signals can be sent from the slave terminal to the driver. (Restricted to master terminal.)

Name: start (coordinator command)

SYNTAX AS A COMMAND

```
start
```

FUNCTION

returns the coordinator to normal operation following a quit signal.

Name: start

SYNTAX AS A COMMAND

```
start
```

FUNCTION

allows the driver to resume operations suspended at other than the normal command level, e.g., after a quit signal. Its function is similar to the standard Multics start command. The start command cannot be issued at normal command level (see the go command).

After a quit signal, this is the only command that allows control to be returned to the point of process interruption. The action of the hold command is reset when a start command is issued.

Name: station

SYNTAX AS A COMMAND

```
station station_id {station_password}
```

FUNCTION

used by a driver to identify and validate a remote station. This command is similar to the standard Multics login command.

ARGUMENTS

station_id
is the registered id of the station, as defined by the administrator.

station_password
is the registered password for the remote station.

NOTES

The station's identifier and password are registered in the PNT using the card input password as the station password and are supplied by the administrator for each station location.

If the remote station includes an operator's terminal with keyboard and CRT or printer, the station password may be omitted from the station command. The system will then request the station password and either suppress printing of the password or hide it with a suitable mask. This feature is particularly useful when a remote station is actually a high-quality letter printer (e.g., a Diablo 1640), where the printer is used both as the slave console and as the actual output device.

Remote stations that have no input device do not have to give a station command. However, these stations must use a dedicated phone line and have the station identifier specified in the iod_tables for Type II remote stations.

Name: status

SYNTAX AS A COMMAND

```
status {-control_arg}
```

FUNCTION

prints information about the current status of the driver. The information provided is:

1. The I/O daemon driver version.
2. The device name and channel.
3. The request type (per minor device if more than one).
4. Whether a request is in progress and the request number.
5. The device status: ready, halted, or not attached. (If there are minor devices, this is provided per minor device.)
6. Whether there are any pending requests and their request numbers.
7. Whether step mode is set.
8. The names of any minor devices (to be used with the ready and halt commands).

CONTROL ARGUMENTS

-long, -lg
to print the status of inactive minor devices (devices that cannot be made ready).

Name: step

SYNTAX AS A COMMAND

```
step {arg}
```

FUNCTION

sets (puts the driver into) or resets (takes the driver out of) step mode. When in step mode, the driver returns to command level after processing each request from the coordinator. When not in step mode, the driver processes requests from the coordinator as soon as received without operator interaction. Step mode is useful for checking the alignment of paper on the printer or other device functions prior to allowing the driver to run continuously without operator interaction.

ARGUMENTS

set, reset
to put the driver into or take the driver out of step mode. If no argument is supplied, step mode is set. The driver is not in step mode immediately after driver initialization.

Name: term (coordinator command)

SYNTAX AS A COMMAND

```
term device_name
```

FUNCTION

terminates a driver so that the major device (and all minor devices) assigned to it can be assigned to another driver. The device_name for the driver must be specified following the command. Normally, driver termination is performed automatically when a driver logs out. In the case where a driver process terminates abnormally, the coordinator does not discover that the process is terminated until a new driver attempts to log in; then it is unassigned from the old driver process and is assigned to the new driver process. Therefore, the only time it is necessary to use the term command is when one wishes to terminate an active driver that cannot be logged out. This might be necessary, for example, if the driver is logged in from a remote location. (If the driver process is running, the term command will not cause the driver to detach the channel associated with the major device. It will cause the driver to eventually fault and probably destroy itself.)

ARGUMENTS

device_name
is the name of the driver to be terminated. This argument is required.

Name: wait_status (coordinator command)

SYNTAX AS A COMMAND

```
wait_status
```

FUNCTION

causes the coordinator to print a list of device classes for which requests have been added to the wait list. The number of waiting requests for each of these device classes is also printed. Requests are added to the wait list whenever a driver gives the "next" command, or if the coordinator finds a request for a device class that is not currently active. At sites having only one device class per request type, no requests are automatically added to the wait list. At sites having multiple device classes per request type, requests may be held waiting whenever one or more drivers are active for a rquest type. By examining how many requests are waiting for various device classes, the operator can judge when it is apprporaite to switch a device from one device class to another so that all device classes receive adequate service.

x

Name: x

SYNTAX AS A COMMAND

```
x function {args}
```

FUNCTION

allows drivers to execute an admin exec_com on a site-defined basis.

ARGUMENTS

function
is a site-defined function name.

CONTROL ARGUMENTS

are any arguments needed to implement function.

NOTES

When the user issues the x command, the driver constructs the command line:

```
exec_com >ddd>idd>NAME function {args}
```

where function and args are as above; NAME is either <major_device>_admin.ec for standard drivers or <station_id>_admin.ec for remote drivers. If NAME is not found, the driver will look for the default of iod_admin.ec. Added names can be used to group exec_coms into categories.

An I/O daemon admin exec_com is written by a site administrator to provide site-defined driver x command functions. The use of admin exec_coms is optional, but when missing, the driver x command will not work. See the *Multics System Maintenance Procedures* manual, Order No. AM81, for the application of the admin exec_com to the creation of a driver-to-driver message facility.

Each I/O daemon admin exec_com is located in the >ddd>idd directory and follows standard exec_com rules. There are two types of admin exec_coms: general and device specific. These differ only in segment name, to allow the site to separate x command functions by device name (station_id for remote stations). The iod_admin.ec segment is the general exec_com and will be used by any driver that cannot find a device-specific exec_com. A <device>_admin.ec segment is a device-specific exec_com for the given major device; for example, prta_admin.ec is specific to device prta. Added names can be used to group several devices under a single device-specific exec_com.

The Multics command iod_command may be used within an admin exec_com to execute arbitrary I/O daemon commands. For example:

```
iod_command defer_time 30
```

may be used in an admin exec_com to change the auto defer time limit for the current driver to 30 minutes. The iod_command command is described in Section 2.

When writing an I/O daemon admin exec_com, the administrator must remember that the process that executes it will, most likely, have full SysDaemon access and privileges to the system. Therefore, care must be given in choosing what functions should be placed at the hands of a remote station operator or an inexperienced device operator.

What follows is a sample section of an admin exec_com. It includes examples of how some iod_val active function keys can be used to protect against operator errors. This sample is for illustration only; see the iod_admin.ec segment supplied in the release for working purposes.

X

```
& ------------------------------------------------------------
&
& iod_admin.ec    (to be found in >ddd>idd)
&
& This is the exec_com for the IO Daemon driver "x" command.
& The first argument to the "x" command is &1 in this exec_com.
& The standard action is to transfer control to a label
& which will implement the function of &1.
&
& Any arguments associated with an "x" command function begin
& with &2 in this exec_com.

&command_line off
&goto &1.command

&label help.command
&
& For "x help" print a list of x command functions.
&
&print cdr -user Pers.Proj <seg_ident>
&print car -user Pers.Proj <seg_ident>
&print pq {ldr_args}
&quit

&label cdr.command
&
& For "x cdr -user Pers.Proj <seg_ident>"
$ to cancel a dprint request for this driver
&
&if [not [exists argument &2]]
&then &goto missing_arg.error
cdr -rqt [iod_val request_type] &f2
&quit

&label car.command
&
& For "x car -user Pers.Proj <seg_ident>"
& to cancel an RJE job sent by this station
&
&if [not [exists argument &2]]
&then go to missing_arg.error
car -sender [iod_val station] &f2
&quit

&label pq.command
&
& For "x pq {ldr_args}"
& to list all requests that can be processed by this driver
&
&if [exists argument &2]
&then ldr -a &f2
```

```
&else ldr -a -admin -rqt ([iod_val rqt_string]) -tt
&quit

&label &1.command
&
& This is a catchall for any undefined command functions.
&
&print Undefined driver x command function.
&
ioa_ "received command: ^(^a ^)" &f1
&
&quit

&label missing_arg.error
&
&print Expected argument missing. Try again or type "x help".
&
&quit
```

The Multics command iod_command may be used within an admin exec_com to execute arbitrary I/O daemon commands. For example:

```
iod_command defer_time 30
```

may be used in an admin exec_com to change the auto defer time limit for the current driver to 30 minutes.

SECTION 7

VOLUME BACKUP DAEMON LIMITED SERVICE SUBSYSTEM COMMANDS

The commands in this section are used when communicating with the following daemons: Volume_Dumper.Daemon, Volume_Reloader.Daemon, and Volume_Retriever.Daemon.

The daemons operate within a limited service subsystem. A limited service subsystem is a closed environment in which only preselected commands can be executed. When the daemons are logged in, the start_up.ec causes the daemons to be placed within the limited service subsystem, where the command repertoire is restricted to the set defined in this section.

The necessary command tables (volume_dumper.ct, volume_retriever.ct, and volume_reloader.ct) and the necessary project start_up.ec are supplied by Honeywell with the operating system software. The administrator must copy the vbk_start_up.ec from the tools directory and use it as the project_start_up.ec for the Daemon project. The command for this procedure is illustrated below.

```
copy >t>vbk_project_start_up.ec >udd>Daemon>project_start_up.ec
```

You can then set access on the segment, as appropriate.

The commands available to each of the above-mentioned daemons are listed below:

```
Commands available to Volume_Dumper.Daemon:

complete_volume_dump
consolidated_volume_dump
delete_volume_log
display_pvolog
display_volume_log
dmpr_unlock_pv
end_volume_dump
incremental_volume_dump
merge_volume_log
preattach_dump_volume
purge_volume_log
rebuild_pvolog
recover_volume_log
set_volume_log
set_volume_wakeup_interval
```

```
verify_dump_volume
volume_cross_check
volume_dump_trace_off
volume_dump_trace_on
wakeup_volume_dump
exec_com (see note, below)
help (see note, below)
home_dir (see note, below)
logout (see note, below)
user (see note, below)
system (see note, below)

Commands availabe to Volume_Reloader.Daemon

display_volume_log
merge_volume_log
recover_volume_log
reload_volume
verify_dump_volume
exec_com (see note, below)
help (see note, below)
home_dir (see note, below)
logout (see note, below)
user( see note, below)
system (see note, below)

Commands available to Volume_Retriever.Daemon

list_retrieval_requests
retrieve_from_volume
exec_com (see note, below)
help (see note, below)
home_dir (see note, below)
logout (see note, below)
user (see note, below)
system (see note, below)
```

NOTE: The following are general user commands that are made available in the limited service subsystem for convienience: exec_com, help, home_dir, logout, user, and system. These commands are described in the *Multics Commands and Active Functions* manual, Order No. AG92.

Name: complete_volume_dump

SYNTAX AS A COMMAND

```
complete_volume_dump -control_args
```

FUNCTION

starts a complete volume dump. For each physical volume specified, all segments and directories are dumped.

CONTROL ARGUMENTS

-accounting
specifies that accounting data is to be collected in the directory >sc1>volume_backup_accounts for later processing. The default is off. This control argument is optional.

-auto
specifies that dump volumes are to be selected from the default dump volume pool, which is also known as the tape pool or the volume pool segment, and is located at >ddd>volume_backup>Volume_Dumper.volumes. For more information, see the description of the manage_volume_pool command in the *Multics Commands and Active Functions* manual, Order No. AG92. This control argument is optional.

-control entry
where entry is the entryname of the control segment. The suffix "dump" is assumed if not specified. This control argument is required.

-error_on
specifies that all error messages are to be written to the error_output switch as well as to the dmpr_err.mm/dd/yy.hhmm.s segment. The default is that error messages are written only to the dmpr_err.mm/dd/yy.hhmm.s segment. This control argument is optional.

-incr_skip_count N
specifies that the volume log of each physical volume dumped is to be purged and that N consolidated or complete volume dump volumes must be encountered before incremental dump volumes are purged. This control argument is optional.

-manual_free
specifies that if purging is done, volumes must be freed from the volume pool manually. This means that dump volumes are purged from the respective volume logs but are not reused until they are marked as free in the volume pool. The default is to automatically free volumes in the volume pool. This control argument should be used only with the -auto control argument.

-mod_after DT
where DT is a time acceptable to the convert_date_to_binary_ subroutine. Only those segments and directories whose date-time-modified exceeds DT are dumped. This control argument is optional.

-names
specifies that the entire name space of each directory dumped is recorded online. This online data base significantly improves the performance of the volume retriever, but requires a large amount of disk storage. This control argument is optional.

-no_object
specifies that segments and directories are not to be dumped. Only the header information that describes them is dumped. The default is to dump the segments and directories. This control argument is optional.

-no_trace
specifies that the dumping process is not be be traced. This is the default.

-no_update
specifies that the VTOCE of segments and directories dumped is not to be updated. This speeds up dumping at the cost of invalidating the dtd and volume ID fields of the VTOCE. The default is to update the VTOCE. This control argument is optional.

-operator STR
where STR specifies the name or initials of the user. This control argument is required.

-output_volume_desc STR
where STR specifies a value to be used instead of the default attach description. The dump volume name is inserted in the attach description at the first occurrence of the string "^a" in the attach description. The default attach description is:

```
tape_mult_ ^a -write -system
```

If STR contains blanks, it must be enclosed in quotation marks. This control argument is optional.

-preattach N
specifies that N volumes are to be preattached before writing on the first volume. This allows a site to operate the volume dumper without user intervention, by premounting the requested or supplied volumes. The default is to not preattach. This control argument is optional.

-restart STR
specifies that the dump should restart from physical volume STR, which is implicitly or explicitly defined in the control file. The default is to start at the last volume dumped. This control argument is optional.

-trace
specifies that the dumping process is to be traced. As each segment is dumped to the output tape volume, its unique id and pathname is written to the user_output I/O switch.

-working_dir, -wd
specifies that the volume backup databases are to be searched for relative to the working directory. The default is to search relative to the >ddd>volume_backup directory. This control argument is optional.

NOTES

If a dump terminates abnormally, reinvoke the complete_volume_dump command. The dump will be restarted and reload groups will be kept consistent. If the dump terminates again, use the new_proc command to clean up the dumper process. Then reinvoke the complete_volume_dump command with the -restart argument. The -restart argument must be used after a new_proc to keep reload groups consistent.

Name: consolidated_volume_dump

SYNTAX AS A COMMAND

```
consolidated_volume_dump -control_args
```

FUNCTION

starts a consolidated volume dump. For each physical volume specified, those segments and directories that have been incrementally dumped since the last consolidated volume dump are dumped. This command frees for use the incremental tapes that it makes obsolete, consistent with the -incr_skip_count control argument.

CONTROL ARGUMENTS

-accounting
specifies that accounting data is to be collected in the directory >sc1>volume_backup_accounts for later processing. The default is off. This control argument is optional.

-auto
specifies that dump volumes are to be selected from the default dump volume pool, which is also known as the tape pool or the volume pool segment, and is located at >ddd>volume_backup>Volume_Dumper.volumes. For more information, see the description of the manage_volume_pool command in the *Multics Commands and Active Functions* manual, Order No. AG92. This control argument is optional.

-control entry
where entry is the entryname of the control segment. The suffix "dump" is assumed if not specified. This control argument is required.

-cumulative
specifies that segments and directories dumped are not to be removed from the list of objects to be dumped next time. This provides cumulative dumping if desired. The default is off. This control argument is optional.

-error_on
specifies that all error messages are to be written to the error_output switch as well as to the dmpr_err.mm/dd/yy.hhmm.s segment. The default is that error messages are written only to the dmpr_err.mm/dd/yy.hhmm.s segment. This control argument is optional.

-incr_skip_count N
specifies that the volume log of each physical volume dumped is to be purged and that N consolidated or complete volume dump volumes must be encountered before incremental dump volumes are purged. Purging only takes place for consolidated volume dumps if N is nonzero. This control argument is optional.

-manual_free
specifies that if purging is done, volumes must be freed from the volume pool manually. This means that dump volumes are purged from the respective volume logs but are not reused until they are marked as free in the volume pool. The default is to automatically free volumes in the volume pool. This control argument should be used only with the -auto control argument.

-names
specifies that the entire name space of each directory dumped will be recorded online. This online data base significantly improves the performance of the volume retriever, but requires a large amount of disk storage. This control argument is optional.

-no_object
specifies that segments and directories are not to be dumped. Only the header information that describes them is dumped. The default is to dump the segments and directories. This control argument is optional.

-no_trace
specifies that the dumping process is not be be traced. This is the default.

-no_update
specifies that the VTOCE of segments and directories dumped is not to be updated. This speeds up dumping at the cost of invalidating the dtd and volume id fields of the VTOCE. The default is to update the VTOCE. This control argument is optional.

-operator STR
where STR specifies the name or initials of the user. This control argument is required.

-output_volume_desc STR
where STR specifies a value to be used instead of the default attach description. The dump volume name is inserted in the attach description at the first occurrence of the string "^a" in the attach description. The default attach description is:

```
tape_mult_ ^a -write -system
```

If STR contains blanks, it must be enclosed in quotation marks. This control argument is optional.

-preattach N
specifies that N volumes are to be preattached before writing on the first volume. This allows a site to operate the volume dumper without user intervention, by premounting the requested or supplied volumes. The default is to not preattach. This control argument is optional.

-restart STR
specifies that the dump should restart from physical volume STR, which is implicitly or explicitly defined in the control file. The default is to start at the last volume dumped. This control argument is optional.

-trace
specifies that the dumping process is to be traced. As each segment is dumped to the output tape volume, its unique id and pathname is written to the user_output I/O switch.

-working_dir, -wd
specifies that the volume backup databases are to be searched for relative to the working directory. The default is to search relative to the >ddd>volume_backup directory. This control argument is optional.

NOTES

If a dump terminates abnormally, reinvoke the consolidated_volume_dump command. The dump will be restarted and reload groups will be kept consistent. If the dump terminates again, use the new_proc command to clean up the dumper process. Then reinvoke the consolidated_volume_dump command with the -restart argument. The -restart argument must be used after a new_proc to keep reload groups consistent.

Name: delete_volume_log

SYNTAX AS A COMMAND

```
delete_volume_log vlname {-control_arg}
```

FUNCTION

deletes the volume log segment used by the volume backup subsystem in a way that keeps the other segments used by the volume backup subsystem consistent.

ARGUMENTS

vlname
is the name of the volume log. The volog suffix is assumed if not specified.

CONTROL ARGUMENTS

-manual_free
specifies that if deleting is done, volumes are not freed from the volume pool automatically; manual freeing is required. This means that dump volumes are deleted from the respective volume logs but are not reused until they are marked as free in the volume pool. The default is to automatically free volumes in the volume pool.

NOTES

This command should only be used to delete volume log segments of physical volumes for which neither retrieval nor reloading is intended in the future.

Name: display_pvolog

SYNTAX AS A COMMAND

```
display_pvolog dvname {-control_args}
```

FUNCTION

displays the data content of a physical volume log.

ARGUMENTS

dvname
is the dump volume name.

CONTROL ARGUMENTS

-no_header, -nhe
specifies no header. The default is to have one.

-pvname STR, -pv STR
displays only data associated with the disk volume specified by STR.

-working_dir, -wd
searches for the physical volume log in the directory [wd]>pvolog, rather than in the default system directory.

Name: display_volume_log

SYNTAX AS A COMMAND

```
display_volume_log vlname {-control_args}
```

FUNCTION

displays a summary of the dump volumes that contain information dumped from a specified physical volume.

ARGUMENTS

vlname
is the name of the volume log. The volog suffix is assumed if not specified.

CONTROL ARGUMENTS

-complete, -comp
specifies that information about only complete mode dump volumes is to be displayed.

-consolidated, -cons
specifies that information about only consolidated mode dump volumes is to be displayed.

-incremental, -incr
specifies that information about only incremental mode dump volumes is to be displayed.

