

**UNISYS**

System 80

OS/3

Interactive Services

Commands and Facilities

**Programming and Operations**

**Quick-Reference Guide**

Copyright © 1988 Unisys Corporation

All rights reserved.

Unisys is a registered trademark of Unisys Corporation.

OS/3 Release 13.0

January 1990

Priced Item

Printed in U S America  
UP9973 Rev.1 - Update B

**PAGE STATUS SUMMARY**  
**ISSUE: Update B - UP-9973 Rev. 1**

Part/Section	Page Number	Update Level	Part/Section	Page Number	Update Level
Cover		B	7	Tab Breaker 1 2 thru 5	A B Orig.
Title Page/Disclaimer		B	8	Tab Breaker 1 thru 31	A Orig.
PSS	iii	B	9	Tab Breaker 1 thru 3	A Orig.
About This Guide	Tab Breaker v thru vii viii ix, x	A Orig. B Orig.	10	Tab Breaker 1 thru 7	A Orig.
Contents	xi xii xiii, xiv	A B A	11	Tab Breaker 1, 2	A B
1	Tab Breaker 1, 2	A Orig.	12	Tab Breaker 1 thru 4	A Orig.
2	Tab Breaker 1, 2	A B	Index	Tab Breaker 1, 2 3 thru 5 6 7 8 9 thru 11	A B A B A B A
3	Tab Breaker 1 thru 9	A Orig.	User Comments Form		
4	Tab Breaker 1 thru 3	A B	Back Cover		Orig.
5	Tab Breaker 1 thru 8	A Orig.			
6	Tab Breaker 1, 2 3 4 thru 11 12, 13 14 thru 16 17 18 19	A Orig. B Orig. B Orig. B Orig. B			

Technical changes are denoted by a change bar in the margin.

**UNISYS**

System 80

OS/3

Interactive Services  
Commands and Facilities

**Programming and Operations  
Quick-Reference Guide**

Copyright © 1988 Unisys Corporation  
All rights reserved.  
Unisys is a trademark of Unisys Corporation.

OS/3 Release 12.0

October 1988

Priced Item

Printed in U S America  
UP-9973 Rev. 1

**PAGE STATUS SUMMARY**  
**ISSUE: Update A - UP-9973 Rev. 1**  
**RELEASE LEVEL: 12.0 Forward**

Part/Section	Page Number	Update Level	Part/Section	Page Number	Update Level
Cover		Orig.	7	Tab Breaker 1 thru 5	A* Orig.
Title Page/Disclaimer		Orig.	8	Tab Breaker 1 thru 31	A* Orig.
PSS	iii	A	9	Tab Breaker 1 thru 3	A* Orig.
About This Guide	Tab Breaker v thru x	A* Orig.	10	Tab Breaker 1 thru 7	A* Orig.
Contents	xi thru xiv	A	11	Tab Breaker 1 thru 3	A* Orig.
1	Tab Breaker 1, 2	A* Orig.	12	Tab Breaker 1 thru 4	A* Orig.
2	Tab Breaker 1, 2	A* Orig.	Index	Tab Breaker 1 thru 11	A* A*
3	Tab Breaker 1 thru 9	A* Orig.	User Comment Form		
4	Tab Breaker 1 thru 3	A* Orig.	Back Cover		Orig.
5	Tab Breaker 1 thru 8	A* Orig.			
6	Tab Breaker 1 thru 19	A* Orig.			

\* New page

**PAGE STATUS SUMMARY**  
**ISSUE: UP-9973 Rev. 1**  
**RELEASE LEVEL: 12.0 Forward**

Part/Section	Page Number	Update Level	Part/Section	Page Number	Update Level
Cover			11	1 thru 3	
Title Page/Disclaimer			12	1 thru 4	
PSS	iii		User Comments Form		
About This Guide	v thru x		Back Cover		
Contents	xi thru xiv				
1	1, 2				
2	1, 2				
3	1 thru 9				
4	1 thru 3				
5	1 thru 8				
6	1 thru 19				
7	1 thru 5				
8	1 thru 31				
9	1 thru 3				
10	1 thru 7				

# About This Guide

## Purpose

This summary helps you use the interactive commands and facilities available on OS/3.

## Scope

Included in this guide are:

- All the commands you may enter from a workstation
- Editor commands
- Information on:
  - Entering information from a workstation
  - Command conventions used in this manual
  - LOGON and ICAM terminal sign on procedures
  - Initiating the various interactive facilities (such as job control dialog)
  - Function keys
  - Use of various types of terminals as workstations

**Section 5. Controlling Spooling**

Describes the commands that allow you to control the processes of input and output spooling for your job.

**Section 6. Utility Commands**

Describes the commands that permit you to perform utility functions, such as copying files, scratching files, and obtaining status information on various aspects of your OS/3 interactive processing system.

**Section 7. Interactive Facilities**

Describes the interactive facilities you can use.

**Section 8. General Editor Commands**

Describes the commands used by the Unisys OS/3 General Editor.

**Section 9. RPG II Editor and COBOL Editor**

Explains how to activate and use the RPG II and COBOL editors.

**Section 10. DDP Commands**

Briefly explains the special commands that perform DDP functions.

**Section 11. Interactive Terminals**

Describes the different procedures used to access interactive services through the four different types of terminals.

**Section 12. Function Key Summary**

Provides an overview of the function keys used by interactive serices and the various interactive facilities.

## Notation Conventions

The conventions used to illustrate the commands presented in this guide are as follows:

- If part of a command or parameter is underscored, it means that only the underscored portion has to be keyed in. When a command or parameter is not underscored, you must enter the entire command or parameter. In the following example, ER of the ERASE command is underscored. You must enter at least ER for the command to work, but you can enter ERA, ERAS, or ERASE.

ERASE

where ER is the minimum keyin required to successfully use the ERASE command.

- Parameters printed in lowercase letters designate undefined variables:

,VSN=volume

where volume is the serial number of the volume you're working with.

- Optional parameters are enclosed in brackets:

[,WRPASS= password]

- Alternate choices for a parameter are enclosed in braces:

,DEVICE= { nnn  
          { RDR }

# Contents

About This Guide .....	v
<b>Section 1. Entering Information on a Workstation</b>	
Using the Correct Mode .....	1-1
Using More than One Line to Enter a Command, Response, or Message .....	1-1
What You Can Control .....	1-2
<b>Section 2. LOGON Procedure</b>	
Standard Terminal Dialog for ICAM Terminals .....	2-2
<b>Section 3. Running Jobs and Changing Job Scheduling</b>	
FILE Command .....	3-1
RUN Command .....	3-2
RV Command .....	3-3
SI Command .....	3-4
SC Command .....	3-5
EXECUTE Command .....	3-6
ULD Command .....	3-6
DLOAD Command .....	3-6
UNLOAD Command .....	3-7
BEGIN Command .....	3-7
HOLD Command .....	3-8
CHANGE Command .....	3-8
DELETE Command .....	3-9
DISPLAY JBQ Command .....	3-9

FSTATUS Command .....	6-11
HELP Command .....	6-11
MENU Command .....	6-11
PRINT Command .....	6-12
PUNCH Command .....	6-13
REBUILD Command .....	6-15
RECALL Command .....	6-15
RECOVER Command .....	6-16
REMARK Command .....	6-16
REMOVE Command .....	6-17
RESUME Command .....	6-17
SCREEN Command .....	6-18
STATUS Command .....	6-19
TELL Command .....	6-19
VTOC Command .....	6-19

**Section 7. Interactive Facilities**

Interactive Data Utilities .....	7-1
Job Control Dialog .....	7-2
Screen Format Generator .....	7-2
Using Screen Formats in Your Program .....	7-2
Menu Generator .....	7-3
Using Menus in Your Program .....	7-3
Interactive DUMP/RESTORE Hardware Utility .....	7-4
BASIC .....	7-5
ESCORT .....	7-5

**Section 8. General Editor Commands**

EDT Commands .....	8-1
EDT Variable Commands .....	8-17
EDT Procedure File Commands .....	8-19
EDT Directives .....	8-22
EDT Screen Commands .....	8-27
EFP Commands .....	8-30

# Section 1

## Entering Information on a Workstation

### Using the Correct Mode

To enter commands and respond to messages at the System 80 workstation, you must be in SYSTEM mode. To get into SYSTEM mode, press the FUNCTION key, hold it down, and press the SYS MODE key.

To enter data and output to user-created and system programs (including facilities like EDT) you must be in WORKSTATION mode. To get into WORKSTATION mode, press the FUNCTION key, hold it down, and press the WS MODE key.

For instructions on how to use a terminal other than the System 80 workstation, refer to the discussion entitled INTERACTIVE TERMINALS in this manual.

### Using More than One Line to Enter a Command, Response, or Message

If you need more than one line (80 characters) of a workstation screen to enter a command, response, or message, place a dash (-) as the last character on the line. The dash is a continuation character. It prompts the system to ask you, in a message, for further input. Use as many lines of the screen as you need; just place a dash at the end of every line except the last one.

## Section 2

# LOGON Procedure

You must begin every workstation session with the LOGON command. To log on, press XMIT and fill in the LOGON menu displayed to you. You may also log on by entering SYSTEM mode and entering the LOGON command as follows:

```
LOGON user id[,acct][,password][,exec-pro] [BULLETIN= { NO }  
{ YES } ] [LOG= { NO }  
{ YES } ]  
[ ,NEWPASS=new-password]
```

Once you are logged on, Interactive Services automatically terminates if you don't perform any activity for 15 minutes. If this occurs you must enter another LOGON command.

### Explanation

user id

Is 1-6 alphanumeric characters.

acct

Is 1-4 alphanumeric characters.

password

Is 1-8 alphanumeric characters.

exec-pro

Is 1-8 alphanumeric characters.

## Section 3

# Running Jobs and Changing Job Scheduling

The following commands are used to run jobs from the workstation and change the scheduling for execution of those jobs.

### FILE Command

The FILE command files jobs and job control procedures (jprocs) into the permanent job control stream library file (\$Y\$JCS) or into an alternate SAT library file.

$$\text{FILE } \left\{ \begin{array}{l} ([did], \text{label}) \\ (RDR, \text{label}) \end{array} \right\} \Delta \left[ \begin{array}{l} : \text{alt-filename} \\ : \left( \text{alt-filename}, \left\{ \begin{array}{l} \text{RES} \\ \text{RUN} \\ \text{vsn} \end{array} \right\} \right) \\ : \left( \text{alt-filename}, \left[ \begin{array}{l} \text{RES} \\ \text{RUN} \\ \text{vsn} \end{array} \right], \text{write-pass} \right) \end{array} \right]$$

## RV Command

The RV command enables you to execute user jobs from the workstation. The command causes the job control stream associated with your job to be read, expanded, and scheduled for execution. The RV command is used when there is no input device needed.

RVA jobname [(new-name)]

{	:alt-filename	}
{	: (alt-filename, { RES RUN vsn })	}
{	: (alt-filename, { RES RUN vsn }, read-pass)	}

{	PRE HIGH NOR LOW	}	[,time]	+	{	d1 . . . d9	}	[,key-1=val-1,...,key-n=val-n]
---	---------------------------	---	---------	---	---	-------------------------	---	--------------------------------

## SC Command

The SC command enables you to run jobs that you have previously saved in their expanded state. The SC command is used when the job control stream does not require an input device to replace embedded data.