-no_header, -nhe
specifies that the output header is not to be printed. The default is to print the output header.

-volname STR
specifies that only the information pertinent to the dump volume named STR is to be printed.

NOTES

If no modes (-incr, -cons, or -comp) are specified, information about all modes is displayed.

Name: dmpr_unlock_pv

SYNTAX AS A COMMAND

```
dmpr_unlock_pv pvname -control_arg
```

FUNCTION

unlocks a physical volume for volume dumping should the dumper process terminate abnormally with the lock set. A control argument must be used to specify the dump mode that is locked.

ARGUMENTS

pvname
is the name of the physical volume that was locked.

CONTROL ARGUMENTS

-complete, -comp
unlocks the complete dump mode.

-consolidated, -cons
unlocks the consolidated dump mode.

-incremental, -incr
unlocks the incremental dump mode.

ACCESS REQUIRED

This command requires hc_backup_ access.

Name: end_volume_dump

SYNTAX AS A COMMAND

```
end_volume_dump
```

FUNCTION

closes open volume logs and detaches attached switches. It is used at the end of volume dump cycles.

Name: incremental_volume_dump

SYNTAX AS A COMMAND

```
incremental_volume_dump -control_args
```

FUNCTION

starts an incremental volume dump. For each physical volume specified, those segments and directories that are stored on the volume are dumped if they have been modified since the last incremental volume dump.

CONTROL ARGUMENTS

-accounting
specifies that accounting data is to be collected in the directory >sc1>volume_backup_accounts for later processing. The default is off. This control argument is optional.

-auto
specifies that dump volumes are to be selected from the default dump volume pool, which is also known as the tape pool or the volume pool segment, and is located at >ddd>volume_backup>Volume_Dumper.volumes. For more information, see the description of the manage_volume_pool command in the *Multics Commands and Active Functions* manual, Order No. AG92. This control argument is optional.

-control entry
where entry is the entryname of the control segment. The suffix "dump" is assumed if not specified. This control argument is required.

-cumulative
specifies that segments and directories dumped are not to be removed from the list of objects to be dumped next time. This provides cumulative dumping if desired. The default is off. This control argument is optional.

-detach
specifies that dump volumes be detached after each pass of the Dumper. The default is -no_detach. This control argument is optional.

-error_on
specifies that all error messages are to be written to the error_output switch as well as to the dmpr_err.mm/dd/yy.hhmm.s segment. The default is that error messages are written only to the dmpr_err.mm/dd/yy.hhmm.s segment. This control argument is optional.

-names
specifies that the entire name space of each directory dumped will be recorded online. This online data base significantly improves the performance of the volume retrieve, but requires a large amount of disk storage. This control argument is optional.

-no_detach
specifies that dump volumes not be detached after each pass of the Dumper, but only when they become full, when they have an error, or at the end of the dump. This is the default.

-no_object
specifies that segments and directories are not to be dumped. Only the header information that describes them is dumped. The default is to dump the segments and directories. This control argument is optional.

-no_trace
specifies that the dumping process is not be be traced. This is the default.

-no_update
specifies that the VTOCE of segments and directories dumped is not to be updated. This speeds up dumping at the cost of invalidating the dtd and volume ID fields of the VTOCE. The default is to update the VTOCE. This control argument is optional.

-operator STR
where STR specifies the name or initials of the user. This control argument is required.

-output_volume_desc STR
where STR specifies a value to be used instead of the default attach description. The dump volume name is inserted in the attach description at the first occurrence of the string "^a" in the attach description. The default attach description is:

```
tape_mult_ ^a -write -system
```

If STR contains blanks, it must be enclosed in quotation marks. This control argument is optional.

-preattach N
specifies that N volumes are to be preattached before writing on the first volume. This allows a site to operate the volume dumper without user intervention, by premounting the requested or supplied volumes. The default is to not preattach. This control argument is optional.

-restart STR
specifies that the dump should restart from physical volume STR, which is implicitly or explicitly defined in the control file. This only occurs in the first pass. This control argument is optional.

-trace
specifies that the dumping process is to be traced. As each segment is dumped to the output tape volume, its unique id and pathname is written to the user_output I/O switch.

-wakeup N
where N is a wakeup interval in minutes for the incremental volume dumper. The interval must be between 1 and 1440 minutes. The default is 60 minutes. This control argument is optional.

-working_dir, -wd
specifies that the volume backup databases are to be searched for relative to the working directory. The default is to search relative to the >ddd>volume_backup directory. This control argument is optional.

NOTES

After a dump cycle has completed, you should invoke the end_volume_command to close any open volume logs and detach any attached switches. The end_volume_dump command can also be used to clean up a dumper process prior to the reinvocation of dumping after a problem has occurred.

Should the alarm clock timer fail, the incremental volume dumper can be restarted by using the wakeup_volume_dump command.

Name: list__retrieval__requests, lrr

SYNTAX AS A COMMAND

```
lrr {path} {-control_args}
```

FUNCTION

lists retrieval requests in the retrieval daemon queues. The request identifier and entryname of each request are printed.

ARGUMENTS

path
is the pathname of a request to be listed. The star convention is allowed. Only requests matching this pathname are selected. If you give no path, all pathnames are selected. This argument is incompatible with -entry.

CONTROL ARGUMENTS

-absolute_pathname, -absp
prints the full pathname of each selected request, rather than the just entryname.

-admin {User_id}, -am {User_id}
selects the requests of all users, or of the user specified by User_id. If you don't choose -admin, only your own requests are selected. This control argument is incompatible with -user. (See "Access Required" below.)

-all, -a
searches all queues and prints the totals for each nonempty queue whether or not any requests are selected from it. This control argument is incompatible with -queue.

-brief, -bf
does not print the state and comment of each request. This control argument is incompatible with -long and -total.

-entry STR, -et STR
selects only requests whose entrynames match STR. The star convention is allowed. Directory portions of request pathnames are ignored when selecting requests.

-id ID
selects only requests whose identifiers match the specified ID.

-long, -lg
prints all the information pertaining to a retrieval request. If you omit -long, only the full pathname of the object or subtree to be retrieved is printed.

-long_id, -lgid
prints the long the request identifier.

-position, -psn
prints the position within its queue of each selected request. When used with -total, it prints a list of all the positions of the selected requests. (See "Access Required.")

-queue N, -q N
searches only queue N. If you don't select -queue, all queues are searched but nothing is printed for queues from which no requests are selected.

-total, -tt
prints only the total number of selected requests and the total number of requests in the queue plus a list of positions if you choose -position. If the queue is empty, it is not listed.

-user User_id
selects only requests entered by the specified user. (See "Access Required.")

ACCESS REQUIRED

You must have o access to the queue(s). You must have r extended access to the queue(s) to use -admin, -position, or -user, since it is necessary to read all requests in the queue(s) to select those entered by a specified user.

NOTES

The default condition is to list only pathnames for the default queue.

The User_id arguments specified after -admin or -user can have any of the following forms:

```
Person_id.Project_id    matches that user only
Person_id.*             matches that person on any project
Person_id               same as Person_id.*
*.Project_id            matches any user on that project
.Project_id             same as *.Project_id
*.*                     same as -admin with no User_id following it.
```

If you select no arguments, only your own requests are selected for listing. (See enter_retrieval_requests.)

Name: merge_volume_log

SYNTAX AS A COMMAND

```
merge_volume_log vlname1 vlname2 {vlname3}
```

FUNCTION

merges the contents of two volume logs that describe the same physical volume. It should only be used to merge the segment produced by a recover_volume_log operation with the volume log already in existence.

ARGUMENTS

vlname1
: is the pathname of the old volume log. After vlname1 and vlname2 are combined, the merged copy is given the name of the old volume log (vlname1), unless the optional argument vlname3 is specified.

vlname2
: is the pathname of the new volume log.

vlname3
: is the pathname of the resultant merged volume log.

NOTES

The suffix volog is assumed for all vlnames, if not specified.

Name: preattach_dump_volumes

SYNTAX AS A COMMAND

```
preattach_dump_volumes N
```

FUNCTION

allows output dump volumes to be attached to a process which is actively performing an incremental volume dump. This command performs the same functions as the -preattach control argument when it is used with the incremental_volume_dump command.

ARGUMENTS

N
: is the number of tape volumes to be preattached. N must be less than or equal to 50.

NOTES

This command may be invoked repeatedly during the life of the incremental volume dump process, but at no time may more than 50 tape volumes be preattached.

Name: purge_volume_log

SYNTAX AS A COMMAND

```
purge_volume_log vlname {-control_args}
```

FUNCTION

removes, in a consistent manner, those entries in a volume log that represent out-of-date information. Each entry in a volume log is part of a reload group. A reload group consists of a set of dump volumes that would be used during a reload of the physical volume.

ARGUMENTS

vlname
is the name of the volume log. The volog suffix is assumed if not specified.

CONTROL ARGUMENTS

-auto
if specified, the volumes purged are freed in the default dump volume pool, which is also known as the tape pool or the volume pool segment, and is located at >ddd>volume_backup>Volume_Dumper.volumes. For more information, see the description of the manage_volume_pool command in the *Multics Commands and Active Functions* manual, Order No. AG92. The default is off.

-force
if specified, causes purging even when some required databases cannot be located.

-incr_skip_count N
specifies that N consolidated or complete dumps must be encountered in a reload group before incremental dump volumes are purged. This allows a site to retain the more recent incremental dump volumes in a log volume even though they have been superseded by later consolidated dump volumes. This control argument is optional. If not specified they are purged.

-test
runs in test mode and reports what would have been done. No databases are affected.

-working_dir -wd
specifies that the volume logs are in the working directory, and the physical volume logs searched for in the [wd]>pvolog directory. The default is to search relative to the >ddd>volume_backup directory.

NOTES

After purging, two reload groups are left in a volume log unless the default number of reload groups has been reset (see the set_volume_log command). A reload group consists of the following: all incremental volume dumps not superseded by consolidated volume dumps; all incremental and consolidated volume dumps not superseded by a complete volume dump; a complete volume dump. The incr_skip_count control argument allows a site to keep incremental dump volumes that contain the "only" copy of some segment for a period of time. An example of such a segment is one that is created, incrementally dumped, then deleted before the consolidated dump is run.

Name: rebuild_pvolog

SYNTAX AS A COMMAND

```
rebuild_pvolog dvname {-control_args}
```

FUNCTION

rebuilds a physical volume log by scanning all the volume logs for instances of the dump volume specified.

ARGUMENTS

dvname
is the name of a dump volume.

CONTROL ARGUMENTS

-brief, -bf
specifies brief output.

-long, -lg
specifies long output.

-working_dir -wd
searches for the volume logs in the working directory; the resulting rebuilt physical volume log is placed in the directory [wd]>pvolog rather than the system default directories.

Name: recover_volume_log

SYNTAX AS A COMMAND

```
recover_volume_log pvnames {-control_args}
```

FUNCTION

recovers volume logs from dump volumes. It should be invoked only if volume logs for physical volumes to be reloaded cannot be found. Its input is a list of the latest dump volumes for the physical volumes in question as specified by the caller.

ARGUMENTS

pvnames
: are the names of the physical volumes whose volume logs are to be recovered. All pvnames supplied must be valid physical volume names (i.e., the physical volumes must be valid members of a registered logical volume).

CONTROL ARGUMENTS

-input_volume_desc STR
: where STR is the attach description used to replace the default attach description "tape_mult_ ^a -system". The dump volume name is inserted in the attach description at the first occurrence of the string "^a" in the attach description.

-working_dir, -wd
: specifies that the volume backup databases are to be recovered relative to the working directory. The default is to recover them relative to the >ddd>volume_backup directory. This control argument can be used to recover the volume logs for physical volumes that are not part of the currently mounted storage system. This control argument is optional.

NOTES

No announcement of the recovery of any volume logs is provided until all volume logs have been recovered or the query for the dump volume name is answered with a period (".").

After a recover_volume_log operation is executed, it may be necessary to run a merge_volume_log operation.

Name: reload_volume

SYNTAX AS A COMMAND

```
reload_volume -control_args
```

FUNCTION

reconstructs the contents of physical volumes using the dump volumes produced by the volume dumper facility. You can use it in ring 1 or 4.

CONTROL ARGUMENTS

-disk_model STR
where STR is the type of disk being reloaded. STR must be one of the following:

3380
3381
d400
d402
d451
d500
d501

If not specified, the registration information for the physical volume to be reloaded is used. This control argument is not allowed when more than one physical volume is being reloaded.

-error_on
specifies that error messages are written to the error_output switch as well as the rldr_err.mm/dd/yy.hhmm.s segment. (Default: off)

-input_volume_desc STR
where STR is an attach description for the dump volumes the reloader reads. The dump volume name is inserted in the specified attach description at the first occurrence of the string "^a" within the attach description. The default attach description is

```
tape_mult_ ^a -system
```

-manual
specifies that the dump volumes are requested by the reloader, rather than being automatically determined. If there are no more dump volumes to read, type a period.

-no_detach
specifies that, at the completion of the reload, neither the dump volume nor the physical volume are detached. This control argument is not allowed when more than one physical volume is being reloaded. (Default: off)

-no_object
specifies that segments and directories are not read from the dump volumes and thus not written to the physical volume. In this case only the VTOC is reloaded. (Default: segments and directories are read)

-operator STR
specifies the name of the user doing the reload. (Required)

-output_volume_desc STR
where STR is an attach description for the physical volume the reloader writes. The physical volume name is inserted at the first occurrence of the string "^a" in the attach description, and the type at the second occurrence. The default attach description is

```
rdisk_ ^a ^a -write -system
```

-pvname STR1 STR2...STRs
specifies the name(s) of the physical volume(s) to be reloaded. You can give it more than once. (Required)

-pvname_device STRP1 STRD1...STRPs STRDs
-pvdv STRP1 STRD1...STRPs STRDs
specifies the name(s) of the physical volume(s) to be reloaded and what device(s) the volume(s) will be on. STRPs and STRDs make up an ordered pair list of pvname (STRPs) followed by the device name (STRDs) that will contain the physical volume (e.g., -pvdv pub01 dska_00a pub02 dska_00b). This control argument is useful when reloading devices that have fixed media, and is the only way to reload a physical volume to a subvolume of a device. You can do this only with the default output attach description. You must set the device usage for "ip" by using the set_drive_usage command. If you use -pvdv, there is no need to use the Multics assign_resource command.

-restart
specifies that the reloader is restarted using control information contained in the control segment in the working directory; the suffix "control" is assumed if not specified. Use -restart only if a system failure occurs during a reload sequence. (Default: off)

-working_dir, -wd
specifies that the volume backup databases are to be searched relative to the working directory. You can use -wd to cause reloading of physical volumes that do not belong to the currently mounted storage system. All specified physical volumes must "belong" to the same RPV. (Default: to search relative to the >ddd>volume_backup directory)

NOTES

When volume-reloading on an MSU 500/501 disk drive, the procedures differ slightly depending on the ring of execution.

When performing the operation in ring 1, the ordering of the pvnames (because of the implementation of RCP) should be the same as that specified for the init_vol, set_drive_usage, and reload_volume commands. Ensure that the only "io" devices configured are those to be reloaded.

When performing the operation in ring 4, you must use the assign_resource command to assign the disk_drive resource on which you want the physical volume to reside after the reload. For this assignment to succeed, the operator must have set both this disk_drive resource and its partner on the shared spindle to "io" with the set_drive_usage command.

*

Name: retrieve_from_volume

SYNTAX AS A COMMAND

```
retrieve_from_volume {-control_args}
```

FUNCTION

processes queued retrieval requests. This command can be operated in an automatic or step mode. It must be run from a process that has access to the retrieval queues located in >daemon_dir_dir>volume_retriever, the volume logs and contents segments produced by the volume dumper, and the ring 0 gates hc_backup_ and system_privilege_.

CONTROL ARGUMENTS

-accounting
specifies that accounting data is to be collected in the directory >sc1>volume_backup_accounts for later processing. The default is off. This control argument is optional.

-all, -a
specifies that all queues are to be processed. This is the default.

-error_on
specifies that all error messages are to be written to the switch error_output as well as the file retv_err.mm/dd/yy.hhmm.s. The default is off.

-input_volume_desc i_att_desc
where i_att_desc is used to replace the default attach description "tape_mult_ ^a -system. The dump volume name is inserted in the attach description at the first occurrence of the string "^a" in the attach description.

-list
lists all requests in the private queue and the options for each request.

-long
logs all requests in the error file.

-manual
specifies that the dump volumes are requested by the volume retriever, rather than being automatically determined. If there are no more dump volumes to read, the operator should type a period (.).

-queue N, -q N
specifies the queue that requests are to be taken from. If not specified, all queues are searched.

-step
specifies that each request in the queue is to be printed on the terminal before processing so that the operator can decide if he wishes to proceed. If not specified, each request is processed until the queue is exhausted. See "Notes" below for more information.

-working_dir, -wd
specifies that the volume backup databases are to be searched for relative to the working directory. The default is to search relative to the >ddd>volume_backup directory. This control argument is optional.

NOTES

If the -step control argument is specified, each retrieval request is printed and the retriever waits for the user to type one of the following commands:

proceed, p
do the retrieval request and delete the request from the queue.

skip, s
skip the retrieval request but leave the request in the queue.

cancel, c
do not process the retrieval request and delete it from the queue.

quit, q
stop processing and return to command level.

help, h
print out the acceptable arguments to the -control control argument.

A list operation only lists the retrieval requests remaining. It does not process them.

Name: set_volume_log

SYNTAX AS A COMMAND

```
set_volume_log vlname {-control_arg}
```

FUNCTION

alters certain parameters that describe the contents of a volume log.

ARGUMENTS

vlname
is the name of the volume log. The volog suffix is assumed if not specified.

CONTROL ARGUMENTS

-reload_groups N
sets the number of reload groups to N, where N>1. The default value, to be set at volume log creation, is 2.

Name: set_volume_wakeup_interval

SYNTAX AS A COMMAND

```
set_volume_wakeup_interval interval
```

FUNCTION

sets a new wakeup interval for a running incremental volume dump process.

ARGUMENTS

interval
is the new wakeup interval in minutes. The interval must be between 1 and 1440 minutes.

NOTES

The new interval becomes effective immediately. If it is less than the current one, the wakeup time may be in the past; in that case, the timer is effective immediately and the dumper is awakened.

Name: verify_dump_volume

SYNTAX AS A COMMAND

```
verify_dump_volume dvname {-control_args}
```

FUNCTION

checks the validity and readability of a dump volume produced by the volume dumper subsystem. Optionally it also produces a pathname list of the segments and directories, if known, on the dump volume.

ARGUMENTS

dvname
is the name of the dump volume.

CONTROL ARGUMENTS

-brief, -bf
produces a short-form list of objects on the dump volume consisting only of pathnames.

-input_volume_desc STR
uses the attach description STR, where the default STR is "tape_mult_ ^a -system". The volname is inserted at the first occurrence of ^a in the attach description.

-long, -lg
produces a long-form list of objects on the dump volume consisting of pathname, object type, and other information where applicable.

-stop_vtocx N
stops the verification process when a VTOCE with octal index N is encountered, calls the debug command, and prints a pointer to the VTOCE.

NOTES

If you give no control argument, the entire dump volume is read and an error count message is printed at the end of the volume. If read errors occur, the same resynchronization mechanism used by the volume reloader and volume retriever is used. If the pathname cannot be determined, the string "UNKNOWN" is used.

Name: volume_cross_check

SYNTAX AS A COMMAND

```
volume_cross_check dvname {-control_args}
```

FUNCTION

checks the validity of the various volume backup databases. Where it detects inconsistent information, it corrects and rebuilds the databases. The cross-checking is performed using the volume pool and the various volume logs.