SCΔ jobname [(new-name)]

$$\left[ \begin{array}{l} \text{:alt-filename} \\ \text{: (alt-filename, (RES))} \\ \quad \left. \begin{array}{l} \text{RUN} \\ \text{vsn} \end{array} \right\} \\ \text{: (alt-filename, (RES),read-pass)} \\ \quad \left. \begin{array}{l} \text{RUN} \\ \text{vsn} \end{array} \right\} \end{array} \right]$$

$$\left[ \begin{array}{l} \text{PRE} \\ \text{HIGH} \\ \text{NOR} \\ \text{LOW} \end{array} \right] [,time] + \left[ \begin{array}{l} \text{d1} \\ \cdot \\ \cdot \\ \cdot \\ \text{d9} \end{array} \right]$$

## UNLOAD Command

The UNLOAD command frees devices that were allocated to a downline-loaded program.

UNLOAD

## BEGIN Command

The BEGIN command institutes the rescheduling of jobs deferred by the HOLD command. You may reschedule individual jobs or all jobs in the designated queue that you are allowed to control.

### Rescheduling All Jobs or All Jobs on a Particular Job Queue

BEGINΔJBQ { A  
H  
N  
P  
L }

### Rescheduling an Individual Job

BEGINΔjobname

## DELETE Command

The DELETE command permits you to remove jobs that you are allowed to control from scheduling queues. Jobs deleted will not be executed. You may remove single jobs, all jobs on a particular queue, or all jobs on all queues.

### Deleting All Jobs from One or All Job Queues

DELETEΔJBQ,  $\left( \begin{array}{c} \text{A} \\ \text{H} \\ \text{N} \\ \text{P} \\ \text{L} \end{array} \right)$  [, LOG]

### Deleting a Specific Job

DELETEΔjobname[, LOG]

## DISPLAY JBQ Command

The DISPLAY JBQ command permits you to display, on the workstation screen, the contents (names of jobs) of each scheduling queue for the jobs you can control.

### Format

DISPLAYΔJBQ,  $\left( \begin{array}{c} \text{A} \\ \text{H} \\ \text{N} \\ \text{P} \\ \text{L} \end{array} \right)$

## Section 4

# Controlling Jobs

The following commands permit your workstation to act as a miniature system console to control those jobs initiated or run under your user-id. You may *only* control jobs initiated or run under your user-id, unless you have been given global control privileges by the system administrator.

### CANCEL Command

The CANCEL command enables you to immediately halt processing of a job; the currently executing job step is not completed. You may specify whether or not you want a dump taken when you enter the CANCEL command.

`CANCEL` *jobname* [ , { D } ]

### CJ Command

The CJ command enables you to soft cancel a job. This command is similar to the CANCEL command except that it only takes effect when the following conditions are met:

- The job is not in shared code
- The job is in its own key

`CJ` *jobname* [ , { D } ]

## GO Command

The GO command reactivates jobs suspended by the PAUSE command or by certain statements within the job control stream associated with the suspended job.

`GOΔjobname`

## STOP Command

The STOP command enables you to terminate a job at the completion of the currently executing (or specified) job step.

`STOPAjobname[, job-step-no]`

## Section 5

# Controlling Spooling

The following commands allow you to control the processes of input and output spooling for your job. These commands affect all jobs, not just those initiated under your user-id, if you have been given global control privileges by the system administrator. For your convenience, the spool file directories are reproduced here:

Directory	File Function
ALL	Makes all directories accessible to the command in which it is specified
LOG	Indicates that the file referenced is a job log file
PUNCH	Indicates that the file is to be output to either a card punch or a diskette
PRINT	Indicates that the file is to be output to a printer
RDR	Indicates that the file was originally from a card reader or diskette

## HOLD SPL Command

The HOLD SPL command permits you to place files in a hold state, making them unavailable for processing.

HOLDASPL,  $\left\{ \begin{array}{l} \text{ALL} \\ \text{PRINT} \\ \text{PUNCH} \\ \text{LOG} \\ \text{RDR} \end{array} \right\}$  [,ACCT=acctno][,BNUMB=binary jobno]

[,CART=cartridge-name]  $\left[ \text{DEV} = \left\{ \begin{array}{l} 770 \\ 776 \\ 789 \end{array} \right\} \right]$

[,FILE=filename][,FORM=formname][,JOB=jobname][,STEP=stepno]

## DELETE SPL Command

The DELETE SPL command enables you to delete files from spool queues. You may delete completed files only; files being created may not be deleted.

```
DELETEASPL, ( ALL
              LOG
              PRINT
              PUNCH
              RDR ) [ ,ACCT=acctno][ ,BNUMB=binary jobno]
```

```
[ ,CART=cartridge-name] [ ,DEV= ( 770
                                776
                                789 ) ]
```

```
[ ,FILE=filename][ ,FORM=formname][ ,JOB=jobname][ ,STEP=stepno]
```

**Note:** *Global control users will delete every qualifying spool file in the system, not just the spool files initiated under their user-ids.*

## BRKPT Command

The BRKPT (breakpoint) command closes one or more spool files and makes them available to an output writer for printing or punching. The remainder of the file created after the BRKPT command is issued is placed on another file.

$$\text{BRKPT} \Delta \left\{ \begin{array}{l} \text{P} \\ \text{I} \end{array} \right\}, \left\{ \begin{array}{l} \text{PR} \\ \text{PU} \end{array} \right\} [ , \text{ACCT}=\text{acctno} ] [ , \text{CART}=\text{cartridge-name} ]$$

$$\left[ \begin{array}{l} , \text{DEV} = \left\{ \begin{array}{l} 770 \\ 776 \\ 789 \end{array} \right\} \\ \left[ , \text{FILE}=\text{filename} \right] \left[ , \text{FORM}=\text{formname} \right] , \text{JOB}=\text{jobname} \left[ , \text{HOLD} \right] \end{array} \right]$$

## PR/PU Command

The PR/PU command enables you to manually load an output writer to print (PR) or punch (PU) spooled files associated with your job.

$$\left\{ \begin{array}{l} \text{PR} \\ \text{PU} \end{array} \right\} \Delta [ \text{function-code} ] [ , \text{ACCT}=\text{acctno} ] [ , \text{BNUMB}=\text{binary jobno} ]$$

$$[ , \text{CART}=\text{cartridge-name} ] [ , \text{FILE}=\text{filename} ] \\ [ , \text{FORM}=\text{formname} ] [ , \text{JOB}=\text{jobname} ]$$

## Section 6

# Utility Commands

The following commands permit you to perform utility functions, such as copying files, allocating files, scratching files, and obtaining status information on various aspects of your OS/3 interactive processing system. You may *only* control jobs initiated or running under your user-id, unless you have been given global control privileges by the system administrator.

### ALLOCATE Command

The ALLOCATE command permits you to allocate space for files interactively, without the need for job control statements. You may allocate SAT and MIRAM files using this command.

```
ALLOCATE {ST} ,FILENAME= { filename } [ ,RDPASS=password ] [ ,WRPASS=password ]
          {MI}           { 'filename' }
                       { "filename" }

, VSN=volume [ ,CONTIG= { YES } ] [ ,INC= { n } ] ,SIZE=n
              { NO }
```

**Note:** When allocating space on diskettes:

- If the diskette is recorded in data-set-label (DSL) mode, give the SIZE= and INC= parameters in blocks.
- If the diskette is recorded in format-label (FL) mode, give the SIZE= and INC= parameters in cylinders.

## COPY Command

The COPY command enables you to copy files and, in the process of copying, to alter the format of a file.

The COPY command sets the date/time stamp on source and procedure modules to the *current* date and time. It retains the *original* date and time on load and object modules. To retain the original date and time on source and procedure modules, you must use the librarian to copy the modules. See the *System Service Programs (SSP) Operating Guide* (UP-8841).

To copy one type of file to another type using the COPY command, use the appropriate input and output parameter strings. The input parameter string is everything preceding the word TO; the output parameter string is everything following it. Input and output parameter strings for each file type are shown in the following formats.

### Copying SAT or MIRAM Library Modules

```

COPYMODULE=modulename [ ,TYPE= { module-type } ] ,FILENAME= { filename
                        { '$' } } { filename'
                        { '$' } } { filename"

[ ,RDPASS=password ], VSN=volume Δ TO MODULE=modulename [ ,TYPE= { module-type }
                        { '$' } ]

,FILENAME= { filename } [ ,RCSZ=n ] [ ,WRPASS=password ], VSN=volume
            { filename'
            { '$' } } { filename"

[ ,CONTIG= { YES } ] [ ,INC= { n } ] [ ,SIZE=n ] [ ,SAT= YES ] Δ [ NUMBER ] [ ,HEX ] \
[ ,WAIT ]

```

**Copying Spool Files**

COPYΔ[JOB=jobname] [ ,HOLD= { L  
N  
Y } ] [ ,FILENAME= { filename  
'filename'  
"filename" } ] [ ,ACCT=acct ]

[ ,QUEUE= { LOG  
PRINT  
PUNCH  
RDR } ] [ ,ALL= { YES  
NO } ]

[ ,SKIP= { n  
0 } ] ΔTOΔ[JOB=jobname] [ ,HOLD= { N  
Y } ] [ ,FILENAME= { filename  
'filename'  
"filename" } ]

[ ,QUEUE= { PRINT  
PUNCH  
RDR } ]

Δ[NUMBER][ ,HEX][ ,WAIT]

### Copying Unit Record Files

$\text{COPY}\Delta\text{DEVICE} = \left\{ \begin{array}{l} \text{did} \\ \text{DISKETTE} \\ \text{RDR} \end{array} \right\}, \text{FILENAME} = \left\{ \begin{array}{l} \text{filename} \\ \text{'filename'} \\ \text{"filename"} \end{array} \right\}$

$,\text{VSN}=\text{volume}\Delta\text{TO}\Delta\text{DEVICE} = \left\{ \begin{array}{l} \text{did} \\ \text{DISKETTE} \\ \text{PRINT} \\ \text{PUNCH} \end{array} \right\}$

$\left[ \text{,RCFM} = \left\{ \begin{array}{l} \text{FIX} \\ \text{VAR} \end{array} \right\} \right] \left[ \text{,RCSZ}=\text{n} \right], \text{FILENAME} = \left\{ \begin{array}{l} \text{filename} \\ \text{'filename'} \\ \text{"filename"} \end{array} \right\}$

$,\text{VSN}=\text{volume}\Delta[\text{NUMBER}][,\text{HEX}][,\text{WAIT}]$

## DEFKEY Command

The DEFKEY command lets you assign any interactive command to a function key or the MESSAGE WAITING key.

$\text{DEFKEY}\Delta \left\{ \begin{array}{l} \text{F\#nn} \\ \text{MW} \end{array} \right\}, \left\{ \begin{array}{l} \text{'command string'} \\ \text{"command string"} \end{array} \right\}$

## DEFKEY (Delete) Command

To free a function key or the MESSAGE WAITING key from a command assignment, you use the delete form of the DEFKEY command. The delete DEFKEY command is exactly like the DEFKEY command, minus the command string.

$\text{DEFKEY}\Delta \left\{ \begin{array}{l} \text{F\#nn} \\ \text{MW} \end{array} \right\}$

**To Run a Command Stream from a Spooled File**

ENTERA  $\left[ \begin{array}{l} \text{HOLD} = \left\{ \begin{array}{l} N \\ Y \end{array} \right\} \end{array} \right] , \text{FILENAME} = \left\{ \begin{array}{l} \text{filename} \\ 'filename' \\ "filename" \end{array} \right\} \text{QUEUE} = \text{RDR}$

**To Run a Command Stream from a DSL Diskette**

ENTERA  $\left[ \begin{array}{l} \text{FILENAME} = \left\{ \begin{array}{l} \text{filename} \\ 'filename' \\ "filename" \end{array} \right\} \end{array} \right] [ , \text{RDPASS} = \text{password} ] , \text{VSN} = \text{volume}$

$\left[ \begin{array}{l} , \text{DEVICE} = \left\{ \begin{array}{l} \text{did} \\ \text{DISKETTE} \end{array} \right\} \end{array} \right]$

**To Run a Command Stream from a Tape**

ENTERA  $\left[ \begin{array}{l} \text{FILENAME} = \left\{ \begin{array}{l} \text{filename} \\ 'filename' \\ "filename" \end{array} \right\} \end{array} \right] [ , \text{RDPASS} = \text{password} ] , \text{VSN} = \text{volume} \left[ \begin{array}{l} , \text{DEVICE} = \left\{ \begin{array}{l} \text{did} \\ \text{TAPE} \end{array} \right\} \end{array} \right]$

**To Run a Command Stream from a Card Reader**

ENTERA  $\text{DEVICE} = \left\{ \begin{array}{l} \text{did} \\ \text{RDR} \end{array} \right\}$

## FSTATUS Command

The FSTATUS command permits you to interactively obtain information about files and their contents. You may reference only library file modules.

$$\text{FSTATUS}\Delta[\text{MODULE}=\text{modulename}] \left[ \text{,TYPE}=\left\{ \begin{array}{l} \text{module-type} \\ \$ \end{array} \right\} \right] \text{,FILENAME}=\left\{ \begin{array}{l} \text{filename} \\ \text{'filename'} \\ \text{"filename"} \end{array} \right\}$$

$$[\text{,RDPASS}=\text{password}],\text{VSN}=\text{volume}\Delta[\text{LONG}]$$

## HELP Command

The HELP command permits you to obtain information about how to use various workstation commands and their parameters and how to respond to error messages from the system. Entering HELP and a command message number or keyword parameter produces a display explaining the command, message, or parameter. You may not enter an abbreviated form of the command or parameter.

$$\text{HELPA} \left\{ \begin{array}{l} \text{command} \\ \text{message-no} \\ \text{keyword-parameter} \end{array} \right\}$$

## MENU Command

The MENU command displays either a standard system menu or a user-created menu.

$$\text{MENU}\Delta \left[ \text{[menu-name]} \left[ \text{,} \left( \text{filename, vsn} \right) \right] \right]$$

$$\left[ \text{SYSTEM} \left[ \text{,} \left( \text{RES} \right) \right] \right]$$

### Printing a Unit Record File

```
PRINTADEVICE= { did
                DISKETTE
                RDR } ,FILENAME= { filename
                                   'filename'
                                   "filename" }
, VSN=volumeΔ[DIRECT][,NUMBER][,HEX][,WAIT][FORM=formname]
```

### Printing a Spool File

```
PRINTA[JOB=jobname] [ ,HOLD= { L
                               N
                               Y } ] [ ,FILENAME= { filename
                                                    'filename'
                                                    "filename" } ] [ ,ACCT=acct ]
, QUEUE= { LOG
           PRINT
           PUNCH
           RDR } [ ,ALL= YES
                  NO ] [ ,COPIES= n
                             ] [ ,SKIP= n
                                     ]
Δ[DIRECT][,NUMBER][,HEX][,WAIT]
```

## PUNCH Command

The PUNCH command allows you to make a punched-card copy of a library file or module, a spool file, a MIRAM file, a tape file, or a unit record file. PUNCH can be run as a background job to allow you to use the workstation for other functions while the file is being punched.

### Punching a Library Module

```
PUNCHMODULE=modulename [ ,TYPE= { module-type
                                   $ } ] [ ,FILENAME= { filename
                                                       'filename'
                                                       "filename" } ]
[ ,RDPASS=password ] ,VSN=volume [ ,COPIES= { n
                                               } ] Δ[DIRECT][,WAIT]
```

### Punching a Unit Record File

PUNCHΔDEVICE= { did  
DISKETTE } , FILENAME= { filename  
'filename'  
"filename" } , VSN=volumeΔ[DIRECT][,WAIT]

## REBUILD Command

The REBUILD command restores messages requiring a response or a GO command to your workstation screen.

REBUILD

## RECALL Command

The RECALL command lets you display all or part of your workstation or console log file at your workstation or console screen. You can view selected portions of the log file by specifying a particular time span. Or you can indicate the number of messages, prior to the most current one, that you'd like to see.

RECALL [ { LASTΔnn  
hh:mm:ss-hh:mm:ss } ] [,message-prefix]

**Note:** *The RECALL command will not span the time change at midnight. To display messages from 23:00 to 1:00, you must use the RECALL command twice.*

## REMOVE Command

The REMOVE command terminates a single command, a specific workstation or terminal user session, or all user sessions. A cancellation message is displayed on the user's screen after the session is terminated. If a user has global control privileges, he can remove any task-id or user-id in the system.

### Format

```
REMOVEA { task-id }  
        { user-id }  
        { ALL }
```

## RESUME Command

The RESUME command enables you to resume execution of a subsystem, such as the general editor or BASIC, that was suspended when the workstation entered SYSTEM mode.

RESUME

## STATUS Command

The STATUS command enables you to obtain information about various aspects of your system, including usage of terminals and workstations, numbers of jobs running on the system, system main storage resources in use and available, jobs running under your user-id, and which interactive commands and facilities are executing under your user-id. STATUS can also give you a listing of the disk, tape, and diskette volumes currently mounted on your system.

If a partial or complete user-id (one to six characters) is entered following the TERMINALS parameter, status is displayed for all terminals that are active with that partial or complete user-id.

```
STATUSA [ [ TERMINALS[,uid]
           [ RESOURCES
           [ JOBS
           [ FUNCTIONS
           [ VOLUMES
           [ LIMITS ] ] ] ] ] ] ]
```

## TELL Command

The TELL command enables you to send messages that do not require a response to the system operator or other workstation users.

```
TELLA [ [ user-id ] , ] 'text'
       [ ALL ] ] ]
```

## VTOC Command

The VTOC command enables you to produce a listing of the files present on a disk or diskette volume. You may list all the files on a volume, or only those whose file names begin with a prefix you specify. You can use command options to further restrict the listing. The MIRAM option lets you list only MIRAM and IRAM files. The SAT option lets you list only SAT files. You can use the FREE option to obtain a listing of each free extent available and its size. The LONG option provides additional date/time stamp information for each listed file.

```
VTOCA[ 'file-prefix' , ] VSN=volumeA [ FREE ]A [ SAT ]A [ MIRAM ]A [ LONG ]
```

# Section 7

## Interactive Facilities

### Interactive Data Utilities

To initialize the interactive data utilities, enter the following command:

```
RVAI@DATA[(new-name)]I,,MEM=nnnnn,I[ACT=act-no,] [DBG= { Y } ] [SPL= { HOLD } ]
```

#### Explanation

(new-name)

Permits the concurrent use of data utilities.

MEM=nnnnn

Specifies, in hexadecimal notation, the minimum main storage needed to run your data utilities job. Default is 8000<sub>16</sub> (32,767<sub>10</sub>).

ACT=act-no

Specifies a 1- to 4-character alphanumeric account number.

DBG= { Y }  
      { N }

Specifies that data utilities run in the debugging mode, used to provide documentation for reporting in a user communication form (UCF).

SPL= { HOLD }  
      { H }

Specifies that the spool files produced by data utilities will be held for printing at a later time. *If the SPL parameter is not specified, the spool file will not be held.*

{ nnn }  
{ i }

Specifies the number of formats to reside in main storage for use with a given file.

[, screen-format-1=alias-1...[, screen-format-12=alias-12]]

Specifies that a name other than the real name of the format is to be used. Maximum of 12 aliases permitted. More than 12 causes rejection of the job control stream.

## Menu Generator

Use the following command to initialize the menu generator:

MENUGEN

There are no parameters associated with this command.

## Using Menus in Your Program

Use the following job control statement to include menus in your program:

```
//[symbol]USE MENU [ { menu-file-lfd-1/menu-file-lfd-2 } [, initial-menu]
                    { $SY$FMT/menu-file-lfd-2
                      menu-file-lfd-1/$SY$FMT
                      menu-file-lfd-1
                      $SY$FMT } ]
```

[ { nnn } [, menu-1=alias-1...[, menu-12=alias-12]]  
{ i }

## BASIC

BASIC is an interactive programming language you can use from your workstation. To initialize BASIC, enter the following command:

BASIC

There are no parameters associated with this command.

## ESCORT

ESCORT™ is an interactive programming language that uses English statements to create a program. The ESCORT language allows you to generate reports and perform inquiry and update routines through the use of simple, sentence-like programs, entered through your workstation. To use ESCORT, log on and enter the following command:

ESCORT

There are no parameters associated with this command.

---

ESCORT is a trademark of Unisys Corporation.

## Section 8

# General Editor Commands

The following is a listing of the commands used by the Unisys OS/3 General Editor. For more information, see the *General Editor (EDT) Operating Guide*, UP-9976.

### EDT Commands

#### Command

a

#### Format

a line-number [increment] : { data }  
+ { command }  
-

#### Explanation

Sets the current line number and increment for data and command lines keyed in at the workstation.

---



**Command**

NUMBER

**Format**

@NU 'sequence-string'[\*n][BY increment]

**Explanation**

Inserts sequence numbers into input lines.

---

**Command**

PRINT

**Format**

@P [[line-range][search-string[\*n]]

**Explanation**

Displays specified lines from the current work-space file on the workstation screen.

---

**Command**

PUNCH

**Format**

@PU [[line-range]['search-string'[\*n]]Δ[IMMEDIATE]

**Explanation**

Reproduces specified lines from the current work-space file on cards.

---

**Command**

READ

**Format**

$\text{QRA}\Delta\text{FILENAME} = \left\{ \begin{array}{l} \text{filename} \\ \text{'filename'} \\ \text{"filename"} \end{array} \right\} [\text{,RDPASS=password}]$   
 $\text{,VSN=volume} \left[ \text{,KEYNO} = \left\{ \begin{array}{l} n \\ \emptyset \end{array} \right\} \right] \left[ \text{,DEVICE} = \left\{ \begin{array}{l} \text{did} \\ \text{DISK} \\ \text{DISKETTE} \end{array} \right\} \right]$   
 $[\text{,BFSZ=n}] \left[ \text{,TRUNC} = \left\{ \begin{array}{l} \text{YES} \\ \text{NO} \end{array} \right\} \right]$   
 $\Delta \left[ \left\{ \begin{array}{l} \text{KEY=start-col-no:end-col-no} \\ \text{KKEY=start-col-no:end-col-no} \\ \text{SHOW}\Delta\text{first-col-no:last-col-no} \end{array} \right\} \right]$

**Explanation**

Reads a MIRAM data file from disk or format label diskette.

---

**Command**

READ

**Format**

$$\text{@R}\Delta\text{FILENAME} = \left\{ \begin{array}{l} \text{filename} \\ \text{'filename'} \\ \text{"filename"} \end{array} \right\}, \text{VSN} = \text{volume}$$

$$, \text{DEVICE} = \left\{ \begin{array}{l} \text{did} \\ \text{DISKETTE} \\ \text{RDR} \end{array} \right\} \left[ \text{,TRUNC} = \left\{ \begin{array}{l} \text{YES} \\ \text{NO} \end{array} \right\} \right] \Delta \left[ \left\{ \begin{array}{l} \text{KEY} = \text{start-col-no: end-col-no} \\ \text{KKEY} = \text{start-col-no: end-col-no} \\ \text{SHOW}\Delta \text{first-col-no: last-col-no} \end{array} \right\} \right]$$

**Explanation**

Reads a unit record file from a data set label diskette or card reader.