ARGUMENTS

dvname
is the name of the dump volume.

CONTROL ARGUMENTS

-brief, -bf
specifies brief output.

-long, -lg
specifies long output.

-working_dir, -wd

searches the working directory for volume logs and the Volume_Dumper.volumes volume pool; searches the directory [wd]>contents for the contents segment and the contents names segment; searches the directory [wd]>pvolog for the physical volume log. If you give no -wd, the default system directories are searched.

NOTES

To cross-check all the volumes in use by the volume backup system, issue the following command sequence:

```
manage_volume_pool u >ddd>volume_backup>Volume_Dumper
volume_cross_check ([manage_volume_pool ls -name])
```

Name: volume_dump_trace_off

SYNTAX AS A COMMAND

```
volume_dump_trace_off
```

FUNCTION

turns off dump process tracing in a currently running incremental volume dump process.

Name: volume_dump_trace_on

SYNTAX AS A COMMAND

```
volume_dump_trace_on
```

FUNCTION

turns on dump process tracing. As each segment is dumped to the output tape volume, the command causes the segment's unique ID and pathname to be written to the user_output I/O switch. This command is useful only for incremental volume dumps. You can use it to start tracing of a currently running incremental volume dump process.

wakeup_volume_dump

Name: wakeup_volume_dump

SYNTAX AS A COMMAND

```
wakeup_volume_dump
```

FUNCTION

restarts the incremental volume dumper. It is used when the alarm clock timer fails, and when an additional pass of the dump is needed (e.g., before a shutdown).

SECTION 8

HIERARCHY BACKUP DAEMON LIMITED SERVICE SUBSYSTEM COMMANDS

The commands in this section are used when communicating with the Backup.SysDaemon and the Dumper.SysDaemon. The Backup.SysDaemon is used to produce incremental and consolidated dumps. The Dumper.SysDaemon is used to produce complete dumps.

The Backup.SysDaemon and the Dumper.SysDaemon operate within a limited service subsystem. A limited service subsystem is a closed environment in which only preselected commands can be executed. When the Backup.SysDaemon and the Dumper.SysDaemon are logged in, the start_up.ec causes the daemons to be placed within the limited service subsystem, where the command repertoire is restricted to the set defined in this section.

The necessary command table (hierarchy_dumper.ct) and the necessary project start_up.ec are supplied by Honeywell with the operating system software. The administrator must copy the sysdaemon_project_start_up.ec from the tools directory and use it as the project_start_up.ec for the SysDaemon project. The commands used for this procedure are illustrated below.

```
copy >t>sysdaemon_project_start_up.ec >udd>sd>project_start_up.ec
sa >udd>sd>project_start_up.ec r* -replace
```

Please note that a limited service subsystem is not defined for the Retriever.SysDaemon or the Reloader.SysDaemon. Thus, access to these daemons must be limited to trusted individuals.

The commands available in the limited service subsystem are listed below.

```
catchup_dump
complete_dump
end_dump
start_dump
wakeup_dump
exec_com (see note, below)
help (see note, below)
home_dir (see note, below)
logout (see note, below)
system (see note, below)
user (see note, below)
```

Note: The following are general user commands that are made available in the limited service subsystem for convenience: exec_com, help, home_dir, logout, system, and user. These commands are described in the *Multics Commands and Active Functions* manual, Order No. AG92.

Name: catchup_dump

SYNTAX AS A COMMAND

```
catchup_dump {ctl_file {operator {ntapes {int}}}} {-control_args}
```

FUNCTION

begins a catchup hierarchy dump. It dumps all storage system segments and directories which have been modified since a specified date and time (the default is midnight two days prior). The catchup_dump command calls the backup_dump command to do the actual dumping.

The catchup_dump command requires a dump control file, containing the full pathnames of the objects to be dumped. It calls backup_dump once for each line in the control file. See "Notes on Format of a Dump Control File" below. When the control file is exhausted, i.e., when the catchup dump pass is finished, the catchup_dump command starts up the incremental hierarchy dumper in the same manner as the start_dump command. (Incremental and catchup hierarchy dumps are normally performed by the same process.) You can prevent this by using the -nocontin control argument. A map of all objects dumped is created in the working directory, with the name <unique-name>.dump.map.

If you don't specify the -debug control argument, catchup_dump increases the scheduling priority of the dumping process by calling hphcs_$set_timax with a value of 7000000.

The catchup_dump command is one of the commands used to control hierarchy dumping of storage system segments and directories to magnetic tape. The other commands are:

```
backup_dump
complete_dump
end_dump
start_dump
wakeup_dump
```

You should note that argument processing for all of the hierarchy backup commands is performed by a common argument processing procedure. The values of all arguments are remembered in static storage and remain in effect for the life of the process, unless changed by arguments given in subsequent invocations of backup commands. It should also be noted that the dumping commands and the reloading/retrieving commands are all part of the same hierarchy backup system, and argument values set by the dumping commands remain in effect for the reloading/retrieving commands and vice versa, unless overridden. However, dumping and reloading cannot be done in the same process; use the new_proc command between dumping and reloading. See "Notes on Default Arguments" below.

You may define certain frequently used arguments to the catchup_dump command in two different ways. The first way is to precede the argument with an identifying control argument. The second way is to place the argument in a certain position on the command line. In the first case, the combination of control argument and argument may appear anywhere on the command line after any positional information. In the second case, the argument must appear at the beginning of the command line, before any control argument-argument combinations or any control arguments which stand alone.

For the catchup_dump command, the frequently used arguments are the control file, the operator's name, the number of tape copies, and the wakeup interval. When defined to the command positionally, i.e., without identifying control arguments, these arguments must appear in the following positions:

```
catchup_dump {ctl_file {operator {ntapes {int}}}} {-control_args}
```

The control file and the operator's name must be specified, in one way or the other.

ARGUMENTS

ctl_file
is the pathname of the dump control file (see the -control control argument).

operator
is the operator's name or initials (see the -operator control argument).

ntapes
is the number of copies to be made of each hierarchy dump tape (see the -tapes control argument).

int
is the wakeup interval (see the -wakeup control argument).

CONTROL ARGUMENTS

-all
causes all directory entries (in the specified subtree) to be dumped, regardless of their dates modified or dates dumped. This argument overrides a previously given -dtd control argument or DATE argument. This is the default.

-brief_map, -bfmap
creates a map file that lists the processed entries.

-contin
causes catchup_dump to start the incremental hierarchy dumper after the catchup pass is finished. This is the default.

-control path
indicates that path is the pathname of a dump control file. The suffix "dump" is assumed. For example, "-control sys_dirs" specifies a control file named sys_dirs.dump, in the working directory. See "Notes on Format of a Dump Control File" below.

-debug
disables those phcs_ and hphcs_ calls that deactivate dumped segments and set quotas.

-destination STR, -ds STR
specifies a destination for printing maps and error files. The default is "incremental" for maps and "error file" for error files. *

-dtd
tests and dumps each segment only if the segment or its branch has been modified since the last time it was dumped.

-error_off
writes error messages into a file rather than online. The name of the error file is printed when the first error is encountered. This is the default.

-error_on
writes error messages on the user's terminal.

-header STR, -he STR
specifies a heading for dprinting maps and error files.

-hold
leaves the current hierarchy dump tape or tapes mounted and inhibits rewinding after the current hierarchy dump cycle is completed. *

-map
writes a list of the segments and directories processed into a file. This is the default.

-nocontin
prevents catchup_dump from starting the incremental hierarchy dumper after the catchup pass is finished.

-nodebug
enables hphcs_ calls to set quotas and the transparency switches. This is the default. *

-nohold
rewinds and unloads the hierarchy dump tape or tapes at the end of the current dump pass. This is the default.

-nomap
inhibits listing of the names of processed segments and directories and turns the tape switch on (see -tape below).

-nooutput
inhibits writing hierarchy dump information even if the tape switch is on. This is used for a test run or debugging.

-noprimary, -npri
uses each pathname as given. The default is -primary.

-notape
inhibits writing of a hierarchy tape. This argument also causes a map to be created even if it was previously inhibited. (See -map above.)

-only
indicates that only the requested segment or directory and its branch are to be dumped. This is the opposite of -sweep.

-operator STR
indicates that STR is the user's name or initials (up to 16 characters in length).

-output
writes hierarchy dump information onto the tape if the tape switch is on. This is the default.

-primary, -pr
replaces all directory names in each pathname with the primary names. This is the default.

-pvname STR
indicates that segments and directories may only be dumped if they reside on the physical volume specified by STR.

-request_type STR, -rqt STR
specifies an output request type for printing maps and error files. Available request types can be listed by using the print_request_types command (described in the *Multics Commands and Active Functions* manual, Order No. AG92). The default is "printer."

-restart path
indicates that path is the pathname of a segment or directory where hierarchy dumping is to be restarted. Use of this feature assumes that there is a dump control file. It is normally used to restart a hierarchy dump that was interrupted by a system failure.

-sweep
indicates that the whole subtree beginning with the given directory is to be dumped, subject to the criteria of the -dtd control argument or the DATE argument if either has been invoked. This is the default.

-tape
allows writing of a tape. This is the default.

-tapes N
indicates that N is the number of output tape copies to be made where N can be either 1 or 2. The default is 1.

-1tape
sets the number of tape copies to 1 as an alternative to the -tapes argument.

-2tapes
sets the number of tape copies to 2 as an alternative to the -tapes argument.

-wakeup N
indicates that N is the wakeup interval between incremental hierarchy dump passes, in minutes. The default is 60 minutes.

DATE
an argument beginning with a character other than "-" or ">" is assumed to be a date, in a format acceptable to the convert_date_to_binary_ subroutine. If the argument can be converted to a date, then only segments and directories modified after that date are dumped.

NOTES ON DEFAULT ARGUMENTS

The values of arguments given to any of the hierarchy backup commands are remembered in static storage and remain in effect for the life of the process, unless explicitly changed during the invocation of a subsequent backup command.

The following defaults are in effect for the dumper before any backup commands are given; they are not, however, reset to these values at the start of each backup command, except as noted.

```
-all
-contin
-error_of
-map
-nodebug
-nohold
-output
-primary
-sweep
-tape
```

The following defaults are set automatically at the time the respective commands are executed:

```
catchup_dump
     -tape
      (default date yesterday at midnight)

complete_dump
     -all
     -tape

start_dump
     -dtd
     -hold
     -tape
     -wakeup 60
```

NOTES ON FORMAT OF A DUMP CONTROL FILE

The control file specified by "-control path" is an ASCII segment containing absolute pathnames of entries (segments, MSFs, and directory subtrees) to be dumped, each on a separate line.

Name: complete_dump

SYNTAX AS A COMMAND

```
complete_dump {ctl_file {operator {ntapes}}} {-control_args}
```

FUNCTION

begins a complete hierarchy dump. It dumps all storage system segments and directories, regardless of their modification dates. The complete_dump command calls the backup_dump command to do the actual dumping.

The complete_dump command requires a dump control file, containing the full pathnames of the objects to be dumped. It calls backup_dump once for each line in the control file. See "Notes on Format of a Dump Control File" below. A map of all objects dumped is created in the working directory, with the name <unique-name>.dump.map.

The complete_dump command is one of the commands used to control hierarchy dumping of storage system segments and directories to magnetic tape. The other commands are:

```
backup_dump
catchup_dump
end_dump
start_dump
wakeup_dump
```

You should note that argument processing for all of the hierarchy backup commands is performed by a common argument processing procedure. The values of all arguments are remembered in static storage and remain in effect for the life of the process, unless changed by arguments given in subsequent invocations of backup commands. It should also be noted that the dumping commands and the reloading/retrieving commands are all part of the same hierarchy backup system, and argument values set by the dumping commands remain in effect for the reloading/retrieving commands and vice versa, unless overridden. However, dumping and reloading cannot be done in the same process; use the new_proc command between dumping and reloading. See "Notes on Default Arguments" below.

You may define certain frequently used arguments to the complete_dump command in two different ways. The first way is to precede the argument with an identifying control argument. The second way is to place the argument in a certain position on the command line. In the first case, the combination of control argument and argument may appear anywhere on the command line after any positional information. In the second case, the argument must appear at the beginning of the command line, before any control argument-argument combinations or any control arguments which stand alone.

For the complete_dump command, the frequently used arguments are the control file, the operator's name, and the number of tape copies. When defined to the command positionally, i.e., without identifying control arguments, these arguments must appear in the following positions:

```
complete_dump {ctl_file {operator {ntapes}}} {-control_args}
```

The control file and the operator's name must be specified, in one way or the other.

ARGUMENTS

ctl_file
: is the pathname of the dump control file (see the -control control argument).

operator
: is the operator's name or initials (see the -operator control argument).

ntapes
: is the number of copies to be made of each hierarchy dump tape (see the -tapes control argument).

CONTROL ARGUMENTS

-all
causes all directory entries (in the specified subtree) to be dumped, regardless of their dates modified or dates dumped. This argument overrides a previously given -dtd control argument or DATE argument. This is the default.

-brief_map, -bfmap
creates a map file that lists the processed entries.

-control path
indicates that path is the pathname of a dump control file. The suffix "dump" is assumed. For example, "-control sys_dirs" specifies a control file named sys_dirs.dump, in the working directory. See "Notes on Format of a Dump Control File" below.

-debug
disables those phcs_ and hphcs_ calls that deactivate dumped segments and set quotas.

-destination STR, -ds STR
specifies a destination for printing maps and error files. The default is "incremental" for maps and "error file" for error files.

-dtd
tests and dumps each segment only if the segment or its branch has been modified since the last time it was dumped.

-error_of
writes error messages into a file rather than online. The name of the error file is printed when the first error is encountered. This is the default.

-error_on
writes error messages on the user's terminal.

-header STR, -he STR
specifies a heading for dprinting maps and error files.

-hold
leaves the current hierarchy dump tape or tapes mounted and inhibits rewinding after the current hierarchy dump cycle is completed.

-map
writes a list of the segments and directories processed into a file. This is the default.

-nodebug
enables hphcs_ calls to set quotas and the transparency switches. This is the default.

-nodprint, -no_dprint, -ndp
causes the start_dump command not to dprint maps and error files. The default is -dprint. This control argument cannot be used with the backup_dump command.

-nohold
rewinds and unloads the hierarchy dump tape or tapes at the end of the current dump pass. This is the default.

-nomap
inhibits listing of the names of processed segments and directories and turns the tape switch on (see -tape below).

-nooutput
inhibits writing hierarchy dump information even if the tape switch is on. This is used for a test run or debugging.

-noprimary, -npri
uses each pathname as given. The default is -primary.

-notape
inhibits writing of a hierarchy tape. This argument also causes a map to be created even if it was previously inhibited. (See -map above.)

-only
indicates that only the requested segment or directory and its branch are to be dumped. This is the opposite of -sweep.

-operator STR
indicates that STR is the user's name or initials (up to 16 characters in length).

-output
writes hierarchy dump information onto the tape if the tape switch is on. This is the default.

-primary, -pr
replaces all directory names in each pathname with the primary names. This is the default.

-pvname STR
indicates that segments and directories may only be dumped if they reside on the physical volume specified by STR.

-queue N, -q N
specifies a queue number for any maps and error files that are dprinted. The default is queue 1.

-request_type STR, -rqt STR
specifies an output request type for printing maps and error files. Available request types can be listed by using the print_request_types command (described in the *Multics Commands and Active Functions* manual, Order No. AG92). The default is "printer."

-restart path
indicates that path is the pathname of a segment or directory where hierarchy dumping is to be restarted. Use of this feature assumes that there is a dump control file. It is normally used to restart a hierarchy dump that was interrupted by a system failure.

-sweep
indicates that the whole subtree beginning with the given directory is to be dumped, subject to the criteria of the -dtd control argument or the DATE argument if either has been invoked. This is the default.

-tape
allows writing of a tape. This is the default.

-tapes N
indicates that N is the number of output tape copies to be made where N can be either 1 or 2. The default is 1.

-1tape
sets the number of tape copies to 1 as an alternative to the -tapes argument.

-2tapes
sets the number of tape copies to 2 as an alternative to the -tapes argument.

-wakeup N
indicates that N is the wakeup interval between incremental hierarchy dump passes, in minutes. The default is 60 minutes.

DATE
an argument beginning with a character other than "-" or ">" is assumed to be a date, in a format acceptable to the convert_date_to_binary_ subroutine. If the argument can be converted to a date, then only segments and directories modified after that date are dumped.

NOTES ON DEFAULT ARGUMENTS

The values of arguments given to any of the hierarchy backup commands are remembered in static storage and remain in effect for the life of the process, unless explicitly changed during the invocation of a subsequent backup command.

The following defaults are in effect for the dumper before any backup commands are given; they are not, however, reset to these values at the start of each backup command, except as noted.

```
-all
-contin
-error_of
-map
-nodebug
-nohold
-output
-primary
-sweep
-tape
```

The following defaults are set automatically at the time the respective commands are executed:

```
catchup_dump
     -tape
      (default date yesterday at midnight)

complete_dump
     -all
     -tape

start_dump
     -dtd
     -hold
     -tape
     -wakeup 60
```

NOTES ON FORMAT OF A DUMP CONTROL FILE

The control file specified by "-control path" is an ASCII segment containing absolute pathnames of entries (segments, MSFs, and directory subtrees) to be dumped, each on a separate line.

Name: end_dump

SYNTAX AS A COMMAND

```
end_dump
```

FUNCTION

ends an incremental hierarchy dump. It rewinds and detaches the tape, and dprints the dump map and the error file (if one exists). It also disables the automatic wakeup initiated by the start_dump command. You should use end_dump when you don't want any more incremental dumps; for example, before you shut the system down, and before you run a catchup hierarchy dump (but only if the incremental dumper is running).

The end_dump command is one of the commands used to control hierarchy dumping of storage system segments and directories to magnetic tape. The other commands are:

```
backup_dump
catchup_dump
complete_dump
start_dump
wakeup_dump
```

Name: start_dump

SYNTAX AS A COMMAND

```
start_dump {ctl_file {operator {ntapes {int}}}} {-control_args}
```

FUNCTION

begins incremental hierarchy dumping. It dumps all storage system segments and directories which have been modified since the last time they were dumped. After setting a timer to periodically wakeup and begin an incremental dump pass, the start_dump command calls the backup_dump command to do the actual dumping. (The default wakeup interval is one hour.)

The start_dump command requires a dump control file, containing the full pathnames of the objects to be dumped. It calls backup_dump once for each line in the control file. See "Notes on Format of a Dump Control File" below. When the control file is exhausted, the incremental dumper "goes to sleep" until it is awakened by the timer to begin another pass. A map of all objects dumped is created in the working directory, with the name <unique-name>.dump.map.

The start_dump command is one of the commands used to control hierarchy dumping of storage system segments and directories to magnetic tape. The other commands are:

```
backup_dump
catchup_dump
complete_dump
end_dump
wakeup_dump
```

You should note that argument processing for all of the hierarchy backup commands is performed by a common argument processing procedure. The values of all arguments are remembered in static storage and remain in effect for the life of the process, unless changed by arguments given in subsequent invocations of backup commands. It should also be noted that the dumping commands and the reloading/retrieving commands are all part of the same hierarchy backup system, and argument values set by the dumping commands remain in effect for the reloading/retrieving commands and vice versa, unless overridden. However, dumping and reloading cannot be done in the same process; use the new_proc command between dumping and reloading. See "Notes on Default Arguments" below.

You may define certain frequently used pieces of information to the start_dump command in two different ways. The first way is to precede the peice of information with an identifying control argument. The second way is to place the piece of information in a certain position on the command line. In the first case, the combination of control argument and information may appear anywhere on the command line after any positional information. In the second case, the information must appear at the beginning of the command line, before any control argument-information combinations or any control arguments which stand alone.

For the start_dump command, the frequently used pieces of information are the control file, the operator's name, the number of tape copies, and the wakeup interval. When defined to the command positionally, i.e., without identifying control arguments, these peices of information must appear in the following positions:

```
start_dump {ctl_file {operator {ntapes {int}}}} {-control_args}
```

The control file and the operator's name must be specified, in one way or the other.

ARGUMENTS

ctl_file
: is the pathname of the dump control file (see the -control control argument).

operator
: is the operator's name or initials (see the -operator control argument).

ntapes
: is the number of copies to be made of each hierarchy dump tape (see the -tapes control argument).

int
is the wakeup interval (see the -wakeup control argument).

CONTROL ARGUMENTS

-all
causes all directory entries (in the specified subtree) to be dumped, regardless of their dates modified or dates dumped. This argument overrides a previously given -dtd control argument or DATE argument.

-brief_map, -bfmap
creates a map file that lists the processed entries.

-control path
indicates that path is the pathname of a dump control file. The suffix "dump" is assumed. For example, "-control sys_dirs" specifies a control file named sys_dirs.dump, in the working directory. See "Notes on Format of a Dump Control File" below.

-debug
disables those phcs_ and hphcs_ calls that deactivate dumped segments and set quotas.

-destination STR, -ds STR
specifies a destination for printing maps and error files. The default is "incremental" for maps and "error file" for error files.

-dprint, -dp
causes the start_dump command to dprint maps and error files. This is the default.

-dtd
tests and dumps each segment only if the segment or its branch has been modified since the last time it was dumped. This is the default.

-error_of
writes error messages into a file rather than online. The name of the error file is printed when the first error is encountered. This is the default.

-error_on
writes error messages on the user's terminal.

-header STR, -he STR
specifies a heading for dprinting maps and error files.

-hold
leaves the current hierarchy dump tape or tapes mounted and inhibits rewinding after the current hierarchy dump cycle is completed. This is the default.

-map
writes a list of the segments and directories processed into a file. This is the default. *

-nodebug
enables hphcs_ calls to set quotas and the transparency switches. This is the default.

-nodprint, -no_dprint, -ndp
causes the start_dump command not to dprint maps and error files. The default is -dprint.

-nohold
rewinds and unloads the hierarchy dump tape or tapes at the end of the current dump pass. This is the default for all hierarchy dump modes except incremental.

-nomap
inhibits listing of the names of processed segments and directories and turns the tape switch on (see -tape below).

-nooutput
inhibits writing hierarchy dump information even if the tape switch is on. This is used for a test run or debugging.

-noprimary, -npri
uses each pathname as given. The default is -primary.

-notape
inhibits writing of a hierarchy tape. This argument also causes a map to be created even if it was previously inhibited. (See -map above.)

-only
indicates that only the requested segment or directory and its branch are to be dumped. This is the opposite of -sweep.

-operator STR
indicates that STR is the user's name or initials (up to 16 characters in length).

-output
writes hierarchy dump information onto the tape if the tape switch is on. This is the default.

-primary, -pr
replaces all directory names in each pathname with the primary names. This is the default.

-pvname STR
indicates that STR is the name of a physical volume to be dumped. *

-request_type STR, -rqt STR
specifies an output request type for printing maps and error files. Available request types can be listed by using the print_request_types command (described in the *Multics Commands and Active Functions* manual, Order No. AG92). The
* default is "printer."

-sweep
indicates that the whole subtree beginning with the given directory is to be dumped, subject to the criteria of the -dtd control argument or the DATE argument if either has been invoked. This is the default.

-tape
allows writing of a tape. This is the default.

-tapes N
indicates that N is the number of output tape copies to be made where N can be either 1 or 2. The default is 1.

-1tape
sets the number of tape copies to 1 as an alternative to the -tapes argument.

-2tapes
sets the number of tape copies to 2 as an alternative to the -tapes argument.

-wakeup N
indicates that N is the wakeup interval between incremental hierarchy dump passes, in minutes. The default is 60 minutes.

DATE
an argument beginning with a character other than "-" or ">" is assumed to be a date, in a format acceptable to the convert_date_to_binary_ subroutine. If the argument can be converted to a date, then only segments and directories modified after that date are dumped.

NOTES ON DEFAULT ARGUMENTS

The values of arguments given to any of the hierarchy backup commands are remembered in static storage and remain in effect for the life of the process, unless explicitly changed during the invocation of a subsequent backup command.

The following defaults are in effect for the dumper before any backup commands are given; they are not, however, reset to these values at the start of each backup command, except as noted.

```
-all            -nohold
-contin         -output
-error_of       -primary
-map            -sweep
-nodebug        -tape
```

The following defaults are set automatically at the time the respective commands are executed:

```
catchup_dump
     -tape
      (default date yesterday at midnight)

complete_dump
     -all
     -tape

start_dump
     -dtd
     -hold
     -tape
     -wakeup 60
```

NOTES ON FORMAT OF A DUMP CONTROL FILE

The control file specified by "-control path" is an ASCII segment containing absolute pathnames of entries--segments, multisegment files (MSFs), and directory subtrees--to be dumped, each on a separate line. Cross-dumping is specified by "=new_path" following a pathname, with no intervening spaces, where new_path is the pathname of the new parent directory if the string contains >'s; otherwise, it is a new entryname to replace the entryname portion of the pathname dumped. The entry is placed on the tape as if its pathname were the resulting new pathname.

NOTES ON MANAGING TAPES AND STOPPING A DUMP

The hierarchy dumper manages tape reels by querying you for a tape label:

```
Type tape label:
```

or if two tapes are being written:

```
Type primary tape label:
```

and

```
Type secondary tape label:
```

This input is used to request a tape mount and label verification from the system resource control. If the label cannot be verified (because of a mismatch or tape damage) or validated from the console, or if you deny the tape mount, the dumper requests another label.

Name: wakeup_dump

SYNTAX AS A COMMAND

```
wakeup_dump
```

FUNCTION

is called automatically by the start_dump command at a specified time interval, to begin an incremental hierarchy dump pass. You can also execute it manually, just before the end_dump command, to make one last pass of the incremental dumper before you shut the system down.

This command is one of the commands used to control hierarchy dumping of storage system segments and directories to magnetic tape. The other commands are:

```
backup_dump
catchup_dump
complete_dump
end_dump
start_dump
```

SECTION 9

BCE COMMANDS

The commands in this section can be used only when the system is in the bootload command environment (BCE). The only method of communicating with the system in the bootload command environment is through the bootload console.

Name: alert

SYNTAX AS A COMMAND

```
alert message
```

FUNCTION

writes a message on the bootload console with an audible alarm. This is useful in auto exec_coms to inform the operator that the system has crashed. This command is valid at all BCE command levels.

ARGUMENTS

message
is the message you want to write on the console.

EXAMPLES

```
alert The system has crashed!!!
```

Name: bce

SYNTAX AS A COMMAND

```
bce
```

FUNCTION

causes BCE to finish booting. BCE passes from the "early" command level to the "boot" command level, where it is fully initialized.

Name: bce_state, bces

SYNTAX AS A COMMAND

```
bces
```

SYNTAX AS AN ACTIVE FUNCTION

```
[bces]
```

FUNCTION

prints or returns the name of the BCE state which is currently running (early, boot, bce_crash or crash).

NOTES

This command is useful in exec_coms to diagnose cases where the exec_com is only to run during some states and not others.

Name: boot

SYNTAX AS A COMMAND

```
boot {command} {keywords} {-control_arg}
```

FUNCTION

boots Multics. This command is valid at the "boot" and "bce_crash" command levels.

ARGUMENTS

command
can be one of the following ring 1 command abbreviations:

```
mult        multics
salv        salvage_dirs
stan        standard
star        startup
```

keywords
can be one or more of the following:

nodt
recreates the disk table; renames and ignores the existing one.

nolv
recreates the logical volume registration directory (>lv); renames and ignores the existing one.

nosc
boots the system to the ring 4 emergency listener. See "Notes," below.

rlvs
performs a volume salvage of the RPV (root physical volume), a directory salvage of all directories used in initialization, and a volume salvage of all other member volumes of the RLV (root logical volume).

rpvs
performs a volume salvage of the RPV and a directory salvage of all directories used in initialization.

CONTROL ARGUMENTS

-cold
specifies that the root directory is to be recreated, thus destroying the old file system hierarchy. This option should only be used when a cold boot of BCE was also performed. You will be asked whether BCE should continue.

-time
specifies that the system is to halt before setting the system clock and query the user as to the desired clock setting.

NOTES

The following BCE command line will boot the system to the ring 4 emergency listener:

```
boot stan nosc
```

The emergency listener should be used only in the following circumstances:

1. The system cannot be booted to standard level from ring 1, or it refuses to enter admin mode once it has been booted there.

2. The system libraries are thought to be intact.

If 2 is not the case, then the system will either fail to reach the listener level, or you will be unable to execute many commands.

The system will establish the emergency listener even if the RLV is incomplete. If the RLV is incomplete, some or all commands may be unavailable.

The emergency listener level is a full Multics command level. It is impossible to start the answering service from the emergency listener. After repairing the problem (or deciding that you cannot repair it) you must shut down Multics with

```
hphcs_$shutdown
```

which will return you to BCE.

The emergency listener level is primarily useful for two circumstances:

1. If the logical volume registration for the root logical volume is damaged such that the system claims that the root is incomplete when it is in fact complete, you can use the emergency listener level to repair the registration with the volume registration commands.

2. If the log >sc1>as_logs>admin_log is damaged such that Multics cannot automatically recover it, the emergency listener level is entered automatically to allow you to resolve the problem.

*

Name: config_edit, config

SYNTAX AS A COMMAND

```
config {file_name}
```

FUNCTION

enters the config deck editor. This editor is identical to qedx, except that buffer 0 contains a copy (in ASCII source form) of the config deck. This command is not valid at the "crash" command level.

ARGUMENTS

file_name
is the name of a file to be read into the config deck.

NOTES

If you supply file_name, the specified file is read into the config deck without entering the config deck editor.

If you supply no file_name, the current config deck (that found in the CONF partition on the RPV) is read into buffer 0. It is converted to a labeled ASCII form, which is an expanded form of that used in the configuration card description section. You can do arbitrary text-editing operations on this buffer as well as on any other. Performing a write request on buffer 0 writes the edited buffer back into the config deck.

When used with a file name, you can use the read request to insert a file into the copy of the config deck, or the write request to make a copy of the config deck in a file. With a file name, these two requests do not change the default file name of the config_edit's copy of the config deck; without a file name, they always refer to the config_edit's copy of the config deck.

You can use the qedx request line 1,$dr to restore the original config deck if you have issued no write request; conversely, this request line discards only those changes made since the most recent write.

In the labeled format, you can optionally precede each field on a config card by a label, except for the card name. Labeled fields can appear in any order. The interpretation of a card in labeled form is that all labeled fields are placed into their proper places; any unlabeled fields, then, fill in the missing spaces. Thus,

```
iom -state on -port 1 a imu
```

becomes

```
iom a 1 imu on
```

in its standard format.

The various labeled formats are described in the *Multics System Maintenance Procedures Manual*, Order No. AM81. If a card's format has been locally changed, or if its format or type is otherwise unknown, you can place a "." in front of its name to avoid errors during card parsing. That card cannot have any labeled fields.

See "Config Deck and Device Accessibility" in the *Multics System Maintenance Procedures Manual*, Order No. AM81, for implications on the use of this command.

See the qedx command in the *Multics Commands and Active Functions* manual, Order No. AG92, for information on its requests and their options.

Name: continue, go

SYNTAX AS A COMMAND

go

FUNCTION

restores a saved machine image and continues running an interrupted activity (usually Multics). The machine image is saved when Multics is interrupted after a manual return to BCE or after encountering a BCE probe breakpoint. This command is valid at the "bce_crash" and "crash" command levels.

Name: copy_disk

SYNTAX AS A COMMAND

```
copy_disk source_disk target_disk {-control_args disk_range_specifier}
```

FUNCTION

copies a disk.

ARGUMENTS

source_disk
represents the source disk, in the form "dska_02".

target_disk
represents the target disk, in the form "dska_02".

disk_range_specifier
is any specification of a range of a disk to copy, in a form acceptable to copy_disk. You can specify this range using the control arguments listed below. The default is the entire disk, except for the ALT partition.

CONTROL ARGUMENTS

-first_record N, -frec N
specifies the first record to be read and/or written. You can use the keywords "first" and "last" instead of a record number; they specify the first or last record of the device (or partition if you gave one). (Default: the first record number for the device type unless you give a partition, in which case the default is the first record of the partition)

-force, -fc
suppresses the query "Do you wish to write on target_disk?"

-last_record N, -lrec N
specifies the last record to be read and/or written. You can use the keywords "first" and "last" instead of a record number; they specify the first or last record of the device (or partition if you gave one). (Default: the last record number for the device type unless you give a partition, in which case the default is the last record of the partition)

-n_records N, -nrec
specifies the number of records to be read and/or written. If you give both -frec and -nrec, copy_disk reads and/or writes up to N records, where the first record the command operates on is given by -frec. Correspondingly, if you give both -lrec and -nrec, copy_disk reads and/or writes up to N records, where the last record the command operates on is given by -lrec. (See "Notes.")

-partition part_name, -part part_name
specifies a partition, where part_name is the name of the desired partition. If you supply a partition, the defaults for the first and last page become the first and last page of the partition. (Optional)

-record N, -rec N
specifies a single page to be read and/or written. You can use the keywords "first" and "last" instead of a record number; they specify the first or last record of the device (or partition if you give one). (Optional)

NOTES

You can invoke copy_disk at all bce command levels, except for the "early" level since only the RPV is known at that time. The command does not affect the state of an existing Multics image.

The source and target disks must be of the same type.

You can supply control arguments anywhere within the disk range specification.

If you want to operate on more than one record, use two of the following: -frec, -lrec, and -nrec.

Name: delete, dl

SYNTAX AS A COMMAND

```
dl file_name {...file_names}
```

FUNCTION

deletes files within the BCE file system (not the Multics storage system). It is valid at all BCE command levels.

ARGUMENTS

file_name
is the name of a BCE file to be deleted. You can use the star convention

Name: die

SYNTAX AS A COMMAND

```
die {-control_arg}
```

FUNCTION

aborts all BCE activities. It wipes out the BCE toehold, preventing any returns to BCE, manual or otherwise. It should be used only when you want to absolutely destroy any remnants of BCE. It is valid at all BCE command levels.

CONTROL ARGUMENTS

-force, -fc
: destroys BCE without querying you first. If you don't use it, the command asks you if BCE should really be destroyed before it destroys it.

Name: display_disk_label, ddl

SYNTAX AS A COMMAND

```
ddl device
```

FUNCTION

displays information recorded in the physical volume label for a storage system disk volume.

ARGUMENTS

device
: specifies the disk subsystem, drive, and, if the device is a 3380 or 3381, subvolume on which the physical volume is located (e.g., dska_07 or dskc_00b).

EXAMPLES

The command line

```
ddl dska_16
```

produces

```
Label for Multics Storage System Volume rpv on dska_16 d451

PVID              244634617431
Serial            rpv
Logical Volume    root
LVID              244634617572

Registered        11/18/83  1025.7 mst Fri
Dismounted        11/05/86  1415.9 mst Wed
Map Updated       11/05/86  1419.3 mst Wed
Salvaged                    08/03/85  1547.7 mst Sat
Bootload                    11/05/86  1417.8 mst Wed
Reloaded                    11/18/83  1134.8 mst Fri
Dumped
  Incremental     11/06/86  1403.7 mst Thu
  Consolidated    11/05/86  1825.0 mst Wed
  Complete        11/01/86  0002.6 mst Sat

Inconsistencies             0

Minimum AIM                 0:000000
Maximum AIM                 7:777777

Volume contains root (>) at vtocx 0
  disk_table_ at vtocx 14006 (uid 075235676151)

Volume Map from Label

First Record       Size
       2600o        4704o     hc        Partition
       7504o          4o      conf      Partition
     1123450          215o    alt       Partition
     1117270          416o    bos       Partition
     1113270          400o    log       Partition
     1024530        6654o     dump      Partition
      756240        4230o     bce       Partition
     1020540          377o    file      Partition
```

The command line

```
ddl dskk_00a
```

produces

```
Label for Multics Storage System Volume fpdir01 on dskk_00a 3380

PVID              535341556672
Serial            fpdir01
Logical Volume    fpdir_1
LVID              535341556533

Subvolume a 1 of 2
Registered        05/24/86  2031.3 mst Sat
Dismounted        11/05/86  1410.8 mst Wed
Map Updated       11/05/86  1419.8 mst Wed
Salvaged
Bootload                    11/05/86  1417.8 mst Wed
Reloaded
Dumped
  Incremental
  Consolidated
  Complete

Inconsistencies             0

Minimum AIM                 0:000000
Maximum AIM                 7:777777
```

Name: dump

SYNTAX AS A COMMAND

```
dump {macro_keyword} {-process_group segment_option
      {...segment_options}} {-control_args}
```

FUNCTION

produces a diagnostic dump of system memory and tables after a hardware or software failure, for later analysis. The dump is produced by copying binary images of segments and directories into the DUMP partition of the disk described by the part dump config card. Arguments to this command specify which processes are to be examined and which segments from those processes are to be dumped. (See "Notes" for a general-purpose command line.) This command is valid at all BCE command levels.

ARGUMENTS

macro_keyword
specifies one of the following default group of processes and segments to dump:

-brief, -bf
is equivalent to -run hc pp moddir.

-long, -lg
is equivalent to -all wrt.

-standard, -std
is equivalent to -run hc pp moddir -elig hc stk -inzr hc stk.

process_group
specifies a group of processes to be considered for dumping. The segments that get dumped for processes in this group are specified by segment options that follow the process group keyword. Allowed groups are:

-all
all processes

-eligible, -elig
all running and eligible processes (processes being considered for running)

-initializer, -inzr
the initializer process (first apte entry)

-running, -run
processes running on a processor (apte.state = running or stopped)

segment_option
specifies a class of segments to be dumped for the group of processes specified by the process group keyword. Segment classes are:

directories, dir
directory segments (aste.dirsw = "1"b)

hardcore, hc
the pds, kst, dseg and ring 0 stack for the process(es). If a process is running, this also dumps the prds for the processor.

modifying_dirs, moddir
directory segments (aste.dirsw = "1"b) that were being modified at the time of the crash (dir.modify ^= "0"b)

per_process, pp
the segments contained within the process directory of the process(es) (aste.per_process = "1"b)

stacks, stk
all stack segments in the process(es) not already dumped by the hc or pp keywords

writeable, wrt
all segments to which the process(es) have write access. This keyword produces a large dump.

Writable ring 0 segments (system databases) other than directories are dumped regardless of what keywords are specified.

Prefixing a segment option with a "^" reverts an earlier occurrence of the given segment option. Thus, you can turn on a macro_keyword and turn off a specific segment option within it.

CONTROL ARGUMENTS

-bce
dumps BCE itself (the dumper).

-crash
specifies that BCE is to dump the saved Multics image.

-drive, -dv drive_name
places the dump into the dump partition of the volume specified instead of into the drive listed on the PART DUMP card, e.g., dska-07 or dskc_01b (for drives with subvolumes).

-dump #
changes the dump number to a desired value. By default, dumps are assigned numbers sequentially.

-force, -fc
places the dump into the DUMP partition without querying you first, even if this means that an existing dump that hasn't been copied will be overwritten. If you give no -fc, the command asks you if the existing dump should really be overwritten before it overwrites it.