---

**Command**

READ

**Format**

$$\text{@R}\Delta\text{FILENAME} = \left\{ \begin{array}{l} \text{filename} \\ \text{'filename'} \\ \text{"filename"} \end{array} \right\} \left[ \text{,RDPASS} = \text{password} \right]$$

$$, \text{VSN} = \text{volume}, \text{DEVICE} = \left\{ \begin{array}{l} \text{did} \\ \text{TAPE} \end{array} \right\} \left[ \text{,BKNO} = \left\{ \begin{array}{l} \text{YES} \\ \text{NO} \end{array} \right\} \right] \left[ \text{,TRUNC} = \left\{ \begin{array}{l} \text{YES} \\ \text{NO} \end{array} \right\} \right]$$

$$\left[ \left\{ \begin{array}{l} \text{KEY} = \text{start-col-no: end-col-no} \\ \text{KKEY} = \text{start-col-no: end-col-no} \\ \text{SHOW}\Delta \text{first-col-no: last-col-no} \end{array} \right\} \right]$$

**Explanation**

Reads a file from a tape.

---

**Command**

REMOVE

**Format**

@REM 'search-string'[\*n]

**Explanation**

Deletes a specified string from lines in the work-space file.

---

**Command**

SEQUENCE

**Format**

@SEQ { 'sequence-string'[\*n] } BY increment  
      \*

**Explanation**

Inserts sequence numbers into existing lines in the current work-space file.

---

**Command**

UPDATE

**Format**

@U [[line-range] ['search-string'[\*n]]]

**Explanation**

Displays specified lines from the work-space file one at a time for you to edit or change.

---

**Command**

WRITE

**Format**

$$\text{@W}\Delta\text{FILENAME} = \left\{ \begin{array}{l} \text{filename} \\ \text{'filename'} \\ \text{"filename"} \end{array} \right\} [\text{WRPASS}=\text{password}], \text{VSN}=\text{volume}$$

$$\left[ \text{,CONTIG} = \left\{ \begin{array}{l} \text{NO} \\ \text{YES} \end{array} \right\} \right] \left[ \text{,INC} = \left\{ \begin{array}{l} \text{n} \\ 1 \end{array} \right\} \right] \left[ \text{,INIT} = \left\{ \begin{array}{l} \text{NO} \\ \text{YES} \end{array} \right\} \right] \left[ \text{,RCB} = \left\{ \begin{array}{l} \text{YES} \\ \text{NO} \end{array} \right\} \right] \left[ \text{,RCFM} = \left\{ \begin{array}{l} \text{VAR} \\ \text{FIX} \end{array} \right\} \right]$$

$$\text{,RCSZ}=\text{n} \left[ \text{,SCSZ} = \left\{ \begin{array}{l} \text{n} \\ 256 \end{array} \right\} \right]$$

$$\left[ \text{,DEVICE} = \left\{ \begin{array}{l} \text{did} \\ \text{DISK} \\ \text{DISKETTE} \end{array} \right\} \right] \left[ \text{,SIZE}=\text{n} \right] \left[ \text{,EXTEND} = \left\{ \begin{array}{l} \text{YES} \\ \text{NO} \end{array} \right\} \right] \left[ \text{,BFSZ}=\text{n} \right]$$

$$\left[ \text{,KEY}i = \left\{ \begin{array}{l} \text{start-col-no:end-col-no} \\ \left( \text{start-col-no:end-col-no}, \left\{ \begin{array}{l} \text{DUP} \\ \text{NDUP} \end{array} \right\}, \left\{ \begin{array}{l} \text{CHG} \\ \text{NCHG} \end{array} \right\} \right) \end{array} \right\} \right]$$

**Explanation**

Writes a MIRAM data file to disk or format label diskette.



## EDT Variable Commands

### Command

ASSIGN

### Format

$$\text{aAS}\Delta\text{Gn} = \left\{ \begin{array}{l} \text{'string' [*n]} \\ \text{n(x:y)} \\ \text{n[+m]} \\ \text{Gm} \\ \text{LEN(n)} \end{array} \right\}$$

### Explanation

Assigns values to EDT variables.

---

### Command

DISPLAY

### Format

$$\text{aDI}\Delta\text{Gn} = \left\{ \begin{array}{l} \text{'string' [*n]} \\ \text{n(x:y)} \\ \text{n[+m]} \\ \text{Gm} \\ \text{LEN(n)} \end{array} \right\}$$

### Explanation

Displays a specified expression or the value of a specified expression from the work-space file on the workstation screen.

---

## EDT Procedure File Commands

### Command

DO

### Format

@DO proc-number  $\left\{ \begin{array}{l} \text{PRINT} \\ \text{NOPRINT} \\ \text{REVERT} \end{array} \right\}$

### Explanation

Executes a procedure file.

---

### Command

END

### Format

@E

### Explanation

Terminates procedure file definition.

---

**Command**

PROC

**Format**

@PRO [proc-number]

**Explanation**

Begins procedure file definition.

---

**Command**

RETURN

**Format**

@RET

**Explanation**

Terminates procedure file execution.

---

**Command**

DROP

**Format**

@DR

**Explanation**

Deletes all lines in the entire EDT work-space file.

---

**Command**

EFP

**Format**

@EFP

**Explanation**

Activates the error file processor.

---

**Command**

FORMAT

**Format**

@FORMAT parameter-string (for RPBEDT)  
@FORMAT (for COBEDT)

**Explanation**

Used only in conjunction with either RPGEDT or COBEDT. For information on the @FORMAT directive, see the current version of the *Report Program Generator II (RPG II) Editor Operations Guide*, UP-9981, or the *COBOL Editor (COBEDT) Programming Guide*, UP-9974.

---

**Command**

**SET**

**Format**

```

@SA [CHAR=tab-character, TABS= { columns }
    { OFF }
    [ ,LINE= { length }
    { 128 } ] [ ,EXCLUDE= { exclusion-character }
    { OFF } ]
[ ,ATSIGN=command-trigger ] [ ,COLON=range-separator ]
[ ,ENCOL=end-column ] [ ,BUFFER= { record-size }
    { OFF } ]
[ ,WIDTH=device-size ] [ ,CLEAR ] [ ,STRIP= { ON }
    { OFF } ] [ ,DISPLAY ]
[ ,SCRDSPLY= { TRUNCATE }
    { FOLD } ] [ ,ROLL= { 15 (if SCRDSPLY=TRUNCATE)
    8 (if SCRDSPLY=FOLD)
    1-15 } ]
[ ,MODE= { LINE }
    { SCREEN } ] [ ,LANGUAGE= { FREEFORM }
    { FORTRAN
    COBOL
    RPG } ] [ ,RECENTRY= { SINGLE }
    { MULT } ]
[ ,SCRFORM= { UNDERLINE }
    { BLANK } ] [ ,AUTO= { n }
    { OFF } ] [ ,BLANKS= { ON }
    { OFF } ]
[ ,SAVEFILE= { 'string' }
    { "string" } ]
    
```

## EDT Screen Commands

### Command

BLOCK

### Format

@BL

### Explanation

Displays a free-form screen that allows you to switch to block mode for entering multiple commands or data.

---

### Command

HELP

### Format

@HEA[error message code]

### Explanation

Displays help screens for any EDT error messages.

---

**Command**

ROLL

**Format**

@RO

**Explanation**

Displays free-form screens showing the EDT work-space file, where you can update lines or simply view them.

---

**Command**

SUMMARY

**Format**

@EFASUM

**Explanation**

Displays an error file summary for the module you're correcting.

---

## Section 9

# RPG II Editor and COBOL Editor

The RPG II and COBOL editors are actually subeditors of EDT. Therefore, you must first activate the General Editor before you can use a language editor. Once in a language editor session, you can use any of the EDT commands.

## RPG II Editor

To activate the RPG II editor, key in the following command:

```
EDTA@RPG
```

Following are commands you use in conjunction with the RPG II editor.

---

## COBOL Editor

To activate the COBOL editor, key in the following command:

EDT@COBOL

The following are commands you use in conjunction with the COBOL editor:

---

### Command

FORMAT

### Format

@FO parameters

### Explanation

Changes the display format type.

---

### Command

HALT

### Format

@H

### Explanation

Terminates the COBOL editor (and EDT) session.

---

# Section 10

## DDP Commands

The following is a list of special commands to perform DDP functions.

### Command and Format

DDPACREATEFILE= { host-id  
local-host-id } :: file-id

[ ABLOCK\_SIZE= { nnnnnnnn  
256 } ] [ ADENSITY= { 200  
556  
800  
1600  
6250  
host-SYSGEN:option } ]

[ ADEVICE\_CLASS= { DISK  
TAPE  
DISKETTE } ] [ AFILE\_TYPE= { SEQUENTIAL  
INDEXED  
LIBRARY  
UNDEFINED } ]

[ AINCREMENT\_SIZE= { nnnnnnnn  
3 cyl } ] [ AINITIAL\_SIZE= { nnnnnnnn  
3 cyl } ]

continued

### Command and Format

$$\text{DDP}\Delta\text{COPY}\Delta\text{FROM} = \left[ \begin{array}{l} \left\{ \text{originating-host-id} \right\} \\ \left\{ \text{local-host-id} \right\} \end{array} \right] :: \text{originating-file-id}$$

$$\Delta\text{TO} = \left[ \begin{array}{l} \left\{ \text{destination-host-id} \right\} \\ \left\{ \text{local-host-id} \right\} \end{array} \right] :: \text{destination-file-id}$$

$$\Delta\text{ELEMENT\_TYPE} = \left( \begin{array}{l} \text{SYMBOLIC} \\ \text{RELOCATABLE} \\ \text{ABSOLUTE} \\ \text{MACRO} \\ \text{PROC} \\ \text{COMPILED\_JOB} \\ \text{SCREEN\_FORMAT} \end{array} \right)$$

$$\Delta\text{KEY} \left[ \begin{array}{l} \left\{ n \right\} \\ \left\{ i \right\} \end{array} \right] = \text{size, location} \left[ \begin{array}{l} \Delta \left\{ \text{DUPLICATES} \right\} \\ \left\{ \text{NO\_DUPLICATES} \right\} \end{array} \right] \left[ \begin{array}{l} \Delta \left\{ \text{CHANGE} \right\} \\ \left\{ \text{NO\_CHANGE} \right\} \end{array} \right]$$

$$\left[ \begin{array}{l} \Delta\text{MODE} = \left\{ \text{DIRECT} \right\} \\ \left\{ \text{WAIT} \right\} \\ \left\{ \text{INDIRECT} \right\} \end{array} \right] \left[ \begin{array}{l} \Delta\text{POSITION} = \left\{ \text{EOF} \right\} \\ \left\{ \text{SOF} \right\} \end{array} \right] \left[ \begin{array}{l} \Delta\text{TRANSLATE} = \left\{ \text{ASCII} \right\} \\ \left\{ \text{EBCDIC} \right\} \\ \left\{ \text{NONE} \right\} \end{array} \right]$$

### Explanation

The COPY command permits you to copy a file or module from one system to another. You may copy a file or module from one remote system to another, from your local system to a remote system, or vice versa. You may also use the COPY command to copy a file within your local system.

---

**Command and Format**
$$\text{DDP}\Delta\text{CANCEL}\Delta\text{JOB}=\left[\begin{array}{l} \text{host-id} \\ \text{local-host-id} \end{array}\right]::\text{jobname}$$
$$\left[\begin{array}{l} \Delta\text{OUTPUT}=\left\{\begin{array}{l} \text{DISCARD} \\ \text{DELIVER} \end{array}\right\} \\ \Delta\text{COMMAND}=\left\{\begin{array}{l} \text{host-id} \\ \text{local-host-id} \end{array}\right\} \end{array}\right]$$
**Explanation**

The CANCEL command allows you to terminate a job either executing or scheduled for execution on a host system.

---

**Command and Format**
$$\text{DDP}\Delta\text{SUBMIT}\Delta\text{REQUEST}=\text{statement}\left[\begin{array}{l} \Delta\text{HOST}=\left\{\begin{array}{l} \text{host-id} \\ \text{local-host-id} \end{array}\right\} \end{array}\right]$$
**Explanation**

The SUBMIT REQUEST command allows you to send a statement, such as an operator or interactive command, to a host system. The following statements (commands) cannot be used: DISPLAY, DELETE, BREAKPOINT, FILE, IN, SU, TU, PD.