-no_sstnt
disables sst_names_ generation. If sst_names_ generation is enabled for the system (by the astk parm in the config deck), -no_sstnt has no effect.

-sstnt
causes the segment sst_names_ (the sst name table) to be filled in and included in the dump. The segment sst_names_ provides a name for each ASTE in the system. This information is of use to dump analysis programs. If sst_names_ generation is enabled for the system (by the astk parm in the config deck), -sstnt has no effect. (Default)

NOTES

For general-purpose dump analysis, the command line

```
dump -std
```

which is equivalent to

```
dump -run hc pp moddir -elig hc stk -inzr hc stk
```

should give you all the useful processes and segments (to produce a smaller dump, remove the "moddir" keyword). For simplicity, and to remove the possibility of operator error, put this command line into a BCE exec_com, either by itself or in a site-supplied crash exec_com.

This command examines the active process table entries (apte) within the specified image. For each entry, the criteria specified through the keywords are used to decide if any segments from this process are to be dumped. If any segments are to be dumped, the segment options are applied to each segment active within that process to decide whether or not they should be dumped. As each process is dumped, the command produces an output line showing the apte number and the dbr value for the process. After scanning all apte entries, if the process in control when Multics crashed was not one of the processes dumped, it is dumped with a status line showing an apte number of zero. This process is dumped with the running and initializer segment options.

A counter and a valid flag are kept within the DUMP partition. When a dump is placed into the partition, the valid flag is set. It is reset when the dump is copied out during Multics service (by the copy_dump exec command). If the dump in the partition has not been copied, the command asks you if it should be overwritten. You can avoid this query by specifying -fc.

This command provides a severity indicator, indicating the success of its operation. You can get this indicator with the severity command/active function. The interpretation of the severity status is:

```
3 - the dump request was never called.
2 - the dump request was entered, but never completed.
1 - the dump was aborted because the DUMP partition contains
    an older dump.
0 - the dump was successfully generated.
```

Name: emergency_shutdown, esd

SYNTAX AS A COMMAND

```
esd
```

FUNCTION

starts an emergency shutdown of Multics. It destroys the saved crash image; therefore use it only after a dump is taken. Use it to prevent storage system damage whenever the system crashes. It is only valid at the "crash" command level.

Name: exec_com, ec

SYNTAX AS A COMMAND

```
ec path {optional_args}
```

SYNTAX AS AN ACTIVE FUNCTION

```
[ec path {optional_args}]
```

FUNCTION

invokes a BCE exec_com. An exec_com is an ASCII file consisting of a series of commands to invoke. BCE uses exec_com version 1, described in the *Multics Commands and Active Functions* manual (AG92). This command is valid at all BCE command levels.

ARGUMENTS

path
: is the pathname of a segment containing commands to be executed and control statements to be interpreted. The entryname of the segment must have the ec suffix, although you can omit the suffix in the command invocation. If you specify only an entryname, i.e., one containing no < or >, the exec_com search list is used to locate the segment.

optional_args
: are character strings to be substituted for special strings in the exec_com segment.

NOTES

When the boot, bce, continue, or reinitialize command is executed by a runnning BCE exec_com, the exec_com is aborted.

Name: fwload, fw

SYNTAX AS A COMMAND

```
fw mpc_name {-control_arg}...{mpc_name{-control_arg}}
```

FUNCTION

loads firmware into the specified MPCs. It scans the config deck to determine the location of the MPC and the type of peripherals involved to determine the firmware and overlays needed. It is not valid at the BCE "early" command level.

ARGUMENTS

mpc_name
is the name of an MPC into which firmware is to be loaded.

CONTROL ARGUMENTS

-channel name, -chn name
is the channel through which the MPC is to be loaded. If you give no -chn, a system-selected channel is used.

NOTES

Use fw only on fully crossbarred disk MPCs. Load noncrossbarred disk MPCs as part of BCE initialization or by using the reinitialize BCE command.

Name: get_flagbox, gfb

SYNTAX AS A COMMAND

```
gfb flagbox_variable
```

SYNTAX AS AN ACTIVE FUNCTION

```
[gfb flagbox_variable]
```

FUNCTION

determines the values of various variables maintained in the BCE flagbox. These variables are also accessible from Multics and therefore allow a small method of communication between BCE and Multics. This command is valid at all BCE command levels.

ARGUMENTS

flagbox_variable
is one of the valid flagbox variables listed below:

N
where N is from 1 to 36. The returned value is the Nth flagbox flag. These flags have true or false values. Some of them are named and can be referred to by their names, as listed below.

auto_reboot
(also flag 1). Used by the auto BCE exec_com. Refer to the *Multics System Maintenance Procedures* manual, Order No. AM81, for more details.

booting
(also flag 2). Used by the auto BCE exec_com.

rebooted
(also flag 4). Used by the auto BCE exec_com.

unattended
(also flag 5). Used by the auto BCE exec_com.

bce_command
a command that is invoked by BCE whenever it reaches a command level. The result is a character string, quoted. This command may be set so that BCE can be set to automatically boot Multics upon a crash, etc. Refer to the *Multics System Maintenance Procedures* manual, Order No. AM81, for more details.

ssenb
a flag set by Multics indicating whether or not the storage system was enabled at the time of a crash. A value of true indicates that an emergency shutdown needs to be performed (or did not succeed).

call_bce
indicates that BCE was called through a program calling call_bce. This may be the result of the user having entering the bce command.

shut
indicates that Multics successfully shut down. If neither shut nor call_bce is set, Multics either encountered a breakpoint, crashed or was manually brought to BCE.

manual_crash
indicates that BCE was invoked manually, either by the user manually forcing a return to BCE (XED 24000) or by the user hitting the EXECUTE FAULT button.

Name: init_files

SYNTAX AS A COMMAND

```
init_files {-control_arg}
```

FUNCTION

wipes out all files in the BCE file system. It is to be used only if a problem is encountered with the BCE file system. This command is valid at all BCE command levels.

CONTROL ARGUMENTS

-force, -fc
clears the BCE file system without querying you first. If this control argument is not used, the init_files command asks you if the BCE file system should really be cleared before it clears it.

NOTES

Refer to the *Multics System Maintenance Procedures* manual, Order No. AM81, for information on reloading the BCE file system after it has been initialized.

Name: list, ls

SYNTAX AS A COMMAND

```
ls {star_names}
```

SYNTAX AS AN ACTIVE FUNCTION

```
[ls {star_names}]
```

FUNCTION

lists the names of BCE files matching a set of star names. If you specify no star names, list lists the names of all BCE files. As an active function, it returns the set of star names. It is valid at all BCE command levels.

ARGUMENTS

star_names
: are the names against which the names of BCE files are to be matched.

Name: list_requests, lr

SYNTAX AS A COMMAND

```
lr
```

FUNCTION

lists all commands valid at the current command level.

Name: lock_mca

SYNTAX AS A COMMAND

```
lock_mca
```

FUNCTION

locks (disables) input to all maintenance channel adapters (MCAs) from the console.

Name: print, pr

SYNTAX AS A COMMAND

```
pr file_name
```

FUNCTION

prints the contents of a file in the BCE file system. This command is valid at all BCE command levels.

ARGUMENTS

file_name
is the name of the BCE file whose contents are to be printed.

Name: probe, pb

SYNTAX AS A COMMAND

```
pb {-control_arguments}
```

FUNCTION

examines, patches, generally debugs the Multics hardcore and BCE itself, and provides a general memory and disk patch/dump facility. Its requests resemble those of the Multics probe command. It can be used at all BCE command levels.

CONTROL ARGUMENTS

-bce
examines bce itself.

-break
examines the active breakpoint.

-crash
examines the saved crash image.

When you invoke probe at the "boot" command level, the default is to examine BCE. When it is invoked automatically upon encountering a breakpoint, the default is to examine the breakpoint. Otherwise, the default is to examine the crash image.

NOTES

This command reads request lines from the bootload console. Multiple requests can appear on one line separated by semicolons. The syntax of these requests varies from request to request. The recognized requests are listed below. Various other aspects of BCE probe are described in the following sections.

ENTERING NUMERICAL VALUES

You can enter all numerical values in the form Nn, where n is the base numbering system designator, e.g., 12d (decimal), 1300 (octal), 116 (binary), 12x (hexadecimal)). If you specify no base numbering system, a default value is assumed as indicated in the text.

LIST OF ADDRESS FORMS

Several requests in probe take an address describing what should be displayed, modified, etc. The address can take many forms, depending on what is desired. Valid address forms are

N
: specifies absolute memory location N. N can describe any location in the memory. N is assumed to be octal if you give no base designator.

M|N
: specifies the virtual location N in segment M. The interpretation of this virtual address depends on the address space being examined; see the dbr and proc requests. Both N and M are assumed to be octal if you give no base designator.

name|N
: specifies the virtual location N in the hardcore segment with the specified name. This interpretation depends on the address space being examined. N is assumed to be octal if you give no base designator.

M$entry
: specifies the virtual location whose address is that of the specified entry in segment M. This interpretation depends on the address space being examined. M is assumed to be octal if you give no base designator.

M$entry+|-N
: specifies the virtual location offset N (plus or minus) from the address of the specified entry in segment M. This interpretation depends on the address space being examined. Both M and N are assumed to be octal if you give no base designator.

name$entry
: specifies the virtual location whose address is that of the specified entry in the hardcore segment with the specified name. This interpretation depends on the address space being examined.

name$entry+|-N
specifies the virtual location offset N (plus or minus) from the address of the specified entry in the hardcore segment with the specified name. This interpretation is subject to the address space being examined. N is assumed to be octal if you give no base designator.

.{+|-N}
specifies the last location referenced (of any address type) optionally offset by the value N. N is assumed to be octal if you give no base designator.

reg(name)
specifies the named register in the crash image. This address is not valid when examining the live BCE. Valid registers are

```
prN (N = 0 to 7)
xN (N = 0 to 7)
a, q, e
t, ralr
fault, ext_fault, mode, cache
dbr, bar
```

disk(drive_name,record_num,offset)
refers to a specific page of a disk drive. The drive is in the standard form dsk<subsys>_<number (nn)>{subvol (sv)}, e.g., dska_07 or dskc_00b (for devices with subvolumes). Both record_num and offset (within the page) are assumed to be octal if you give no base designator.

PROBE REQUESTS

before {address}
b {address}
sets a breakpoint to be executed before executing the instruction at the specified address. If you give no address, "." is assumed. The address must be a virtual address. The breakpoint is added to the list of breakpoints for the segment. You can set up to 120 breakpoints per hardcore segment; however, all wired hardcore segments share the same breakpoint area, so you can set only a total of 120 breakpoints in wired segments.

continue, c
continues the saved image from a breakpoint. It is the same as exiting probe and entering the continue command. Multics is restarted.

dbr {value1 {value2}}
sets the dbr (descriptor base register) value used in the appending simulation used to access virtual addresses in the Multics image. If value2 is omitted, the second word of the dbr value is obtained from the dbr in effect when Multics crashed. Both value1 and value2 are assumed to be octal if a base designator is not specified.

display address {mode {length}}
ds address {mode {length}}
displays a set of locations in a specified mode. If length is omitted, a value of 1 is assumed. For virtual addresses, a length of "*" may be specified to display to the end of the segment. If mode is omitted, octal is assumed. Valid modes are:

```
a - ASCII characters
d - decimal words
i - instruction format
o - octal words (default)
p - symbolic pointer (double words)
```

The locations are displayed four to a line in the desired format. The value of "." after this request finishes is the first location displayed.

let address = value {... value}
l address = value {...value}
modifies a series of locations starting at the address specified. Each value is converted to a number of words and catenated together to form the new value. Valid values are:

STR
a quoted string of characters. To place a quote character into the string, it must be doubled.

N
a decimal number.

No
an octal number.

Nb
a binary number.

M|N
a pointer to segment M offset N (double word).

name|N
a pointer to the named hardcore segment offset N (double word).

list_requests, lr
lists the valid BCE probe requests.

mc address {-long}
mc address {-lg}
displays, in interpreted form, the SCU data found within the machine conditions at the specified address. Specifying -long also dumps the machine registers from the machine conditions.

name segno
displays the name of the hardcore segment with segment number segno.

proc N
changes the address space used by the appending simulation for displaying virtual addresses to the Nth process in the active process table. A value of 1 specifies the Initializer's process.

quit, q
exits probe.

reset {address}
r {address}
resets a given breakpoint; that is to say, Multics will no longer break when the instruction is encountered. The breakpoint causing the return to BCE can be reset by not specifying an address.

segno name
displays the segment number of the named hardcore segment.

stack address
sk address
displays a stack trace starting at the given address. If the word offset of the address is 0, the address is assumed to refer to a stack header. Otherwise it is assumed to refer to a stack frame. For each frame, the stack frame offset, entry pointer, return pointer and argument pointer is displayed.

status {name|segno}
st {name|segno}
either lists all segments with breakpoints set in them (if no name or segno is specified) or lists all offsets within a single segment at which a breakpoint is set.

HARDCORE BREAKPOINTS

The hardcore breakpoint facility is a collection of facilities within Multics and BCE that allow probe style breakpoints to be set at most BCE and hardcore instructions. They may be used largely as they are within normal Multics probe, with a few cautions.

BREAKPOINT MECHANISM

The following paragraphs describe the mechanism by which hardcore breakpoints are implemented. An understanding of this mechanism will prevent the user from setting a breakpoint in an incorrect path; in particular, breakpoints may not be set in the breakpoint handler's path.

When a hardcore breakpoint is set at an instruction, the instruction at that location is relocated to the end of the segment containing it. Its addressing is changed to reflect its new location. The original location is replaced with a transfer instruction to a breakpoint block at the end of the segment which executes a "drl -1" instruction. This causes the breakpoint to happen. If the breakpoint handler returns without changing the breakpoint, the next instruction in the block will be executed. This is the relocated original instruction. After this, a transfer is made back to the correct place in the original program. It should be noted that the instruction moved cannot be the second or later words of an eis multi-word instruction.

Derail faults are handled in fim. A "drl -1" instruction is special-cased to be a breakpoint. Fim makes a call to pmut$bce_and_return to implement the call to BCE. Any program in this path cannot have a breakpoint placed within it. In other words, a breakpoint cannot be set in the path of code which gets executed between a breakpoint and a return to BCE. This path includes the breakpoint handler in fim, the code in pmut$bce_and_return, any code which sends and handles connects to other processors, etc. Also, the special casing of a "drl -1" to be a breakpoint only applies for derails in ring 0. Thus, breakpoints should not be set in segments that will be executed in other rings.

When BCE is invoked via the toehold, it notices that a breakpoint was the cause of the return to BCE and invokes BCE probe directly. Probe is free to perform a continue operation which eventually returns to pmut, restarts other processors, and returns to fim which restarts the breakpointed operation.

Breakpoints may be set within BCE also. However, they should be set only at the "boot" command level. When set at the "early" command level, a breakpoint will cause a return to the "early" command level. Also, a breakpoint set at the "crash" level is useless since, upon a breakpoint/crash of the "crash" command level, the toehold purposely does not save the crash image to avoid overwriting the Multics image already saved.

BREAKPOINT REFERENCES

When a breakpoint causes a return to BCE, BCE does not execute the bce_command in the flagbox. Instead, it enters probe directly. Probe will assume a default of "-break." Probe may be exited at this time. This does not effect a return to Multics however, only a return to BCE ("crash" or "bce_crash") command level. Probe may also be entered with the control argument "-break" to force examining the breakpoint conditions. The only difference between "-break" and "-crash" for probe is the machine conditions to use. The "-crash" control argument uses registers contained within the toehold when the toehold was invoked. These registers are most interesting when BCE is manually entered. The "-break" control argument uses the registers at the time of the breakpoint; these were saved by the breakpoint handler. The registers will show the register contents at the time of the breakpoint; however, the instruction counter will show the relocated instruction, not its original location.

SETTING BREAKPOINTS AT INITIALIZATION

To set a breakpoint in a collection 2 initialization routine:

1. enter bce probe
2. set a breakpoint at real_initializer$collection_2+5

This causes the system to stop during initialization just after the loading of collection2, at which time you can set a breakpoint in any collection 2 routine.

To set a breakpoint in a collection 3 initialization routine:

1. enter bce probe
2. set a breakpoint at real_initializer$collection_3+5

This causes the system to stop during initialization just after the loading of collection3, at which time you can set a breakpoint in any collection 3 routine.

Name: qedx, qx

SYNTAX AS A COMMAND

```
qx {-control_args} {macro_path} {macro_args}
```

FUNCTION

invokes the qedx text editor to edit a BCE file system file. All requests of the standard Multics qedx editor are supported except for the "e" request. For complete information, see the description of qedx in the *Multics Commands and Active Functions* manual, Order No. AG92. This command is valid at all BCE command levels.

ARGUMENTS

macro_path
: specifies the pathname of a segment from which the editor is to take its initial instructions. Such a set of instructions is commonly referred to as a macro. The editor automatically concatenates the suffix "qedx" to macro_path to obtain the complete pathname of the segment containing the qedx instructions.

macro_args
: are optional arguments that are appended, each as a separate line, to the buffer named "args" (the first optional argument becomes the first line in the buffer and the last optional argument becomes the last line). Arguments are used in conjunction with a macro specified by the macro_path argument.

  The editor executes the qedx requests contained in the segment selected and then waits for you to type further requests. If macro_path is omitted, the editor waits for you to type a qedx request.

CONTROL ARGUMENTS

-no_rw_path
: prevents the user from making read (r) or write (w) requests with a pathname. All read and write requests for buffer 0 affect the pathname specified by the -pathname control argument. The -no_rw_path control argument is intended to be used within exec_coms which are providing a limited environment; the user is prevented from examining or altering segments other than the one specified with -pathname.

-pathname path, -pn path
: causes qedx to read the segment given by path into buffer 0, simulating "r path," before executing a macro (see macro_path). This control argument must precede macro_path. If no macro is specified, the user is placed immediately in the editor request loop.

NOTES

Complete tutorial information on qedx is available in the *qedx Text Editor User's Guide*, Order No. CG40.

Name: reinitialize, reinit

SYNTAX AS A COMMAND

```
reinit {-control_arg}
```

FUNCTION

causes BCE to perform a new initialization pass, thereby reflecting any changes to the config deck made since the last such pass. This command returns the operator to "boot" command level. It is valid at the "boot," "bce_crash" and "crash" command levels.

CONTROL ARGUMENTS

-force, -fc
: when used at the "crash" command level, destroys the saved Multics image without querying you first. If this control argument is not used, the reinitialize command asks you if the saved Multics image should really be destroyed before it destroys it.

-time
: specifies that the system is to halt before setting the system clock and query the user as to the desired clock setting.

Name: rename, rn

SYNTAX AS A COMMAND

```
rn star_name equal_name {...  star_names equal_names}
```

FUNCTION

renames files in the BCE file system. The star and equal conventions are used. This command is valid at all BCE command levels.

ARGUMENTS

star_name
: specifies a file or files to be renamed.

equal_name
: specifies the new names(s).

Name: restore

SYNTAX AS A COMMAND

```
restore {-set} cf_1 {...cf_N} {-set cf_1 {...cf_N}} {-rt cf_1 {...cf_N}}
```

FUNCTION

restores the contents of physical volumes from tape.

ARGUMENTS

cf_N
: defines the name of a control file or set of control files that will make up a restore set. You can define up to 32 control file names per restore. Give at least one. See "List of Control File Requests."

CONTROL ARGUMENTS

-restart_set, -restart, -rt
: prefixes a set of control file names to be restarted. You can use it with -set.

-set
: prefixes a set of control file names. The first set of control files do not require this prefix. You can define up to four control file sets.

LIST OF CONTROL FILE REQUESTS

control_file [control_file], cf [control_file]
: enables control files to be linked together, where control_file defines another control file to be examined. For instance, one control file could define all the tape devices for the restore; the other control files could be broken down into logical volumes that only reference the tape device control file, and could then define the physical volumes.

partition [pv_name] [disk_device] [part_name] {...part_name},
part [pv_name] [disk_device] [part_name] {...part_name}
: where pv_name is the name of the physical volume to be restored; disk_device is the standard name "dska_02," or "dske_02c" (for subvolumes); and part_name is the name of the partition to be restored, or "-all," to restore all the partitions that were saved. If you give -all, all partitions defined on the volume that are not restored are zero filled, except for any "alt" or "hc" partitions and the "bce" partition on the rpv. You can define up to 64 partitions per restore set.

physical_volume [pv_name] [disk_device], pv [pv_name] [disk_device]
: where pv_name is the name of the physical volume to be restored and disk_device is the standard name "dska_02," or "dske_02c" (for subvolumes). You can restore up to 63 volumes per set.

tape_device [tape_device] {density}, td [tape_device] {density}
where tape_device is the standard device identifier (i.e., tapa_05) and density is in the form "d=NNNN," "den=NNNN," "-density NNNN," "-den NNNN," or "-d NNNN." During a restore, the save tape defines the density. The order you enter the devices defines the sequence for using them. You can define up to 16 devices per restore set.

tape_set [tape_set_name], ts [tape_set_name]
where tape_set_name is the name of the collection of tapes to be used for the restore. This name (1) can be up to 32 characters, (2) can be defined by the color of the tape reel (e.g., the "blue" set or the "red" set), (3) is part of the tape label and is checked at each tape mount, and (4) appears in parenthesis after the command name in all output messages. Give one ts request per set.

NOTES ON CONTROL FILE REQUESTS

Give only one request per line. Any lines in a control file that begin with /, &, or " are treated as comments. All white space prior to a request in a line is trimmed before processing.

You can restore partitions on a physical volume without having to restore the VTOC and paging regions only by defining a partition request. By defining a partition request, you can also copy a partition from one volume to another, even of different types.

You can edit the control files using the BCE qedx request, or, while the system is running, you can edit and update them in the file partition by either using bootload_fs or regenerating the MST.

NOTES ON RESTORE

The first tape read during a restore is always the "Info" tape, which was the last tape written when the set was saved. This gives restore the necessary information to properly locate items without wasting time spinning tape.

NOTES ON OPERATOR INTERRUPTS

You can interrupt a restore by using the console "request" key. When you depress it while a restore is in progress, you get the message "restore: Abort request:". Give one of the following responses:

abort
causes the command to abort the entire restore and return to BCE command level.

help, ?
causes the command to display all these responses, with a small description of each.

no, n
causes the command to ignore the request and resume the restore.

restart tape_set
allows you to restart the specified tape_set, using its current tape device. You then need to mount the "restart" tape--the tape that you wish to restart from--on the device. Once the set has been restarted, the remaining sets continue operation.

stop tape_set
causes the command to abort the specified tape_set, by marking it complete, and to resume the restore of the other sets.

NOTES ON TAPE ERROR RECOVERY

During a restore, read-data errors may occur that require special handling. The command retries those errors up to eight times; if it is unable to read the data, the error becomes unrecoverable.

When an unrecoverable error occurs, a message is displayed showing the error interpreted in English, with detailed status in hex if required. You are asked to select the appropriate recovery procedure request. Below is an example error output and the possible recovery requests.

```
restore(blue): Device Attention, Handler check on tapa_12.
     detailed status: 20 8C 2B 6D 0A 01 16 00 00 16 48 87 24
                      18 06 00 00 0C 00 00 08 08 80 00 00 00
restore: Action:
```

abort
causes the command to abort the entire restore and return to BCE command level.

help, ?
causes restore to display all these requests, with a small description of each.

remove_device_from_set, remove
works like restart_set, but removes the current tape device from the set and sequences to the next device before going through the restart process. This request is not valid if the current tape device is the only one left in the set.

restart_set, restart, rt
allows you to restart this set, using the current tape device. You are then required to mount the "restart" tape on the device. Once the set has been restarted, the remaining sets continue operation.

retry, r
for errors that can be tried again, forces the retry process to be redone.

skip, s
: skips the unreadable record; restore continues by attempting to read the next record. This response is valid only for unrecoverable data alert errors detected while doing a restore.

stop_set, stop
: causes this set to be aborted, but all others continue.

Name: save

SYNTAX AS A COMMAND

```
save {-set} cf_1 {...cf_N} {-set cf_1 {...cf_N}} {-rt cf_1 {...cf_N}}
```

FUNCTION

saves the contents of physical volumes on tape.

ARGUMENTS

cf_N
: defines the name of a control file or set of control files that will make up a restore set. You can define up to 32 control file names per restore. Give at least one. See "List of Control File Requests."

  You can't specify a control file multiple times for a given set, but you can specify it in more than one set. This way, you can save a set of volumes to several sets of tapes at one time.

CONTROL ARGUMENTS

-restart_set, -restart, -rt
: prefixes a set of control file names to be restarted. You can use it with -set.

-set
: prefixes a set of control file names. The first set of control files do not require this prefix. You can define up to four control file sets.

LIST OF CONTROL FILE REQUESTS

control_file [control_file], cf [control_file]
: enables control files to be linked together, where control_file defines another control file to be examined. For instance, one control file could define all the tape devices for the save; the other control files could be broken down into logical volumes that only reference the tape device control file, and could then define the physical volumes.

partition [pv_name] [disk_device] [part_name] {...part_name},
part [pv_name] [disk_device] [part_name] {...part_name}
where pv_name is the name of the physical volume to be saved; disk_device is the standard name "dska_02," or "dske_02c" (for subvolumes); and part_name is the name of the partition to be saved, or "-all," to save all the defined partitions. You can't save the RPV partition "bce" and any "hc" or "alt" partitions. If you don't give the RPV partitions "conf," "file," or "log" when saving the RPV, a message is displayed stating that they are not being saved, in case you want to save them. You can define up to seven partitions per volume and up to 64 partitions per restore set.

physical_volume [pv_name] [disk_device] {-all},
pv [pv_name] [disk_device] {-all}
where pv_name is the name of the physical volume to be saved and disk_device is the standard name "dska_02," or "dske_02c" (for subvolumes). The -all control argument specifies that all the VTOC and paging records should be saved and salvaged (if required), instead of just saving the paging records in use. You can save up to 63 volumes per set.

tape_device [tape_device] {density}, td [tape_device] {density}
where tape_device is the standard device identifier (i.e., tapa_05) and density is in the form "d=NNNN," "den=NNNN," "-density NNNN," "-den NNNN," or "-d NNNN." The default density is 6250 bpi. The order you enter the devices defines the sequence for using them. You can define up to 16 devices per restore set.

tape_set [tape_set_name], ts [tape_set_name]
where tape_set_name is the name of the collection of tapes to be used for the save. This name (1) can be up to 32 characters, (2) can be defined by the color of the tape reel (e.g., the "blue" set or the "red" set), (3) is part of the tape label and is checked at each tape mount, and (4) appears in parenthesis after the command name in all output messages. Give one ts request per set.

NOTES ON CONTROL FILE REQUESTS

Give only one request per line. Any lines in a control file that begin with /, &, or " are treated as comments. All white space prior to a request in a line is trimmed before processing.

You can save partitions on a physical volume without having to save the VTOC and paging regions only by defining a partition request.

You can edit the control files using the BCE qedx request, or, while the system is running, you can edit and update them in the file partition by either using bootload_fs or regenerating the MST.

NOTES ON SAVE

When a save set is complete, write one last tape, called the "Info" tape. That tape contains information used during a restore to quickly locate the tapes the items are on.

NOTES ON OPERATOR INTERRUPTS

You can interrupt a save by using the console "request" key. When you depress it while a save is in progress, you get the message "save: Abort request:". Give one of the following responses:

abort
causes the command to abort the entire save and return to BCE command level.

help, ?
causes the command to display all these responses, with a small description of each.

no, n
causes the command to ignore the request and resume the save.

restart tape_set
allows you to restart the specified tape_set, using its current tape device. You then need to mount the "restart" tape on the device. The restart tape is either the last good tape written or the current tape (as long as the tape label has been written). Once the set has been restarted, the remaining sets continue operation.

stop tape_set
causes the command to abort the specified tape_set, by marking it complete, and resume the save of the other sets.

NOTES ON TAPE ERROR RECOVERY

During a save, write-data errors may occur that require special handling. The command retries those errors up to eight times; if it is unable to read the data, the error becomes unrecoverable. (The retries consist of a backspace and erase followed by the original write.)

When an unrecoverable error occurs, a message is displayed showing the error interpreted in English, with detailed status in hex if required. You are asked to select the appropriate recovery procedure request. Below is an example error output and the possible recovery requests.

```
restore(blue): Device Attention, Handler check on tapa_12.
     detailed status: 20 8C 2B 6D 0A 01 16 00 00 16 48 87 24
                      18 06 00 00 0C 00 00 08 08 80 00 00 00
restore: Action:
```

abort
causes the command to abort the entire save and return to BCE command level.

help, ?
causes save to display all these requests, with a small description of each.

remove_device_from_set, remove
works like restart_set, but removes the current tape device from the set and sequences to the next device before going through the restart process. This request is not valid if the current tape device is the only one left in the set.

restart_set, restart, rt
allows you to restart this set, using the current tape device. You are then required to mount the "restart" tape on the device. Once the set has been restarted, the remaining sets continue operation.

retry, r
for errors that can be tried again, forces the retry process to be redone.

stop_set, stop
causes this set to be aborted, but all other sets continue.

Name: set_flagbox, sfb

SYNTAX AS A COMMAND

```
sfb flagbox_variable value
```

SYNTAX AS AN ACTIVE FUNCTION

```
[sfb flagbox_variable value]
```

FUNCTION

changes the values of various flagbox variables. When used as an active function, it also returns the previous value of the variable. It is valid at all BCE command levels.

ARGUMENTS

flagbox_variable
is one of the valid flagbox variables listed below:

N
where N is from 1 to 36. The returned value is the Nth flagbox flag. These flags have true or false values. Some of them are named, and can be referred to, by their names, as listed below.

auto_reboot
(also flag 1). Used by the auto BCE exec_com. Refer to the *Multics System Maintenance Procedures* manual, Order No. AM81, for details.

booting
(also flag 2). Used by the auto BCE exec_com.

rebooted
(also flag 4). Used by the auto BCE exec_com.

unattended
(also flag 5). Used by the auto BCE exec_com.

bce_command
a command that is invoked by BCE whenever it reaches a command level. The result is a quoted character string. You can set this command so that BCE can automatically boot Multics upon a crash, etc. Refer to the *Multics System Maintenance Procedures Manual*, Order No. AM81, for details.

ssenb
a flag set by Multics indicating whether or not the storage system was enabled at the time of a crash. A value of true indicates that an emergency shutdown needs to be performed (or did not succeed).

call_bce
indicates that BCE was called through a program calling call_bce. This may be caused by your having entering the bce command.

shut
indicates that Multics successfully shut down. If neither shut nor call_bce is set, Multics encountered a breakpoint, crashed, or was manually brought to BCE.

manual_crash
indicates that BCE was invoked manually, either by your manually forcing a return to BCE (XED 24000) or by your hitting the EXECUTE FAULT button.

value
is either a character string (for the bce_command variable) or the string "true" or "false" (for other flagbox variables).

Name: severity

SYNTAX AS A COMMAND

```
severity prog_name
```

SYNTAX AS AN ACTIVE FUNCTION

```
[severity prog_name]
```

FUNCTION

returns the severity, or extent of completion, of a preceding BCE command. This command is valid at all BCE command levels. Currently, the dump command provides such a severity status. Future BCE commands may also.

ARGUMENTS

prog_name
is the name of the command whose severity is to be checked.

Name: shutdown_state, sds

SYNTAX AS A COMMAND

```
sds
```

SYNTAX AS AN ACTIVE FUNCTION

```
[sds]
```

FUNCTION

returns the state of completion of the Multics shutdown. It does this by examining the shutdown_state flag in the label of the RPV. This command is valid at all BCE command levels.

NOTES

The interpretation of the shutdown states follows:

```
0       Normal Multics shutdown (no esd)
1       esd part 1 started (memory flush of modified pages
        of segments)
2       esd part 1 completed
3       shutdown or esd completed with lock errors
4       shutdown or esd completed with no errors
other   shutdown completed with errors, or not completed for
        one or more disk errors
```

Name: test_disk, td

SYNTAX AS A COMMAND

```
td key device {-control_args}
```

FUNCTION

tests disks for errors.

ARGUMENTS

key
must be one of the following:

w
for writing to specified pages

r
for reading specified pages

wr
for writing and then rereading/checking specified pages

rw
for reading and then rewriting specified pages

device
is a disk known to the system, e.g., dska_03 or dskc_06c (for devices that have subvolumes).

CONTROL ARGUMENTS

-first_record N, -frec N
specifies the first record to be read and/or written. The default is the first record number for the device type unless a partition is specified (using -partition), in which case the default is the first record of the partition. You can use the keywords "first" and "last" instead of a record number to specify the first or last record of the device (or partition).

-last_record N, -lrec N
specifies the last record to be read and/or written. The default is the last record number for the device type unless a partition is specified (using -partition), in which case the default is the last record of the partition. You can use the keywords "first" and "last" instead of a record number to specify the first or last record of the device (or partition).

-n_record N, -nrec N
specifies the number of records to be read and/or written. Specifying -frec and -nrec causes test_disk to read and/or write to N records, where the first record to be operated on is specified by -frec. Specifying -lrec and -nrec causes test_disk to read and/or write to N records, where the last record to be operated on is specified by -lrec. To operate on more than one record, use two of the following: -frec, -lrec, -nrec.

-partition part_name, -part part_name
specifies a partition, where part_name is the name of the desired partition. If you give a partition, the defaults for the first and last page become the first and last page of the partition.

-pattern word1 word2 word3..., -pat word1 word2 word3...
specifies the pattern of words to be written to a record. You can specify an arbitrary number of words. (Default: an all-zero page)

-record N, -rec N
specifies a single page to be read or written. You can use "first" and "last."

NOTES

If the disk is going to be written to, test_disk shows what page(s) are going to be written to, then asks the operator whether or not it should continue. If the operator tells it to continue, it queries the operator again if reading the disk's label shows that it is a Multics storage system device.

If the disk is of a type that allows ALT partitions, test_disk does not operate within the bounds of where the ALT partition would be, unless overridden by -lrec.

Name: unlock_mca

SYNTAX AS A COMMAND

```
unlock_mca mca_number
```

FUNCTION

unlocks (enables) console input to the MCA specified by the argument.

ARGUMENTS

mca_number
is the decimal number of the MCA to be unlocked.

SECTION 10

BOS COMMANDS

The commands in this section can be used only when communicating with the bootload operating system (BOS). The only method of communicating with the bootload operating system is through the bootload console.

Name: BOOT

SYNTAX AS A COMMAND

```
BOOT {number} {COLD}
```

FUNCTION

initiates a bootload of Multics.

ARGUMENTS

number
is the number of the tape drive to boot from; number is always assumed to be a decimal number. If a number is not given, tape drive 3 will be used.

COLD
initiates a cold bootload of the Multics directory hierarchy, including an initialization of the RPV label, which destroys any existing hierarchy. It should be used only after express instructions of the programming staff. This option causes Multics to enter the init_vol request loop to initialize the RPV before entering ring one command level.

NOTES

If, at boot time, the calendar clock indicates a time earlier than the last use of the storage system (as recorded on the RPV label), the boot is aborted with a diagnostic. If a "boot_delta" parameter is provided on the clok config card (described in the *Multics System Maintenance Procedures* manual, Order No. AM81), and the clock indicates that the storage system has not been used for that many hours or more, you are asked whether the clock is indeed correct before booting proceeds; booting is aborted if not. In this case the BOOT command prints out the time the hierarchy was used, and the current time (as read from the clock).

Name: BOSTAP

SYNTAX AS A COMMAND

```
BOSTAP n {BRIEF} {RUNCOM} {density}
```

FUNCTION

generates a new BOS tape, which can be bootloaded or read by LOADDM as any other BOS tape. This tape contains all commands, runcoms, and config decks which have been loaded into the current BOS command directory. All editing changes of saved config decks will appear on the new tape. All new versions of commands, etc., which may have been read in via LOADDM will appear that way on the new tape.

ARGUMENTS

n
specifies a tape drive. If n is not supplied, drive 1 is assumed.

BRIEF
suppresses the tape listing, typed on the bootload console by default.

RUNCOM
causes only runcoms and config decks to be written.

density
may be D=556 or D=800 or D=1600 to specify the density at which to record the tape. Only one density may be specified; it must be specified in the form shown here. The default density is D=800.

NOTES

The config deck currently in use is not written onto the new BOS tape. If it is to be retained, the SAVE command must be issued before the BOSTAP command. If the RUNCOM control argument is given, only runcoms and config decks are written on the tape. A summary of what is being written and its length in words is written on the bootload console. If the BRIEF control argument is given, this summary is not printed.

Name: CARDS

SYNTAX AS A COMMAND

```
CARDS channel_number
```

FUNCTION

causes input to be accepted by the card reader.

ARGUMENTS

channel_number
must be specified the first time this command is used, and when changing card readers; otherwise the argument is optional.

NOTES

Reading of input reverts to the previous source when a Multics EOF card is read. For more information, see the description of BOS input sources in the *Multics System Maintenance Procedures* manual, Order No. AM81.

Name: COLD

SYNTAX AS A COMMAND

```
COLD model channel drive
```

FUNCTION

reloads the entire BOS partition. (It is one of the BOS loader control commands.) The BOS command directory is reset, the configuration deck (residing on disk) and machine conditions are cleared, and the contents of the BOS tape are loaded into the BOS partition.

The BOS loader reads the label of the disk pack specified to it by the COLD loader control command. The disk drive number, model number and channel number of the disk subsystem where the pack is mounted must all be specified on the command line. Refer to the description of the prph config card in the *Multics System Maintenance Procedures* manual, Order No. AM81, for model numbers.

ARGUMENTS

model
: is the model number, with decimal point, of the disk drive, e.g., "400.".

channel
: is the number of the IOM channel connected to the disk subsystem. This number must be less than 40 (octal) or 32 (decimal).

drive
: is the number of the drive on which the BOS residence volume is mounted. This number must be decimal. This disk pack is normally the root physical volume (RPV).

EXAMPLES

```
COLD 400. 30 1.
```

In this example, when BOS is bootloaded, the loader reads the label of the MSU0400 disk pack at channel 30 (octal), drive number 1. The label of this pack is checked for the existence of a BOS partition, and the extent of that partition found in the label is used to load BOS.

When a COLD control command is used to load BOS, a message of the following form is printed:

```
BOS PARTITION AT frec FOR nrec
```

where frec is the first record of the BOS partition and nrec is the number of records. If the volume label cannot be interpreted following a COLD command, one of the following messages will appear:

```
CANNOT READ VOLUME LABEL
VOLUME DOES NOT HAVE A MULTICS LABEL
BOS PARTITION NOT DEFINED IN VOLUME LABEL
```

Name: CONFIG

SYNTAX AS A COMMAND

```
CONFIG {keyword}
```

FUNCTION

reads configuration cards from the same input source that was used to issue the CONFIG command. (The CARDS and TTY commands cannot be used to change the input source while configuration cards are being read.) For more information, see the description of BOS command language in the *Multics System Maintenance Procedures* manual, Order No. AM81.