---

**Command and Format**

DDPATALKΔMESSAGE='string'ΔUSER=  $\left[ \begin{array}{l} \{ \text{host-id} \\ \{ \text{local-host-id} \} \end{array} \right] :: \left[ \begin{array}{l} \{ \text{OPERATOR} \} [\Delta\text{WAIT}] \\ \{ \text{user-id} \} \end{array} \right]$

**Explanation**

The TALK command allows you to send a message to a remote operator or user.

---

# Section 11

## Interactive Terminals

OS/3 interactive services can be accessed through local workstations, remote workstations, or terminals. Workstations are not supported on the model 7E. The following table shows the different procedures used to access interactive services through these types of terminals.

*Note: You do not need to enter the \$\$\$SON command to log on to either a workstation, the System 80 console workstation, or a non-DCP connected terminal. However, some system programs such as the Information Management System (IMS) require the use of ICAM to connect to workstations and terminals. In such cases, log on normally and then enter \$\$\$SON to sign on to IMS or other similar programs.*

## Interactive Terminals

Procedure Workstation	Workstation	UNISCOPE Terminals	UTS 400	System 80 Console Workstation
Function keys	Press FUNCTION and F1-F22 keys.	Press MESSAGE WAITING key. Then press F1-F4; for rest, press F, the pound symbol (#), then 5-22.	Press F1-F22 UPPER FUNCTION when required.	Press FUNCTION and F1-F22 keys.
Message waiting indicator	SYS MSG displayed on indicator line.	MESSAGE WAITING light is lit and the audible alarm sounds.	MESSAGE WAITING light is lit and the audible alarm sounds.	SYS MSG displayed on indicator line.
Save and restore SYS mode lines	YES	NO	NO	YES
LOGOFF	YES	YES	YES	YES
\$\$\$OFF	NO	YES	YES	NO

## Section 12

# Function Key Summary

This summary provides an overview of the function keys used by interactive services and the various interactive facilities.

Software Component	Key	Function
Interactive services	F15	Informs the system you have no more data to input from the workstation (end of file)
	F17	Temporarily halts the workstation display. (See F19.)
	F19	Restarts workstation display after it has been temporarily stopped using F17 or when the screen is full of data
General Editor	F1	Suppresses any printing options associated with a command. (Same as F18)
	F2	Terminates processing of a command
	F3	Displays a screen showing the parameters on the @SET directive, or those that make up your EDT environment. F3 is the same as issuing the PARAMS screen command.
	F4	Displays a freeform screen through which you can switch to block mode for entering multiple commands or data. F4 is the same as issuing the BLOCK screen command.

continued

## Function Key Summary

Software Component	Key	Function
Screen format services	F1	Returns the screen format generator to the HOME screen. Returning home deletes all work done up to that point.
	F5	Causes the screen format generator to breakpoint the spool file to a printer. All records currently in the spooled printer output are directed to a printer. Records added to the spool file after the breakpoint are held until a subsequent breakpoint is requested or until the end of the job.
	F13	Displays a HELP screen appropriate to the step you're at when generating a screen format. (See F14.)
	F14	Removes the HELP screen displayed by using F13; returns the screen format generator to the point it was at before display of the HELP screen.
	F15	Indicates end of input data
	F16	Indicates input data cannot be entered properly
	F20	Restores the screen to its original contents for the current pass if it has inadvertently been destroyed.
BASIC	F1	<p>Pauses or terminates execution of a BASIC program. When no I/O operation is in progress, the system displays the message:</p> <p style="text-align: center;">EXECUTION PAUSED AT LINE xxxx CONTINUE (Y,N)?</p> <p>Key in Y to continue (resume) execution. Key in N to terminate the program.</p>

continues

# Index

## A

- Allocating file space on diskettes, 6-1
- Altering the format of a file, 6-3
- Asking questions of other users or the system operator, 6-2
- Assigning any interactive command to a function key or MESSAGE  
WAITING key, 6-7
- Assigning values to EDT variables, 8-17
- Auxiliary printer, spool files, 5-8

## B

- BASIC
  - command, 7-5
  - function key summary, 12-3
- Beginning procedure file definition, 8-21
- Branching within a procedure file, 8-20

## C

- Card reader
  - running a command stream from, 6-9

## Changing

- device type or number of copies of a spool file, 5-6
- scheduling queue, 3-8, 8-2

## CHECK directive, 8-22

## Closing

- one or more spool files, 5-7
- workstation log files, 6-2

## COBOL directive, 8-22, 9-3

## COBOL editor, 8-22, 9-3

## Command stream

- entering comments, 6-16
- running
  - from a card reader, 6-9
  - from a DSL diskette, 6-9
  - from a library file, 6-8
  - from a spool file, 6-9
  - from a tape, 6-9

## Commands

- ALLOCATE, 6-1
- ASK, 6-2
- assigning to a function key, 6-7
- BASIC, 7-5
- BEGIN, 3-7
- BEGIN SPL, 5-4
- BRKPT, 5-7
- BRKPT LOG, 6-2
- CANCEL, 4-1, 10-5
- CJ, 4-1
- CHANGE, 3-8
- CHANGE SPL, 5-6
- changing job scheduling, 3-8
- COMMENT, 6-2
- CONNECT, 4-2

Copying files or modules  
  DDP, 10-3  
  EDT, 8-2  
CREATE command, DDP, 10-1

## D

Data files (*See* Files)  
Data set label diskette, running a  
  command stream 6-9  
Data utilities, interactive, 7-1  
DDP commands  
  CANCEL, 10-5  
  COPY, 10-3  
  CREATE, 10-1  
  PURGE, 10-4  
  STATUS, 10-6  
  SUBMIT, 10-4  
  SUBMIT REQUEST, 10-5  
  TALK, 10-7  
Deferring job scheduling, 3-8  
Deleting  
  files from spool queues, 5-5  
  jobs that are controlled from  
    scheduling queues, 3-9  
  library files, data files, and library  
    modules, 6-10  
Devices, freeing, 3-7  
Discarding queued system messages,  
  6-10  
Disconnecting a workstation from a  
  job, manually, 4-2  
Disk volume, file listing, 6-19

Diskette  
  running a command stream, 6-9  
  volume, file listing, 6-19  
Displaying  
  all or part of your workstation or  
    console log file, 6-15  
  contents of scheduling queues,  
    3-9  
  EDT command menu screen or  
    help screens for any EDT  
    command, 8-28  
  error file summary for the module  
    you are currently correcting,  
    8-31  
  errors, 8-30  
  function and MESSAGE  
    WAITING key assignments,  
    6-8  
  screen that shows the parameters  
    on the @SET directive, 8-28  
  specified lines from the  
    work-space file one at a time,  
    8-11  
  system or user-created menus,  
    6-11  
DMPRST routine, 7-4  
DROP directive, 8-23  
Dump  
  downline, 3-6  
  upline, 3-6  
DUMP/RESTORE hardware utility,  
  7-4

- function keys or MESSAGE
  - WAITING key from a command assignment, 6-7
- Function keys
  - assigning a command, 6-7
  - summary, 12-1
- G**
- General editor commands
  - @, 8-1
  - ASSIGN, 8-17
  - BLOCK, 8-27
  - CHANGE, 8-2
  - COPY, 8-2
  - DELETE, 8-2
  - directives (See General editor directives)
  - DISPLAY, 8-17
  - DO, 8-19
  - EFP commands, 8-30
  - END, 8-19, 8-30
  - FIND, 8-3
  - FSTATUS, 8-3
  - GOTO, 8-20
  - HELP, 8-27
  - IF, 8-18
  - INSERT, 8-4
  - LIST, 8-4
  - MOVE, 8-4
  - NOP, 8-20
  - NUMBER, 8-5
  - PARAMS, 8-28
  - PRINT, 8-5
  - PROC, 8-21
  - procedure file commands, 8-19
  - PROMPT, 8-28
  - PUNCH, 8-5
  - READ, 8-6
  - REMOVE, 8-11
  - RESTORE, 8-28
  - RETURN, 8-21
  - ROLL, 8-29
  - screen commands, 8-27
  - SEQUENCE, 8-11
  - SUMMARY, 8-31
  - UPDATE, 8-11
  - variable commands, 8-17
  - WRITE, 8-12
- General editor directives
  - CHECK, 8-22
  - COBOL, 8-22, 9-3
  - DROP, 8-23
  - EFP, 8-23
  - FORMAT, 8-23, 9-2
  - function key summary, 12-1
  - HALT, 8-24, 9-2
  - initializing, 6-8
  - RPG, 8-24, 9-1
  - SET, 8-25
  - SYSTEM, 8-26
- H**
- HALT directive, 8-24, 9-2
- Halting job processing
  - DDP, 10-5
  - description, 4-1
- Hardware utility, interactive
  - DUMP/RESTORE, 7-4

Library modules  
  deleting, 6-10  
  obtaining information about, 6-11  
  printing, 6-12  
  punching, 6-13  
  SAT (*See* SAT library modules)

Lines, editing (*See* General editor commands)

Loading  
  output writer to print a file, 5-4  
  output writer to print (PR) or punch (PU) spooled files, 5-7  
  program to terminal memory, 3-6

Loading and executing a procedure file, 8-20

Locating the first occurrence of a string in a work-space file, 8-3

Log files  
  closing, 6-2  
  displaying, 6-15  
  displaying status, 6-8

Logging on to a workstation session, 2-1

## M

Manually disconnecting a workstation from a job, 4-2

Manually loading output writer to print (PR) or punch (PU) spooled files, 5-7

Menu generator, 7-3

Menus  
  displaying, 6-11  
  including in your program, 7-3

MESSAGE WAITING key, assigning a command to, 6-7

Messages  
  error, HELP command, 6-11  
  restoring, 6-15  
  sending, 6-19  
  system, discarding, 6-10  
  using more than one line to enter, 1-1

MIRAM data files  
  copying, 6-4  
  deleting, 6-10  
  printing, 6-12  
  punching, 6-14  
  reading, 8-7  
  writing, 8-13

MIRAM library modules  
  copying, 6-3  
  printing, 6-12  
  reading, 8-6  
  writing, 8-12

Mode  
  SYSTEM, 1-1  
  WORKSTATION, 1-1

Modules  
  copying, 10-3  
  library (*See* Library modules)

## N

Notation conventions, ix

**S**

## SAT library modules

- copying, 6-3
- printing, 6-12
- reading, 8-6
- recovering, 6-16
- writing, 8-12

## Scheduling queues

- changing, 3-8
- displaying contents, 3-9
- removing jobs, 3-9

## Screen commands, EDT, 8-27

## Screen format generator, 7-2

## Screen format services, function key

- summary, 12-3

## Screen formats, using in your

- program, 7-2

## Sending messages that do not require

- a response, 6-19

## @SET directive, 8-25

## Showing the EDT work-space file,

- 8-29

## Spool file directories, 5-1

## Spool files

- changing device type or number of copies, 5-6
- closing, 5-7
- copying, 6-5
- deleting, 5-5
- holding, 5-3
- obtaining information about, 5-2
- print to auxiliary printer, 5-8
- printing, 5-7, 6-13
- punching, 5-7, 6-14
- reading, 8-8
- releasing, 5-4

- running a command stream, 6-9
- writing, 8-14

## Spooling commands

- BEGIN SPL, 5-4
- BRKPT, 5-7
- CHANGE SPL, 5-6
- DELETE SPL, 5-5
- DISPLAY ACT, 5-2
- DISPLAY SPL, 5-2
- HOLD SPL, 5-3
- PR/PU, 5-7
- RP, 5-8