ARGUMENTS

keyword
can be chosen from the following:

L
The L keyword is followed by configuration cards to be loaded as replacements for all existing configuration cards. The configuration cards must be followed by a QUIT request that returns control to the BOS control program.

L name
The BCD file in the BOS command directory with name "name" is read to input the configuration deck.

SAVE name
The configuration deck is saved in the BOS command directory in the BCD file with name "name."

A

The A keyword is followed by configuration cards that are to be added to the existing configuration deck. The additional configuration cards must be followed by a QUIT request.

C

The C keyword is followed by configuration cards that specify changes in the existing configuration. The configuration changes must be followed by a QUIT request. Each configuration change card causes a search to be made through the existing configuration deck to compare card fields with fields in the existing configuration entries. The first card in the existing configuration deck that matches both the first and second fields of the card is replaced by the change card data. If no match is found, the first card in the existing configuration deck that matches the first field of the card is replaced by the change card data. If no match is found in either of the above comparisons, the data in the change card is added to the existing configuration deck.

Example of configuration deck:

```
cpu a 4
mem c 512. on
mem d 512. off
```

Changes:

```
CPU B 5
MEM D 512. ON
TBLS PRDS 11.
```

New configuration deck:

```
cpu b 5
mem c 512. on
mem d 512. on
tbls prds 11.
```

P {arguments}

Prints the current configuration deck. If arguments are supplied, only configuration cards whose first fields match entries in the arguments list are printed.

Example:

```
CONFIG P CPU TBLS
```

Output:

```
cpu b 5
tbls prds 11.
```

D {arguments}
Deletes selected configuration cards by performing a first field match with the arguments list. All occurrences of a specified configuration card are deleted.

R
The CONFIG R command operates in the same manner as the CONFIG C command except that an error message is printed if the fields of the configuration change card being input do not match fields in the configuration deck. CONFIG R replaces configuration data; it does not add to the existing configuration.

Name: CONTIN

SYNTAX AS A COMMAND

```
CONTIN
```

FUNCTION

restores the machine image and continues running an interrupted activity (usually BCE). CONTIN allows the machine image to be restored so that Multics can continue running from the point at which BOS was entered.

Name: CORE

SYNTAX AS A COMMAND

```
CORE SAVE n {density}
```

or

```
CORE RESTOR {GO} n
```

FUNCTION

saves and restores main memory onto tape.

ARGUMENTS

n
is an optional decimal tape number; the default is 1. The machine registers and main memory image are written onto a tape on the drive specified by n.

density
is the density specification (D=556, D=800, D=1600, or D=6250) for the density at which to record the tape. Only one density may be specified. The default density is D=800.

GO
specifies that the machine image is to be restarted; that is, the interrupted activity (usually Multics) is to be resumed.

n
is the optional decimal tape number; the default is 1.

Name: DELETE

SYNTAX AS A COMMAND

```
DELETE name
```

FUNCTION

deletes a specified entry in the BOS directory.

ARGUMENTS

name
is an entry in the BOS directory. Only one name argument is accepted.

Name: DIE

SYNTAX AS A COMMAND

```
DIE
```

FUNCTION

destroys the data BOS has stored in main memory and on the BOS partition, ensuring that a test version of BOS will not accidentally be used. This command is not used during normal operation.

Name: DMP355

SYNTAX AS A COMMAND

```
DMP355 {tag} {ABS}
```

FUNCTION

dumps FNP information onto the online printer.

ARGUMENTS

tag
: dumps trace table contents and main memory from the FNP specified by tag on the online printer. Tag may be A, B, C, or D; if tag is not specified, all configured FNPs are dumped.

ABS
: dumps only the main memory from the specified FNPs.

NOTES

The FNP image is read into main memory and printed. Thus, the printer should be configured to the main-frame and not to the FNP. This comand dumps only the first 32K of the first four configured FNPs.

*

Name: EDIT

SYNTAX AS A COMMAND

```
EDIT {file_name}
```

FUNCTION

provides a primitive text editing function in BOS for BCD files. The EDIT command accepts input only from the bootload console. It operates in two modes, input mode and edit mode. In input mode, lines typed at the bootload console are placed directly into the BCD file being edited. In edit mode, lines typed at the bootload console are interpreted as editing commands. The EDIT command will go from input mode to edit mode whenever a blank line is typed. A pointer to the current line being edited is maintained. Editing commands may cause this pointer to be moved. The EDIT command can be used either to edit an existing BCD file or to create a new one.

ARGUMENTS

file_name
is the name of the file to be edited. The EDIT command issued without file_name enters input mode to accept input for a new BCD file. A file name must be specified when the W command is issued.

LIST OF REQUESTS
The following is a description of the requests accepted by the EDIT command while in edit mode and their effects:

N {n}
The current line pointer is advanced n lines. If n is not specified, it is advanced one line.

- {n}
The current line pointer is backed up n lines. If n is not specified, it is backed up one line.

P {n}
n lines are printed on the bootload console starting with the line pointed to by the current line pointer. If n is not specified, only one line is printed. The current line pointer will point to the last line printed.

I
Input mode is entered. Lines input to the bootload console will be inserted before the line pointed to by the current line pointer. An empty line causes exit from input mode.

T
The current line pointer is set to the first line of the file.

B
The current line pointer is moved to the bottom of the file. Input mode is entered. Lines input from the bootload console is appended to the end of the file. An empty line causes exit from input mode.

D {n}
n lines are deleted, starting with the line pointed to by the current line pointer. If n is not specified, only one line is deleted. The current line pointer is left pointing to the first line not deleted.

Q
The edit command returns to the BOS command processor without saving the edited file.

W {name}
The edited file is written out onto disk and entered in the BOS directory as BCD file name. If name is not specified, the name used in the original EDIT command is used. The EDIT command processor then returns to the BOS command processor.

*

Name: FD355

SYNTAX AS A COMMAND

```
FD355
```

FUNCTION

dumps the contents of all configured FNP memories and places them in an appropriate area of the DUMP partition.

NOTES

The FD355 command dumps only the first 32K of the first eight configured FNPs. This command is of limited utility because the state of an FNP at the time Multics crashes offers limited diagnostic information. The online FNP dumps produced by the answering service in the >dumps directory are much more useful. For this reason, it * is recommended that the FD355 command not be part of the standard crash runcoms. *

Name: FMT

SYNTAX AS A COMMAND

```
FMT volume_designator extent {keywords}
```

FUNCTION

formats a 451 disk pack or a portion of a 451 disk pack for its use by Multics. This command cannot be used to format fixed media disk devices (MSU0500 and MSU0501 disks.)

In general, new disk packs should be formatted during Multics operation by using the online T&D tool, MTR, under TOLTS and MOLTS. Procedures for formatting both MSU0451 and MSU0500/MSU0501 disk packs with MTR are described in the *Multics Online Test and Diagnostics Reference Manual*, Order No. AU77.

ARGUMENTS

volume_designator
indicates a particular volume or part of a volume. It may be specified in one of the following ways:

subsystem N
where N is a drive number. If N ends in a decimal point, it is considered a decimal number; otherwise it is considered an octal number.

PART partition_name

ROOT

extent
is a range of addresses on a single volume. It may be specified in one of the following forms:

```
device_position TO number
device_position FROM number
device_position ONLY
ENTIRE
```

where:

device_position
specifies a starting or ending device address; it may have one of these forms:

```
RECORD record_number
SECTOR sector_number
record_number
```

number
is either a record number or a sector number, depending on the units of the device_position argument.

keywords
may be chosen from:

CHECK
The track headers are read to look for defective tracks. Good tracks are not reformatted unless record 0 of the track contains incorrect data.

CLEAR
All tracks are formatted as good tracks.

DEFECT
All tracks within the volume-designator and extent are formatted as defective.

The addresses of defective tracks are printed unless the DEFECT keyword is used. If no keywords are present, the track headers are read first to look for defective tracks. All tracks are then formatted as good or defective.

EXAMPLES

```
FMT DSKA 3 ENTIRE
FMT DSKA 4 0 TO 5 CHECK
```

NOTES

If the track headers cannot be read or the pack has never been formatted before, the CLEAR keyword must be used.

Name: FWLOAD

SYNTAX AS A COMMAND

```
FWLOAD ctlr iom chan
or
FWLOAD ur_ctlr iom chan nchan devl...devn
```

FUNCTION

causes MPC firmware to be loaded. This command has two uses; it loads tape or disk controller firmware and it loads unit record controller firmware (plus the necessary firmware overlays).

ARGUMENTS

ctlr
| is an MPC designation for a tape or disk controller. The following values of ctlrs are recognized:

ctlr	MPC_type
T500	MTC501 or MTC502
T501	MTC501
T502	MTC502
T600	MTP0600
T601	MTP0601
T602	MTC0602
T610	MTP0610
T611	MTP0611
* D400	MSP0400
D450	MSP0451 or DSC0451
D451	MSP0451
D500	MSP0601
D601	MSP0601
D603	MSP0603
D607	MSP0607
D609	MSP0609
D611	MSP0611
D612	MSP0612

The MSP0609 and MSP0612 models require loading for each half of the controller individually.

iom
is the tag of the IOM to which the MPC is connected.

chan
is the number of the IOM channel to which the MPC is connected.

ur_ctlr
| is an MPC designation for a unit record controller. The following values of ur_ctlrs are recognized:

ur_ctlr	MPC_type
U002	URC002
U600	URP0600

*
*
nchan
is the number of device ports in the unit record MPC.

dev1...devn
is a list of devices indicating what peripheral is connected to each of the unit record MPC device ports. The following devices specified by devi are recognized.

devi	peripheral
CRP	card reader/punch
CRZ	card reader
CPZ	card punch
PR4	PRT401/402, PRU1000/1200/1600
NONE	no peripheral connected to this port

NOTES

When the FWLOAD command is invoked automatically during the loading of BOS, it accepts requests from the bootload console to load MPC firmware. The syntax of the requests is identical to that given above, except that the command name FWLOAD is not given.

Name: GO

SYNTAX AS A COMMAND

```
GO
```

FUNCTION

is identical to the CONTIN command.

Name: IF

SYNTAX AS A COMMAND

```
IF {NOT} var {test value} command arg1 ... argn
```

FUNCTION

tests the value of several variables in the Multics and BOS environments. The IF command is particularly useful within BOS runcom files. Another BOS command is executed conditionally on the results of the test.

ARGUMENTS

NOT
if present reverses the sense of the test.

var
performs various tests which may be chosen from the following list:

SHUT
causes the Multics system shutdown state to be tested. Possible values are as follows:

Code (Octal)	Meaning
0	Normal Multics operation (no ESD)
1	ESD part 1 started
2	ESD part 1 completed
3	Shutdown or ESD completed with lock errors
4	Shutdown or ESD completed with no errors
Other	Shutdown completed with errors, or not completed for one or more disk drives

SWITCH mask
reads the processor data switches and masks them with the specified octal
* mask before comparison.

RTB argument
tests the reason for a return from Multics to BOS. If argument specifies the actual reason for a return to BOS, its value will be nonzero. Possible arguments are as follows:

argument	reason
SHUT	Normal Multics shutdown.
CALL	Operator call to BOS.

*

test
is either EQ to test for equality or NEQ to test for inequality.

value
value is specified as an octal number. It will be used in a test for equality or inequality depending on whether EQ or NEQ was specified. If {test value} is omitted, the default test of NEQ 0 is performed.

command arg1 ... argn
is the name of any BOS command and its valid arguments. The name of an appropriate runcom may also be used. If the condition specified by {test value}
* is met, command is invoked with the arguments arg1...argn.

Name: LIST

SYNTAX AS A COMMAND

```
LIST
```

FUNCTION

prints a list of all the current disk resident BOS modules and BCD files and their device addresses.

Name: LOADDM

SYNTAX AS A COMMAND

```
LOADDM {n} {name1...namen}
```

FUNCTION

reads modules and ASCII files from the BOS system tape and places them in the BOS directory.

ARGUMENTS

n
: is the decimal number of the tape drive on which the BOS system tape is read. If n is not specified, drive number 1 is used.

namei
: are names of modules or BCD files to be read from the BOS system tape. Only the last copy of a module or BCD file is remembered. When no namei is specified, all segments on the BOS system tape are loaded.

Name: MPCD

SYNTAX AS A COMMAND

```
MPCD {devname} {TRACE} {DUMP}
```

FUNCTION

dumps MPC memory and the trace table kept by the MPC.

ARGUMENTS

devname
is the name of a device connected to the MPC. The device must be described in the configuration deck by a prph card and devname must be in the form used on that card for a peripheral device name.

TRACE
if present, dumps the MPC trace table.

DUMP
if present, dumps the MPC read/write memory.

NOTES

When dumping the MPC trace table, switch 4 on the appropriate MPC should be placed in the UP position in order to inhibit further tracing.

*
*

Name: NLABEL

SYNTAX AS A COMMAND

```
NLABEL model channel drive frec nrec
```

FUNCTION

is used for the first bootload of BOS on an empty BOS residence volume, rather than the COLD command, since the pack has no label. (It is one of the BOS loader control commands.) This command supplies the first record and number of records for the BOS partition. It is used only when a new Multics site is first installed and the Multics file system has not been loaded.

ARGUMENTS

model
is the model number, with decimal point, of the disk drive, e.g., "400.".

channel
is the number of the channel connected to the disk subsystem.

drive
is the number of the drive on which the disk is mounted. This number must be decimal.

frec
is the first record of the BOS partition.

nrec
is the number of records in the BOS partition.

NOTES

The NLABEL control command should be used only when a cold boot of Multics is to follow the loading of BOS. An NLABEL command will initialize the label of the BOS residence volume. The NLABEL command should be used only when booting Multics for the first time, since it can destroy the contents of the storage system if packs have been moved.

The partition defined by the free and nrec arguments should represent the last nrec records of the disk pack.

EXAMPLES

```
NLABEL 400. 30 1. 19000. 270.
NLABEL 451. 30 1. 37988. 270.
NLABEL 500. 30 1. 37988. 270.
```

In the first example, the 270-record BOS partition occupies records 19000-19269 of the MSU0400 pack on drive 1, channel 30 (octal) of the bootload IOM. For the other two models, records occupied are 37988-38257.

Name: PROMPT

SYNTAX AS A COMMAND

```
PROMPT state
```

FUNCTION

controls input prompting on the bootload console.

ARGUMENTS

state
is either ON, to print a prompt string when BOS is ready to accept input, or OFF, for no prompt.

Name: QUIET

SYNTAX AS A COMMAND

```
QUIET state
```

FUNCTION

controls printing of RUNCOM and card reader input lines on the bootload console.

ARGUMENTS

state
is either ON, to print input as it is processed, or OFF, for no printing.

Name: RDLABL

SYNTAX AS A COMMAND

```
RDLABL subsystem N
```

FUNCTION

reads a disk volume label, then displays it on the bootload console.

ARGUMENTS

subsystem
is the name of the disk subsystem.

N
is the disk drive number.

EXAMPLES

To read the volume label of disk drive 5 in subsystem dska, type:

```
RDLABEL DSKA 5
```

If dska 5 is a storage system volume, Multics will print a message like this:

```
DSKA 5 IS MULTICS STORAGE SYSTEM VOLUME LDD2.
LABEL LAST WRITTEN AT 7/31 20:33
```

If dska 5 is not a storage system volume, Multics will print this message:

```
NOT A MULTICS STORAGE SYSTEM VOLUME.
```

Name: READY

SYNTAX AS A COMMAND

```
READY state
```

FUNCTION

controls printing of the BOS ready message on the bootload console.

ARGUMENTS

state
is either ON, to print a ready message of the form:

```
BOS <system_id> AT <hhmm.ss> <zone> <day> <date>
```

after each command executes, or OFF, for no prompt.

Name: RENAME

SYNTAX AS A COMMAND

```
RENAME name1 name2
```

FUNCTION

renames an entry in the BOS directory to another name.

ARGUMENTS

name1
is the name of an entry in the BOS directory that is renamed to the name specified by name2.

Name: RESTOR

SYNTAX AS A COMMAND

```
RESTOR {name {n1 n2 ... n8}}
```

FUNCTION

reloads the contents of disks from a set of tapes or disks written by the BOS SAVE command.

ARGUMENTS

name
: may be either of the strings "TAPE" or "DISK subsystem". If not given, the default is "TAPE".

ni
: is the drive number on which the tape or disk to be reloaded is mounted; up to eight tape drives or disk drives may be specified. These drives are used in the order specified. If no drive numbers are specified, drives 1 and 2 are used. Drive numbers are always assumed to be decimal numbers.

NOTES

RESTOR reads lines from the BOS input source until an END or QUIT request is read. Each request specifies one block of a physical volume to be reloaded, in the form:

```
volume_designator extent vol_name
```

where:

volume_designator
: indicates a particular volume or part of a volume. It may be specified in one of the following way:

  subsystem N
  : where N is a drive number. If N ends in a decimal point, it is considered a decimal number; otherwise, it is considered an octal number.

  PART partition_name

  ROOT

extent
is a range of addresses on a single volume. It may be specified in one of the following forms:

```
device_position TO number
device_position FROM number
device_position ONLY
ENTIRE
```

where:

device_position
specifies a starting or ending device address; it may have one of these forms:

```
RECORD record_number
SECTOR sector_number
record_number
```

number
is either a record number or a sector number, depending on the units of the device_position argument.

vol_name
is the name of the physical volume.

The vol_name specified at SAVE time is checked at RESTOR time, so that physical volumes need not be remounted on the same device addresses. When an END request is encountered, the input requests are compared with the tape or disk header of the input. If they do not match, the following occurs:

1. The record address ranges of the SAVE sequence are printed.
2. The record address ranges of RESTOR input requests are printed.
3. You are is asked if this is the correct tape or disk. If you answer NO, control is returned to BOS.
4. If you answer YES, you are is asked if the entire SAVE sequence should be loaded.
5. If you answer YES, RESTOR input requests are changed to equal the input requests of the SAVE sequence, resulting in all of the SAVE sequence being read. Otherwise, the RESTOR is executed as originally indicated on the input lines.

See the SAVE command description before using the RESTOR command.

EXAMPLES

The following are examples of request lines to the RESTOR command. These lines are not related to each other.

```
DSKA 3 0 FOR 6 ROOT2
DSKA 3 12 TO 17 RPV
DSKA 1 0 ONLY ROOT3
DSKA 3 ENTIRE ROOT2
ROOT ENTIRE RPV
PART LOG ENTIRE RPV
```

The following is a complete example of use of the RESTOR command:

```
RESTOR TAPE 1 2 3
DSKA 2 ENTIRE RPV
DSKA 18. ENTIRE ROOT2
END
```

Name: RUNCOM

SYNTAX AS A COMMAND

```
RUNCOM {-control_arg}
```

FUNCTION

creates, executes, and otherwise manipulates BOS runcom files. See the discussion of BOS runcom files in the *Multics System Maintenance Procedures* manual, Order No. AM81.

CONTROL ARGUMENTS

LOAD name
: creates a BOS runcom file by loading input lines until a RUNCOM END card image is found. The name is added to the BOS command directory, and the file is written on the disk.

PRINT name
: prints the runcom file specified by name on the bootload console.

SKIP name {arg s}
prints successive lines of the runcom file and unlocks the bootload console keyboard after each. If the line just printed is not to be executed, press EOM and the next line is printed. If the line is to be executed, type "X". If the line is to be replaced by another line, type "X" followed by the replacement line. If the line is to be executed followed by all subsequent lines in the runcom file, type "C".

RUN name {arg s}
executes a runcom file. The file may also be executed by simply typing its name as a command to BOS, followed by arguments to the runcom, if any. This forces the execution of "RUNCOM RUN name".

PUNCH name
punches the BCD file name on cards. (A prph card for puna must be present in the configuration deck.)

Name: SAVE

SYNTAX AS A COMMAND

```
SAVE {name n1 n2...n8} {ALL} {BRIEF} {density}
or
SAVE COPY subsys {n1...n8} {ALL} {BRIEF}
```

FUNCTION

writes the contents of disks onto tape or disk. It can also be used to create an exact duplicate of a disk pack (see "Notes on Save Copy" below).

ARGUMENTS

name
can be either "TAPE" or "DISK subsystem". If not given, the default is "TAPE".

ni
is the decimal drive number of the output medium for the SAVE. Up to eight tape drives or disk drives may be specified; they are used in the order specified in either the SAVE TAPE or SAVE DISK usage line specified above. If no drive numbers are specified, drives 1 and 2 are used.

ALL
saves all of the Multics storage system records of the pack(s). (See "Notes on Save" below.)

BRIEF
causes suppression of all recoverable disk error messages.

density
may be D=800, D=1600, or D=6250 to specify the density (bpi) at which to record the tape. Only one density may be specified. The default is D=1600. This argument can only be used with "TAPE."

subsys
is the name of the disk subsystem to which the drive being used for the SAVE COPY belongs.

NOTES ON SAVE

Normally, SAVE writes on tape or disk only those Multics storage system records that currently contain useful data. Partitions are not normally saved. This is done by using the system-maintained table of assigned records for each volume, called the volume map. A volume map contains valid information only after a successful system shutdown, volume salvage, or demount of the volume. If for some reason a salvage cannot be performed, you can include the ALL argument to have SAVE ignore the volume map contents. If the ALL argument is not present, a successful shutdown, demount, or salvage must have preceded the SAVE.

The SAVE command reads request lines from the BOS input source until a QUIT or END request is read. Each request specifies one block of a physical volume to be dumped, in the form:

```
volume_designator extent vol_name
```

where:

volume_designator
indicates a particular volume or part of a volume. It may be specified in one of the following ways:

subsystem N
where N is a drive number. If N ends in a decimal point, it is considered a decimal number; otherwise it is considered an octal number.

PART partition_name

ROOT

extent
is a range of addresses on a single volume. It may be specified in one of the following forms:

```
device_position TO number
device_position FROM number
device_position ONLY
ENTIRE
```

where:

device_position
specifies a starting or ending device address; it may have one of these forms:

```
RECORD record_number
SECTOR sector_number
record_number
```

number
is either a record number or a sector number, depending on the units of the device_position argument.

vol_name
is the name of the physical volume.

When saving to tape, an additional request line of the form:

```
DEN {density}
```

(where density may be 800, 1600, or 6250) may be used. If no density is specified then 1600 bpi is the default.

Unless ALL is specified, saving a pack by specifying the keyword ENTIRE saves only those records in the volume that are actually used, plus the label information, the volume map, and the VTOC. For example, to save the volumes on units DSKA 1 and 2 onto disk packs on DSKA 3 and 4, type:

```
SAVE DISK DSKA 3 4
DSKA 1 ENTIRE RPV
DSKA 18. ENTIRE ROOT2
END
```

(This example assumes that the physical volume name of the pack on unit 1 is RPV and the name of the pack on unit 2 is ROOT2.) The indented lines are request lines read by the SAVE command; each input request is acknowledged by a message of the form:

```
DSKA 1 (M400 A 30) 0. TO 19269.
```

but the SAVE does not actually start until all input has been typed and the END request given. When SAVE or RESTOR begins a new volume of the SAVE medium, it types out a message describing what volume address from the Multics data is on the SAVE medium, in the form:

```
RPV (DSKA 1) 12349. ON TAPE 3
```

As each volume of the SAVE medium (tape reel or disk pack) is finished, SAVE or RESTOR types out the error count. No more than one request per volume may be given.

SAVE and RESTOR do some checking of the volume labels of disk packs. If an attempt is made to RESTOR the contents of one volume onto a volume with a valid label but a different volume name, RESTOR prints:

```
PHYSICAL VOLUME NAME DOES NOT MATCH REQUEST
REQUEST IS FOR PUBLIC, BUT VOLUME NAME IS RPV
```

and refuses to proceed. Similarly, if an attempt is made to use a storage system volume as a SAVE pack, the message:

```
DSKA 4 IS MULTICS STORAGE SYSTEM VOLUME ROOT3
```

is printed and no SAVE or RESTOR is done. Instead of typing END, you should type the QUIT request to return to BOS command level and then check the assignment of packs to disk units very carefully. In very rare cases, a pack may have an incorrect label left over from a previous use, which causes one of the above messages and prevents SAVE and RESTOR from operating. The TEST command should be used to clear the label in this case.

To save the dump partition on tape, type:

```
SAVE  ALL
PART DUMP ENTIRE RPV
END
```

The SAVE to disk feature of the SAVE and RESTOR commands uses disk instead of tape for performing the same functions. As each disk pack becomes full, it is stopped to allow you to mount another pack, and the next drive in the list is used to continue the SAVE operation.

NOTES ON SAVE COPY

The SAVE COPY feature can be used to copy packs in order to provide a backup copy of the disk contents at a particular moment. It can also be used to copy as much as possible from a defective disk pack onto another pack. Each Multics record copied is placed in exactly the same position on the target pack. There is no copy feature in the RESTOR command. The restoration function is effected by mounting the copy in place of the original pack or by reissuing the SAVE COPY command with the specifications of the original pack and the copy interchanged. The ALL argument
* must be given if the volumes to be copied have not been salvaged or shut down.

To use the SAVE COPY feature, type:

```
SAVE COPY DSKA 8 ALL
DSKB 3 ENTIRE ROOT2
END
```

to copy the contents of DSKB 3 to a pack mounted on DSKA 8. The ALL argument causes partitions to be saved. The optional BRIEF argument may be used to suppress all recoverable disk error messages. This is extremely useful when attempting to save or copy a volume containing many bad records.

*

Name: TAPED

SYNTAX AS A COMMAND

```
TAPED n
```

FUNCTION

dumps, in octal, the contents of the tape mounted on a specified drive.

ARGUMENTS

n
is the decimal number of the drive on which the tape to be dumped is mounted.

NOTES

The tape must have been written in binary mode. TAPED prints EOF and ERROR whenever they occur. TAPED may be stopped by pressing the REQUEST button.

Name: TEST

SYNTAX AS A COMMAND

```
TEST {volume_designator extent} {keywords}
```

FUNCTION

tests disks for errors. The portion of the disks to be tested can be specified as with SAVE and RESTOR. TEST may be stopped at any time by pressing the REQUEST button on the bootload console.

ARGUMENTS

volume_designator
indicates a particular volume or part of a volume. It may be specified in one of the following ways:

subsystem N
where N is a drive number. If N ends in a decimal point, it is considered a decimal number; otherwise it is considered an octal number.

PART partition_name

ROOT

extent
is a range of addresses on a single volume. It may be specified in one of the following forms:

```
device_position TO number
device_position FROM number
device_position ONLY
ENTIRE
```

where:

device_position
specifies a starting or ending device address; it may have one of these forms:

```
RECORD record_number
SECTOR sector_number
record_number
```

number
is either a record number or a sector number, depending on the units of the device_position argument.

keywords
may be chosen from:

CHECK
the records are read and the data is also verified. (See PATERN, below.)

WRITE
the records are written. If no word pattern is specified, zeros are written.

REWRIT
The records are read. If any errors are detected, those records are rewritten, possibly correcting the errors.

PATERN n1 {n2} {n3} {n4}
The data that is to be checked or written is changed from all zeros. Depending upon the number of 12-digit octal numbers given, one of the following sequences is used to fill each record:

```
n1 n1 n1 n1...

n1 n2 n1 n2...

n1 n2 n3 n1 n1 n2 n3 n1...

n1 n2 n3 n4 n1 n2 n3 n4...
```

MARK
This keyword, if chosen, must be the last argument on the command line. The first data word of each record to be checked or written contains the device number and device address. The remaining words of the record are zero.

BRIEF
Error messages for recoverable disk errors are suppressed.

NOTES

If neither CHECK, WRITE, nor REWRIT is included in the command, then the records are read only.

Any errors reported by the device are printed.

For more information on formatting packs with an alternate partition, see the *Multics System Maintenance Procedures* manual, Order No. AM81.

EXAMPLES

The following example shows how to correct the check character on a 451 disk pack. A check character error is indicative of data damage and can be corrected by writing over the data that is bad.

```
!  test dska 16. entire brief        reads all records on the pack

   error at d451 3423                error msg: this record is bad

!  test dska 16. 3423 only           reads only the bad record error

!  test dska 16. 3423 only rewrite   reads record and writes it out
```

Name: TIME

SYNTAX AS A COMMAND

TIME

FUNCTION

reads the calendar clock of the bootload SCU and does the time conversion necessary to set the correct time in an SCU. A null input line causes the TIME command to return to the BOS command processor.

LIST OF REQUESTS

After the BOS TIME command has been issued the following input requests are accepted:

mm dd yy hh mm ss
: are the month, day, year, hour, minute, and second. Hour, minute, and second may be omitted.

S
: sets the clock in the low-order memory. (Clocks in Series 60 (Level 66)/6000 SCs must be loaded manually. See the *Multics System Maintenance Procedures* manual, Order No. AM81, for details.)

R
: reads the calendar clock in the low-order memory.

Following the input of data, the time is printed in all three forms as shown:

```
NNNNN,NNNNNN  NNNNNN   TTTTTT  TTTTTT   MM/DD/YY  HH:MM:SS.S
      1                        2                    3
```

1. A 52-bit octal integer as read from the clock.

2. 36-bit integer formed by shifting the 52 bits right 16 places. This is the number to place in the switches to set the clock manually.

3. The date and time as month/day/year hour:minute:second.

To exit from the TIME command, you must press the EOM button.

NOTES

Although the time typed to and printed by the TIME command is local time, the time is set inside the calendar clock in Greenwich mean time (GMT). The TIME command uses the difference between GMT and local time as supplied on the clok configuration card when converting between calendar clock setting and date and time. (See the clok card description in the *Multics System Maintenance Procedures Manual*, Order No. AM81.) For this reason, a configuration deck with a clok card must have been loaded before the command is issued.

An incorrect clock setting can result in a system crash and possible loss of data.

EXAMPLES

```
! TIME
! 11 11 77 12 04 00
  104740,526512 460000 423602 532452 11/11/77 12:4:0
! S (the time is set when EOM is pressed after S is typed)
! R
  104740,526523 570660 423602 532516 11/11/77 12:4:2
  EOM (must push this button to exit from the TIME command)
```

Name: TST3BT

SYNTAX AS A COMMAND

```
TST3BT {FNP_tag {drive_no}}
```

FUNCTION

tests FNPs with the CSD supplied FNP test programs.

ARGUMENTS

FNP_tag
: is the tag of the FNP to be tested. If no tag is given, the default is "a".

drive_no
: is the number of the drive on which the CSD test program tape is mounted. If no number is given, the default is "3".

NOTES

This command functions identically to the CSD offline test program by the same name. For information on this test program, refer to the CSD document *Series 6000 Test and Diagnostics Manual*, Order No. 58008382.

Name: TSTCHN

SYNTAX AS A COMMAND

```
TSTCHN channo device {type} {waittime}
```

FUNCTION

tests an IOM channel by reading lines, consisting of a sequence of requests, from the current input source. This command is intended for use by the programming staff.

ARGUMENTS

channo
: specifies the number of the IOM channel to use.

device
: is the number that selects a unit for tape or disk controllers.

type
: is either PSIA (Peripheral Subsystem Interface Adapter) or CPCHAN (Common Peripheral Channel); PSIA is the default.

waittime
: specifies how long to wait for status; the default is about three seconds.

LIST OF REQUESTS

All of the requests on an input line are used to form a DCW list that is then connected to. Status is reported online. Requests may be selected from the following list.

Q
: returns to command level. Pressing the REQUEST key performs the same function.

N op rep
: issues a nondata transfer command op with repetition count of rep. A count of zero implies 64.

W op tal
: issues a unit record data transfer command operation op with data DCW tally of tal, from a 2048-word buffer.

R op tal
: same as W, except that the first four words of the buffer are typed after the operation.

M
: turns on the marker bit in the previous instruction DCW.

C
turns on the character tally bit in the previous data DCW.

D i word
sets word i of the buffer to word. Has no effect on the DCW list.

P i
prints word i of the buffer.

Name: TTY

SYNTAX AS A COMMAND

```
TTY
```

FUNCTION

causes input to be accepted by the bootload console.

NOTES

Reading of input reverts to the previous source when a blank line is typed. For more information on BOS input sources, see the *Multics System Maintenance Procedures* manual, Order No. AM81.

Name: WARM

SYNTAX AS A COMMAND

```
WARM model channel drive
```

FUNCTION

is used whenever BOS has been previously loaded onto the specified disk, and that copy of BOS is desired. (It is one of the BOS loader control commands.) The firmware loader and the BOS loader are read from the BOS tape into main memory for immediate use. The BOS loader then reads the other parts of BOS from the disk. None of the contents of the BOS tape are loaded onto the disk. The LOADDM command can be issued later to load the command programs from the tape. (See the LOADDM command.)

ARGUMENTS

model
is the model number, with decimal point, of the disk drive, e.g., "400.".

channel
is the number of the IOM channel connected to the disk subsystem. This number must be less than 40 (octal) or 32 (decimal).

drive
is the number of the drive on which the disk is mounted. This number must be decimal.

NOTES

Refer to the description of the prph config card in the *Multics System Maintenance Procedures* manual, Order No. AM81, for model numbers.

When a WARM control command is used to load BOS, a message of the following form is printed:

```
BOS PARTITION AT frec FOR nrec
```

where frec is the first record of the BOS partition and nrec is the number of records. If the volume label cannot be interpreted following a WARM command, one of the following messages will appear:

```
CANNOT READ VOLUME LABEL
VOLUME DOES NOT HAVE A MULTICS LABEL
BOS PARTITION NOT DEFINED IN VOLUME LABEL
```

Name: WRITE

SYNTAX AS A COMMAND

```
WRITE text
```

or

```
WRITE ALERT
```

FUNCTION

sends a message to the bootload console or turns on an audible alarm at the bootload console to alert the operator. It is useful in a BOS runcom.

ARGUMENTS

text
: is the message to be sent. Everything on the command line following the WRITE command is typed out on the bootload console.

APPENDIX A

CONFIGURATION DECK DESCRIPTION

This appendix contains output from the print_configuration_deck command. The output represents the contents of the configuration deck used to create the system from which specified metering samples were taken. (The metering samples that reflect this configuration are so noted in the text for the individual metering commands.)

```
clok  7  mst
iom  a  0  nsa  on
iom  b  1  nsa  on
prph  dska  a  20.  2  451.  16.
chnl  dska  a  26.  2  b  24.  2  b  22.  2
prph  dskb  b  20.  2  0  16.  451.  16.
chnl  dskb  b  26.  2  a  24.  2  a  22.  2
prph  dskc  a  28.  2  501.  16.
chnl  dskc  b  30.  2
prph  dskd  b  28.  2  0  16.  501.  16.
chnl  dskd  a  30.  2
prph  dske  b  32.  2  451.  8.
chnl  dske  b  34.  2
prph  dskf  a  36.  2  501.  16.
chnl  dskf  b  38.  2
prph  dskg  b  36.  2  0  16.  501.  16.
chnl  dskg  a  38.  2
prph  tapa  a  16.  1  630.  2  610.  6  630.  2
prph  opca  a  14.  6601.  80.  on
prph  prtd  a  8.  1600.  600.  160.
prph  prta  a  9.  1200.  600.  136.
prph  prtb  a  10.  1600.  600.  136.
prph  rdra  a  11.  500.
prph  puna  a  12.  300.
prph  prtc  b  8.  1201.  600.  136.
prph  prte  b  9.  1600.  600.  160.
prph  punb  b  10.  300.
prph  rdrb  b  11.  500.
dbmj  64.  700.  400.  150.  60.  25.
parm  chwm  hcpt  ttyb  55296.
part  bos  dska  16.
part  dump  dska  16.
part  log  dska  16.
root  dska  16.  dskb  25.  dskb  23.  dskb  24.  dska  7  dska  11.
schd  400000  4  10  100  2  20.
sst  3500.  1500.  750.  365.
tbls  scav  138.  str  64.
tcd  200.  506.
prph  fnpa  b  18.  on
prph  fnpb  a  18.  on
prph  fnpc  a  13.  on
prph  fnpd  a  17.  on
prph  fnpe  b  19.  on
prph  fnpf  b  15.  on
mpc  mspa  451.  a  20.  4  a  24.  4
mpc  mspb  451.  b  20.  4  b  24.  4
mpc  mspc  607.  a  28.  4
mpc  mspd  607.  b  28.  4
mpc  mspe  451.  b  32.  4
mpc  mspf  607.  a  36.  4
mpc  mspg  607.  b  36.  4
mpc  mtpa  610.  a  16.  1  b  16.  1
mpc  urpa  600.  a  8.  5
mpc  urpb  8004.  b  8.  2
mpc  urpc  8002.  b  10.  2
```

```
intk  warm  3  star
mem   b  2048.  on
mem   a  1024.  on
mem   c  4096.  on
mem   d  2048.  on
cpu   a  3  on  168   80.   2.
cpu   b  4  on  168   80.   2.
cpu   c  5  on  168   80.   2.
cpu   d  6  on  dps8   70.   8.
cpu   e  7  on  dps8   70.   8.
```

INDEX

A

B

D

G

H

M

P

Q

R

S

T

W

X

Z

HONEYWELL INFORMATION SYSTEMS
Technical Publications Remarks Form

TITLE: MULTICS ADMINISTRATION, MAINTENANCE, AND OPERATIONS COMMANDS ADDENDUM A

ORDER NO. GB64-00A

DATED NOVEMBER 1986

ERRORS IN PUBLICATION

SUGGESTIONS FOR IMPROVEMENT TO PUBLICATION

Your comments will be investigated by appropriate technical personnel and action will be taken as required. Receipt of all forms will be acknowledged; however, if you require a detailed reply, check here. ☐

PLEASE FILL IN COMPLETE ADDRESS BELOW.

FROM: NAME _____ DATE _____

TITLE _____

COMPANY _____

ADDRESS _____

_____

CUT ALONG LINE

PLEASE FOLD AND TAPE–
NOTE: U.S. Postal Service will not deliver stapled forms

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 39531 WALTHAM, MA 02154

POSTAGE WILL BE PAID BY ADDRESSEE

HONEYWELL INFORMATION SYSTEMS
200 SMITH STREET
WALTHAM, MA 02154

ATTN: PUBLICATIONS, MS486

Honeywell

Together, we can find the answers.

Honeywell

Honeywell Information Systems
U.S.A.: 200 Smith St., MS 486, Waltham, MA 02154
Canada: 155 Gordon Baker Rd., Willowdale, ON M2H 3N7
U.K.: Great West Rd., Brentford, Middlesex TW8 9DH **Italy**: 32 Via Pirelli, 20124 Milano
Mexico: Avenida Nuevo Leon 250, Mexico 11, D.F. **Japan**: 2-2 Kanda Jimbo-cho, Chiyoda-ku, Tokyo
Australia: 124 Walker St., North Sydney, N.S.W. 2060 **S.E. Asia**: Mandarin Plaza, Tsimshatsui East, H.K.

43126, 7.5C685, Printed in U.S.A.

GB64-00