## SUBMIT command, DDP, 10-4

## SUBMIT REQUEST command, DDP,

- 10-5

## Subsystem, resuming execution of,

- 6-17

## Suspending job processing, 4-2

## Switching to block mode to enter

- multiple commands or data,
- 8-27

## SYSTEM directive, 8-26

## System messages, discarding, 6-10

## SYSTEM mode, 1-1

## System status, 6-19

**T**

## TALK command, DDP, 10-7

## Tape

- reading a file from, 8-9
- running a command stream from,
- 6-9
- writing a file from, 8-15

**V**

Variables, EDT, 8-17

Volumes

file listing of, 6-19

status of, 6-19

**W**

Workstation log files

closing, 6-2

displaying status, 6-8

WORKSTATION mode

connecting to a job, 4-1

description, 1-1

disconnecting from a job, 4-2

Workstations

altering characteristics, 6-18

closing log file, 6-2

controlling jobs, 4-1

disconnecting from a job, 4-2

displaying status of log file, 6-8

entering commands as a batch job,  
6-8

executing jobs, 3-3

HELP command, 6-11

interactive terminals, 11-1

remote terminals, 2-2

resuming subsystem execution,  
6-17

terminating a session, 6-17

using correct mode, 1-1

using more than one line to enter  
a command, 1-1

Writing a SAT or MIRAM library

module to disk or format

label diskette, 8-12



## Index

---

- Tape files
  - copying, 6-6
  - printing, 6-12
  - punching, 6-14
- Terminals
  - altering characteristics, 6-18
  - downline loading memory, 3-6
  - dumping memory, 3-6
  - interactive, 11-1
  - standard dialog, ICAM, 2-2
  - terminating a session, 6-17
- Terminating
  - job at the completion of the
    - currently executing job step, 4-3
  - procedure file definition or the error file processor, 8-19, 8-30
  - procedure file execution, 8-21
  - single command, specific
    - workstation or terminal user session, or all user sessions, 6-17, 8-11
- Transferring specified lines to new line locations in the work-space file, 8-4
  
- U**
- UNISCOPE terminals, 11-1
- Unit record files
  - copying, 6-7
  - printing, 6-12
  - punching, 6-15
  - reading, 8-9
  - writing, 8-14
- Upline dump, 3-6
- Utility commands
  - ALLOCATE, 6-1
  - ASK, 6-2
  - BRKPT LOG, 6-2
  - COMMENT, 6-2
  - COPY, 6-3
  - DEFKEY (Delete), 6-7
  - DISPLAY LOG, 6-8
  - EDT, 6-8
  - ENTER, 6-8
  - ERASE, 6-10
  - FLUSH, 6-10
  - FSTATUS, 6-11
  - HELP, 6-11
  - MENU, 6-11
  - PRINT, 6-12
  - PUNCH, 6-13
  - REBUILD, 6-15
  - RECALL, 6-15
  - RECOVER, 6-16
  - REMARK, 6-16
  - REMOVE, 6-17
  - RESUME, 6-17
  - SCREEN, 6-18
  - STATUS, 6-19
  - TELL, 6-19
  - VTOC, 6-19
- UTS 400
  - downline loading terminal memory, 3-6
  - dumping terminal memory, 3-6
  - interactive terminals, 11-1

## Index

---

### O

- Output writer
  - spool files, 5-7
  - workstation log files, 6-2

### P

- Placing files in a hold state, 5-3
- Printer, spool files, 5-8
- Printing
  - copies, 6-12, 8-5
  - specified lines from the current work-space, 8-4
- Procedure file commands, EDT, 8-19
- Producing a listing of the files on the disk or diskette volume, 6-19
- Programs
  - running in interactive environment, 3-6
  - using menus, 7-3
  - using screen formats, 7-2
- Punched-card copies, making, 6-13, 8-5
- PURGE command, DDP, 10-4
- Putting or replacing a comment in a library module header, 6-2

### Q

- Queues, scheduling (*See* Scheduling queues)

### R

- Reactivating jobs suspended by the PAUSE command, 4-3
- Reading a SAT or MIRAM library module from disk or format label diskette, 8-6
- Recovering deleted source, procedure, or macro modules in the SAT library, 6-16
- Releasing spool files held by a HOLD SPL command, 5-4
- Remote terminal as workstation, using, 2-2
- Removing jobs from scheduling queues, 3-9
- Rescheduling jobs deferred by the HOLD command, 3-7
- Responses, using more than one line to enter, 1-1
- Restarting suspended jobs, 4-3
- Restoring messages that require a response or a GO command, 6-15
- Resuming execution of a subsystem that was suspended when the workstation entered SYSTEM mode, 6-17
- RPG directive, 8-24, 9-1
- RPG II
  - editor commands, 8-24, 9-1
  - errors, 8-30
- Running programs in an interactive job environment, 3-6

## Index

---

### I

ICAM (Integrated Communications Access Method), 2-2

Initializing

- general editor, 6-8
- interactive DUMP/RESTORE hardware utility, 7-4
- menu generator, 7-3

Inserting

- sequence numbers into input lines, 8-5
- sequence numbers into existing lines in the current work-space file, 8-11
- specified string in the current work-space file, 8-4

Integrated Communications Access Method (ICAM), standard terminal dialog, 2-2

Interactive data utilities, 7-1

Interactive DUMP/RESTORE hardware utility, 7-4

Interactive facilities

- commands, 7-1
- function key summary, 12-1

Interactive terminals, 11-1

### J

Job control dialog, 7-2

Job control procedures, filing, 3-1

Job control streams, SUBMIT command, 10-4

Job scheduling, changing, 3-1

Job status, displaying, 4-2

### Jobs

- canceling, 4-1
- changing scheduling queue, 3-8
- connecting WORKSTATION mode, 4-2
- controlling from workstation, 4-1
- deferring scheduling, 3-8
- deleting from scheduling queue, 3-9
- disconnecting a workstation, 4-2
- executing from workstation, 3-2
- filing, 3-1
- obtaining information about, 4-2
- reactivating after suspending, 4-3
- rescheduling, 3-7
- running, 3-1
- running jobs saved in expanded state, 3-4
- soft canceling, 4-1
- suspending processing, 4-2
- terminating, 4-3

jprocs (job control procedures), 3-1

### L

Library files

- deleting, 6-10
- printing, 6-12
- punching, 6-13
- running a command stream, 6-8

Library module header comments, 6-2

## E

- EDT command (*See* General editor commands)
- EFP directive, 8-23
- Embedded data, replacing, 3-4
- Entering
  - a comment in a stream of commands, 6-16
  - a series of workstation commands as a batch processing job, 6-8
  - any workstation command except those directly affecting the workstation device, 6-8
  - commands, responses, or messages, 1-1
  - extra lines for branching or comments into a procedure file, 8-20
- Error file processor
  - EFP commands, 8-30
  - EFP directive, 8-23
- Error messages, HELP command, 6-11, 8-27
- ESCORT
  - command, 7-5
  - function key summary, 12-4
- Executing an EDT command or procedure file command, 8-18
- Executing procedure files, 8-19
- Executing user jobs from workstations in an interactive job environment
  - input device needed, 3-2

- input device not needed, 3-3
  - previous job with changes, 3-4
  - previous job with no changes, 3-5
- Expressions, displaying, 8-17

## F

- Files
- allocating space, 6-1, 10-2
  - cataloging, 10-2
  - copying, 6-3, 10-3
  - creating, 10-2
  - deleting, 6-10, 10-4
  - establishing as a receiving host, 10-2
  - library (*See* Library files)
  - log (*See* Log files)
  - MIRAM (*See* MIRAM data files)
  - obtaining information, 6-11
  - printing, 6-12
  - punching, 6-13
  - reading, 8-6
  - spool (*See* Spool files)
  - unit record (*See* Unit record files)
- Filing jobs and jprocs into a library files, 3-1
- FORMAT directive, 8-23, 9-2, 9-3
- Formatted screen displays in your program, including, 7-2
- FORTRAN IV errors, 8-30
- Freeing
  - devices allocated to downline loaded programs, 3-7

## Index

---

- controlling jobs, 4-1
- COPY, 6-3, 8-2, 10-3
- DDP (See DDP commands)
- DEFKEY (Delete), 6-7
- DELETE, 3-9, 8-2
- DELETE SPL, 5-5
- DISPLAY ACT, 5-2
- DISPLAY JBQ, 3-9
- DISPLAY JS, 4-2
- DISPLAY LOG, 6-8
- DISPLAY SPL, 5-2
- DLOAD, 3-6
- EDT, 6-8, 8-1
- ENTER, 6-8
- ERASE, 6-10
- ESCORT, 7-5
- EXECUTE, 3-6
- FILE, 3-1
- FLUSH, 6-10
- FSTATUS, 6-11, 8-3
- general editor (See General editor commands)
- GO, 4-3
- HELP, 6-11, 8-27
- HOLD, 3-8
- HOLD SPL, 5-3
- HU, 7-4
- interactive data utilities, 7-1
- job control dialog, 7-2
- LOGON, 2-1
- MENU, 6-11
- PAUSE, 4-2
- PR/PU, 5-7
- PRINT, 6-12, 8-5
- PUNCH, 6-13, 8-5
- REBUILD, 6-15
- RECALL, 6-15
- RECOVER, 6-16
- REMARK, 6-16
- REMOVE, 6-17, 8-11
- RESUME, 6-17
- RP, 5-8
- RUN, 3-2
- running jobs, 3-1
- RV, 3-3
- SC, 3-5
- SCREEN, 6-18
- screen format generator, 7-2
- SI, 3-4
- \$\$SON, 2-2, 11-1
- spooling, 5-1
- STATUS, 6-19, 10-6
- STOP, 4-3
- TELL, 6-19
- terminating a single command, 6-17
- ULD, 3-6
- UNLOAD, 3-7
- USE MENU, 7-3
- USE SFS, 7-2
- using more than one line to enter, 1-1
- using screen formats, 7-2
- utility, 6-1
- VTOC, 6-19
- Comments
  - entering in a command stream, 6-16
  - putting or replacing in a library module header, 6-2
- Connecting
  - terminal to ICAM (\$\$SON), 2-2, 11-1
  - WORKSTATION mode to a job running on the system, 4-2
- Console log file, displaying, 6-15
- Continuation character, 1-1
- Controlling
  - input/output spooling commands, 1-1
  - jobs, 1-2

## Function Key Summary

---

Software Component	Key	Function
BASIC (cont.)		<ul style="list-style-type: none"> <li>• If a BASIC program is requesting output, you must press XMIT after pressing F1 to display the above message.</li> <li>• If the BASIC program is sending output, the workstation screen must be full of data for F1 to be recognized. Press F19 to display the above message.</li> </ul>
	F19	See F1.
ESCORT	F1	Signals the end of input and returns to master menu. (Used only with the structure processor)
	F2	Cancels display output. (Used only with the structure processor)
	F3	Cancels the current screen and returns to the previous screen. (Used only for program and tutorial modes)
	F4	<ul style="list-style-type: none"> <li>• Structure processor: aborts structure and returns to menu</li> <li>• Program mode: ends free-form input and returns to previous menu</li> <li>• Run-time processor: terminates program and returns to caller</li> </ul>

## Function Key Summary

---

Software Component	Key	Function
General Editor (cont.)	F5	Displays freeform screens, showing the EDT work-space file where you can update lines or simply view them. F5 is the same as issuing the ROLL screen command.
	F6	Displays help screens for any EDT error messages. F6 is the same as issuing the HELP screen command.
	F12	Shows a previously displayed additional help screen for a specific command when that command requires several help screens to fully describe it.
	F13	Displays the EDT command menu screen and help screens for any of the EDT commands. F13 is the same as issuing the PROMPT screen command. From a help screen, F13 also lets you see subsequent help screens needed to fully describe the command.
	F14	Returns you to the point in your EDT session where you originally entered a screen command. F14 is the same as issuing the RESTORE screen command.
	F15	Recognized as EOF indicator. Halts an EDT session run in batch mode or produces an error message in interactive mode.
	F18	Suppresses printing option associated with a command. (Same as F1)
	F19	Restarts workstation display after it has been temporarily stopped using F17 or when the screen is full of data

continues



## Interactive Terminals

---

Procedure	Local Workstation	Remote Workstation	Terminal Not Connected to DCP	Terminal Connected to I
\$\$\$ON	NO	NO	NO	YES
LOGON	YES	YES	YES	YES
SYSTEM mode	Press FUNCTION and SYS MODE keys, or press TRANSMIT.	Press FUNCTION and SYS MODE keys, or press TRANSMIT.	Press MESSAGE WAITING key.	Press MESSAGE WAITING key.
WORKSTATION mode	Press FUNCTION and WS MODE keys.	Press FUNCTION and WS MODE keys.	Press TRANSMIT.	Press TRANSMIT.
Save and restore SYS mode lines	YES	YES	NO	NO
Message waiting indicator	SYS MSG displayed on indicator line.	SYS MSG displayed on indicator line.	MESSAGE WAITING light is lit. Audible alarm sounds.	MESSAGE WAITING light is lit. Audible alarm sounds.
LOGOFF	YES	YES	YES	YES
\$\$\$OFF	NO	NO	YES	YES



## DDP Commands

---

### Command and Format

```
DDPASTATUSΔ { COMMAND=work-order-number  
FILE= [ { host-id } :: file-id;  
        { local-host-id } ]  
[keyword parameter]  
HOST=host-id  
JOB= [ { host-id } :: jobname  
      { local-host-id } ]  
USER= [ { host-id } :: user-id  
       { local-host-id } ] }
```

### Explanation

The STATUS command enables you to obtain information about:

- Commands entered
  - Host systems in your DDP system
  - Jobs you have submitted
  - Files in your DDP system
  - Other users on your DDP system
-

## DDP Commands

---

### Command and Format

$$\text{DDP}\Delta\text{PURGE}\Delta\text{FILE}=\left[ \begin{array}{l} \text{host-id} \\ \text{local-host-id} \end{array} \right] :: \text{file-id}$$

### Explanation

The PURGE command allows you to physically remove a file, and all references to it, from a host system.

---

### Command and Format

$$\text{DDP}\Delta\text{SUBMIT}\Delta\text{FILE}=\left[ \begin{array}{l} \text{originating-host-id} \\ \text{local-host-id} \end{array} \right] :: \text{file-id}$$
$$\left[ \begin{array}{l} \Delta\text{ELEMENT\_TYPE}=\left\{ \begin{array}{l} \text{SYMBOLIC} \\ \text{COMPILED\_JOB} \end{array} \right\} \end{array} \right]$$
$$\left[ \begin{array}{l} \Delta\text{HOST}=\left\{ \begin{array}{l} \text{destination-host-id} \\ \text{local-host-id} \end{array} \right\} \end{array} \right]$$

### Explanation

The SUBMIT command allows you to send a file of job control streams to a host system for execution. You can also use it to initiate a file of job control streams already at the host system or to bring a job control stream to your local system for execution.

---

## DDP Commands

---

$$\left[ \begin{array}{l} \Delta \text{KEY} - \left\{ \begin{array}{l} n \\ \text{I} \end{array} \right\} = \text{size, location} \left[ \begin{array}{l} \Delta \left\{ \begin{array}{l} \text{DUPLICATES} \\ \text{NO\_DUPLICATES} \end{array} \right\} \\ \\ \Delta \left\{ \begin{array}{l} \text{CHANGE} \\ \text{NO\_CHANGE} \end{array} \right\} \end{array} \right] \end{array} \right]$$
$$\left[ \begin{array}{l} \Delta \text{PARITY} = \left\{ \begin{array}{l} \text{ODD} \\ \text{EVEN} \end{array} \right\} \left[ \begin{array}{l} \Delta \text{RECORD\_FORM} = \left\{ \begin{array}{l} \text{FIXED} \\ \text{VARIABLE} \\ \text{UNDEFINED} \end{array} \right\} \end{array} \right] \end{array} \right]$$
$$\left[ \begin{array}{l} \Delta \text{RECORD\_SIZE} = \left\{ \begin{array}{l} \text{nnnnnn} \\ 256 \end{array} \right\} \left[ \begin{array}{l} \Delta \text{REGISTER} = \left\{ \begin{array}{l} \text{VTOC} \\ \text{CATALOG} \end{array} \right\} \end{array} \right] \end{array} \right]$$

### Explanation

The CREATE command:

- Establishes a file on a receiving host
- Allocates space for the file
- Catalogs the file in your online system catalog
- Records the file in the volume table of contents (VTOC) of the volume at the remote host on which the file is created

*Note: The default for INCREMENT SIZE and INITIAL SIZE is three cylinders. If more or less than three cylinders is needed, the size must be entered in number of blocks (nnnnnnnnn).*

---



## RPG II Editor and COBOL Editor

---

### Command

FORMAT

### Format

@FO specification-type  
@FO specification-type,format-type  
@FO ,format-type  
@FO ,format-type,CMD

### Explanation

Allows you to change the display format type and/or switch the RPG II editor from the update mode to the create mode.

---

### Command

HALT

### Format

@H

### Explanation

Terminates the RPG II (and EDT) session.

---



## EFP Commands

### Command

EFP

### Format

To correct and display COBOL and RPG II errors and FORTRAN IV™ errors for one source module at a time, use:

```
@EF[X]Δ[program-unit-name]Δ[error-range]Δ['search-string']
```

To correct and display FORTRAN IV errors for compilations that process multiple source modules, use:

```
@EFA$SOURCEΔsource-module-name,source-file-name,vsn
```

### Explanation

Displays errors in your error file along with the source lines that contain those errors. Note that EFP is both an EDT directive and an EFP command.

---

### Command

END

### Format

```
@EFAEND
```

### Explanation

Terminates the error file processor.

---

---

FORTRAN IV is a trademark of SuperSoft Associations.

**Command**

PARAMS

**Format**

@PA

**Explanation**

Displays a screen showing the parameters on the @SET directive (those that make up your EDT environment).

---

**Command**

PROMPT

**Format**

@PROMΔ[EDT command]

**Explanation**

Displays the EDT command menu screen or help screens for any EDT command (meaning EDT commands, modifiers, directives, procedure file commands, variables, and screen commands).

---

**Command**

RESTORE

**Format**

@RES

**Explanation**

Returns you to the point in your EDT session where you originally entered a screen command.

---

## General Editor Commands

---

### Explanation

Defines various parameters to EDT that collectively make up your EDT environment.

---

### Command

SYSTEM

### Format

@SYΔ[workstation-command]

### Explanation

Permits workstation commands to be issued during an EDT session or temporarily returns you to system mode.

---

## General Editor Commands

---

### Command

HALT

### Format

@H

### Explanation

Terminates the EDT session.

---

### Command

RPG

### Format

@RPG

### Explanation

Activates the RPG II editor.

---

## General Editor Commands

---

### EDT Directives

#### Command

CHECK

#### Format

aCHECK { OFF }  
          { ON }

#### Explanation

Determines whether processed lines are to be displayed on the workstation screen.

---

#### Command

COBOL

#### Format

aCOB

#### Explanation

Activates the COBOL editor.

---

## General Editor Commands

---

### Command

GOTO

### Format

@GA { line }  
      { label }

### Explanation

Permits branching within a procedure file.

---

### Command

INPUT

### Format

@INPAfile-parameters  $\left[ \begin{array}{c} \text{PRINT} \\ \text{NOPRINT} \\ \text{REVERT} \end{array} \right]$

### Explanation

Loads and executes a procedure file.

---

### Command

NOP

### Format

@NOPΔ[comment]

### Explanation

Enters extra lines for branching or comments into a procedure file.

---

## General Editor Commands

---

### Command

IF

### Format

@IF.condition.command

or

@IF expression relation expression command

### Explanation

Permits an EDT command or EDT procedure file command to be executed based on some condition.

---

## General Editor Commands

---

### Command

WRITE

### Format

@WΔ;Δvalid EDT command

### Explanation

Writes to the same module or file last accessed through a previous @READ or @WRITE command but written now with any valid EDT command specified.

---

## General Editor Commands

---

### Command

WRITE

### Format

$$\text{@W}\Delta[\text{JOB}=\text{jobname}] \left[ \begin{array}{c} \text{HOLD}=\left\{ \begin{array}{c} \text{NO} \\ \text{YES} \end{array} \right\} \end{array} \right] \left[ \begin{array}{c} \text{,FILENAME}=\left\{ \begin{array}{c} \text{filename} \\ \text{'filename'} \\ \text{"filename"} \end{array} \right\} \end{array} \right]$$
$$\left[ \begin{array}{c} \text{,ACCT}=\text{acct-no}, \text{,QUEUE}=\left\{ \begin{array}{c} \text{PRINT} \\ \text{PUNCH} \\ \text{RDR} \end{array} \right\} \end{array} \right] \left[ \begin{array}{c} \text{,COPIES}=\left\{ \begin{array}{c} \text{n} \\ \text{1} \end{array} \right\} \end{array} \right]$$

### Explanation

Writes a file to the spool file.

---

### Command

WRITE

### Format

$$\text{@W}\Delta\text{FILENAME}=\left\{ \begin{array}{c} \text{filename} \\ \text{'filename'} \\ \text{"filename"} \end{array} \right\}, \text{VSN}=\text{volume}, \text{DEVICE}=\left\{ \begin{array}{c} \text{did} \\ \text{DISKETTE} \\ \text{PRINT} \\ \text{PUNCH} \end{array} \right\}$$
$$\left[ \begin{array}{c} \text{,RCFM}=\left\{ \begin{array}{c} \text{FIX} \\ \text{VAR} \end{array} \right\} \end{array} \right] \left[ \text{,RCSZ}=\text{n} \right]$$

### Explanation

Writes a unit record file to a printer, punch, or diskette.

---

## General Editor Commands

---

### Command

WRITE

### Format

@WAMODULE=module-name [ ,TYPE= { module-type } ]

AFILNAME= { filename  
'filename'  
"filename" } [ ,WRPASS=password ] [ ,DEVICE= { did  
DISK  
DISKETTE } ]

,VSN=volume [ ,CONTIG= { NO  
YES } ] [ ,INC= { n  
1 } ] [ ,RCSZ=n ]

[ ,SIZE=n ] [ ,SAT= { NO  
YES } ]

### Explanation

Writes a SAT or MIRAM library module to disk or format label diskette.

---

## General Editor Commands

---

### Command

READ

### Format

@R

### Explanation

Reads the same module or file last accessed through a previous @READ or @WRITE command.

---

### Command

READ

### Format

@RΔ;Δ  $\left\{ \begin{array}{l} \underline{KEY}=\text{start-col-no:end-col-no} \\ \underline{KKEY}=\text{start-col-no:end-col-no} \\ \underline{SHOW}\Delta\text{first-col-no:last-col-no} \end{array} \right\}$  [valid EDT command]

### Explanation

Reads the same module or file last accessed through a previous @READ or @WRITE command but read now with a previous KEY, KKEY, or SHOW parameter or any valid EDT command specified.

---



## General Editor Commands

---

### Command

READ

### Format

@RΔMODULE=module-name,FILENAME= { filename  
  'filename'  
  "filename" }

[ ,TRUNC= { YES } ] [ ,RDPASS=password ],VSN=volume  
  NO }

[ ,TYPE= { module-type } ] [ ,DEVICE= { did  
  DISK  
  DISKETTE } ]  
  S }

Δ [ { KEY=start-col-no:end-col-no  
          KKEY=start-col-no:end-col-no  
          SHOWΔfirst-col-no:last-col-no } ]

### Explanation

Reads a SAT or MIRAM library module from disk or format label diskette.

---

## General Editor Commands

---

### Command

INSERT

### Format

@I 'change-string'[\*n]

### Explanation

Inserts a specified string into lines in the current work-space file.

---

### Command

LIST

### Format

@L [[line-range] ['search-string'[\*n]] Δ[IMMEDIATE]

### Explanation

Prints specified lines from the current work-space file on the printer.

---

### Command

MOVE

### Format

@M [[line-range] ['search-string'[\*n]] TO destination

### Explanation

Transfers specified lines to new line locations in the work-space file and deletes the original lines and line numbers.

---

## General Editor Commands

---

### Command

CHANGE

### Format

@C ['search-string'[\*n]]TO 'change-string'[\*n]

### Explanation

Replaces an existing string in the current work-space file with a new string.

---

### Command

COPY

### Format

@CO [[line-range] ['search-string'[\*n]] TO destination

### Explanation

Copies lines in the current work-space file to new line locations without deleting the original lines.

---

### Command

DELETE

### Format

@D [[line-range] ['search-string'[\*n]]

### Explanation

Erases specified lines and their line numbers from the current work-space file.

---



## Interactive Facilities

---

### Explanation

$\left. \begin{array}{l} \text{menu-file-ld-1/menu-file-ld-2} \\ \text{SYSFMT/menu-file-ld-2} \\ \text{menu-file-ld-1/SYSFMT} \\ \text{menu-file-ld-1} \\ \text{SYSFMT} \end{array} \right\}$

Specifies the lfd name or names for up to two files that will be searched for the correct menu.

`initial-menu`

Specifies the name of the first or only menu to be used by a program.

`nnn`

||

Specifies the number of menus to reside in main storage for use with a given file.

`[,menu-1=alias-1...[,menu-12=alias-12]]`

Specifies that a name other than the real name of the menu is to be used. Maximum of 12 aliases permitted. More than 12 causes rejection of the job control stream.

## Interactive DUMP/RESTORE Hardware Utility

The dump/restore hardware utility lets you interactively initiate and control the DMPRST routine from your workstation. The DMPRST routine creates backup copies of your program and data libraries on disk, tape (including streaming tape), or diskette.

To initialize the interactive dump/restore hardware utility, enter the following command:

`HU`

There are no parameters associated with this command.

### Job Control Dialog

To initialize the job control dialog, enter the following command:

```
RVAJCSBLD
```

### Screen Format Generator

Use the following command to initialize the screen format generator:

```
RVASFGEN
```

There are no parameters associated with this command.

### Using Screen Formats in Your Program

Use the following job control statement to include formatted screen displays in your programs:

```
//[symbol]USE SFS [ { format-file-lfd-1/format-file-lfd-2 }  
                  { format-file-id  
                  {SYSEMT
```

```
[initial-screen] [ { nnn }  
                  { i } ]
```

```
[,screen-format-1=alias-1...,screen-format-12=alias-12]
```

#### Explanation

```
{ format-file-id }  
{SYSEMT
```

Specifies the LFD name(s) of the screen format file.

initial-screen

Specifies the first format name to be used in behalf of the user program.



## SCREEN Command

The SCREEN command enables you to alter certain characteristics of the workstation or terminal you're using. You may use the SCREEN command to change different characteristics on different terminals acting as workstations; consult the *Interactive Services Operating Guide*, UP-9972, for further information.

```
SCREENA { SCROLL } [ , { UPPER } ] [ , XMIT= { VAR } ] [ , XFER= { VAR } ]
        { ROLL }
        { NP }
        { WRAP }
        { NOROLL }
```

```
[ , SPEED= { 9600 } ] [ , SPACEBAR= { DESTRUCT } ] [ , LINES= { 24 } ]
        { 4800 }
        { 2400 }
        { 1200 }
        { 600 }
        { 300 }
```

```
[ , KEYBOARD= { STANDARD } ] [ , INTENSITY= { NORMAL } ] [ , LOG= { ALL } ]
        { KATAKANA }
        { LOW }
        { REVERSE }
```

```
[ , { CENTRAL } ] [ , { NONBURST } ] [ , { CONTINUOUS } ] [ , SI= { TOP } ] [ , { HEADER } ]
        { WKSTN }
        { BURST }
        { PAGE }
        { BOTTOM }
        { NOHEADER }
```

## RECOVER Command

The RECOVER command allows you to recover deleted source, procedure, or macro modules in the SAT library. You may also use the command to rename modules not deleted.

$$\text{RECOVER}\Delta\text{MODULE}=\text{modulename} \left[ \text{,TYPE}=\left\{ \begin{array}{l} \text{module-type} \\ \text{\$} \end{array} \right\} \right] \text{,FILENAME}=\left\{ \begin{array}{l} \text{filename} \\ \text{'filename'} \\ \text{"filename"} \end{array} \right\}$$

[,RDPASS=password][,WRPASS=password],VSN=volume

## REMARK Command

The REMARK command allows you to enter a comment in a stream of commands. It is used principally in card decks for batch processing, but may be used in any situation where a comment needs to be inserted in a command stream.

REMARK $\Delta$ text

## Utility Commands

---

### Punching a MIRAM Data File

$\text{PUNCH}\Delta\text{FILENAME} = \left\{ \begin{array}{l} \text{filename} \\ \text{'filename'} \\ \text{"filename"} \end{array} \right\} [ , \text{RDPASS} = \text{password} ] , \text{VSN} = \text{volume} \left[ , \text{KEYNO} = \left\{ \begin{array}{l} n \\ 0 \end{array} \right\} \right]$

$\left[ , \text{COPIES} = \left\{ \begin{array}{l} n \\ 1 \end{array} \right\} \right] \left[ , \text{DEVICE} = \left\{ \begin{array}{l} \text{did} \\ \text{DISKETTE} \\ \text{DISK} \end{array} \right\} \right] \Delta [ \text{DIRECT} ] [ , \text{WAIT} ]$

### Punching a Spool File

$\text{PUNCH}\Delta\text{JOB} = \text{jobname} \left[ , \text{HOLD} = \left\{ \begin{array}{l} L \\ N \\ Y \end{array} \right\} \right] , \text{FILENAME} = \left\{ \begin{array}{l} \text{filename} \\ \text{'filename'} \\ \text{"filename"} \end{array} \right\} [ , \text{ACCT} = \text{acct} ]$

$\left[ , \text{QUEUE} = \left\{ \begin{array}{l} \text{LOG} \\ \text{PRINT} \\ \text{PUNCH} \\ \text{RDR} \end{array} \right\} \right] \left[ , \text{ALL} = \left\{ \begin{array}{l} \text{YES} \\ \text{NO} \end{array} \right\} \right] \left[ , \text{COPIES} = \left\{ \begin{array}{l} n \\ 1 \end{array} \right\} \right] \left[ , \text{SKIP} = \left\{ \begin{array}{l} n \\ 0 \end{array} \right\} \right] \Delta [ \text{DIRECT} ] [ , \text{WAIT} ]$

### Punching a Tape File

$\text{PUNCH}\Delta \left[ \text{FILENAME} = \left\{ \begin{array}{l} \text{filename} \\ \text{'filename'} \\ \text{"filename"} \end{array} \right\} \right] [ , \text{RDPASS} = \text{password} ] , \text{VSN} = \text{volume} , \text{DEVICE} = \left\{ \begin{array}{l} \text{did} \\ \text{TAPE} \end{array} \right\}$

$\left[ , \text{BKNO} = \left\{ \begin{array}{l} \text{YES} \\ \text{NO} \end{array} \right\} \right] \Delta [ \text{DIRECT} ] [ , \text{WAIT} ]$

## PRINT Command

The PRINT command enables you to make a printed copy of a library file or module, a spool file, a MIRAM file, a tape file, or a unit record file. PRINT can run as a background job to allow you to use the workstation for other functions while the file is being printed.

### Printing a SAT or MIRAM Library Module

```
PRINTMODULE=modulename [ ,TYPE= { module-type } ] ,FILENAME= { filename
                        'filename'
                        "filename" }
[ ,RDPASS=password ] ,VSN=volume [ ,COPIES= { n } ] Δ[DIRECT][ ,NUMBER][ ,HEX]
[ ,WAIT][ ,FORM=formname ]
```

### Printing a MIRAM Data File

```
PRINTFILENAME= { filename
                'filename'
                "filename" } [ ,RDPASS=password ] ,VSN=volume [ ,KEYNO= { n } ]
[ ,COPIES= { n } ] [ ,DEVICE= { did
                               DISKETTE
                               DISK } ] Δ[DIRECT][ ,NUMBER][ ,HEX][ ,WAIT]
[ ,FORM=formname ]
```

### Printing a Tape File

```
PRINTA [ FILENAME= { filename
                  'filename'
                  "filename" } ] [ ,RDPASS=password ] ,VSN=volume ,DEVICE= { did
                                                                              TAPE }
[ ,BKNO= { YES
          NO } ] Δ[DIRECT][ ,NUMBER][ ,HEX][ ,WAIT][ ,FORM=formname ]
```

### ERASE Command

The ERASE command permits you to delete library and data files, as well as library modules. Before executing takes place, the ERASE command double-checks with you to make certain you want to erase a file. This guards against inadvertent erasures.

#### Erasing a Library Module

$$\text{ERASEMODULE}=\text{modulename} \left[ \text{,TYPE}=\left\{ \begin{array}{l} \text{module-type} \\ \$ \end{array} \right\} \right] \text{,FILENAME}=\left\{ \begin{array}{l} \text{filename} \\ \text{'filename'} \\ \text{"filename"} \end{array} \right\}$$

[,WRPASS=password],VSN=volume

#### Erasing Library and MIRAM Data Files

$$\text{ERASEFILENAME}=\left\{ \begin{array}{l} \text{filename} \\ \text{'filename'} \\ \text{"filename"} \end{array} \right\} \text{[,WRPASS=password],VSN=volume}$$

### FLUSH Command

The FLUSH comand permits you to discard system messages that are queued for delivery at your workstation. You can discard all queued messages or only those with a specified message prefix.

#### Format

$$\text{FLUSH} \left\{ \begin{array}{l} \text{*ALL} \\ \text{msg-prefix} \end{array} \right\}$$

## Utility Commands

---

### DEFKEY DISPLAY Command

The DEFKEY DISPLAY command displays your function key and MESSAGE WAITING key assignments.

DEFKEY^DISPLAY

### DISPLAY LOG Command

The DISPLAY LOG command gives you a 1-line report on the status of your workstation log file. The display shows the number of workstation lines used since you logged on.

DISPLAY^LOG

### EDT Command

The EDT command permits you to initialize the General Editor.

EDT^initial command]

### ENTER Command

The ENTER command permits you to enter a series of workstation commands as a batch processing job. Sessions may be entered from a card reader, tape, diskette, spooled file, or library file. Output from the job is always directed to the printer. You may enter any workstation command except those directly affecting the workstation device, such as SCREEN.

#### To Run a Command Stream from a Library File

ENTER^MODULE=modulename [ ,TYPE= { module-type } ] ,FILENAME= { filename  
'filename'  
"filename" }

[ ,RDPASS=password],VSN=volume

## Utility Commands

---

### Copying Tape Files

`COPYΔ`  $\left[ \begin{array}{l} \text{FILENAME=} \\ \text{'filename'} \\ \text{"filename"} \end{array} \right] \left[ , \text{RDPASS}=\text{password} \right], \text{VSN}=\text{volume}, \text{DEVICE}=\left\{ \begin{array}{l} \text{did} \\ \text{TAPE} \end{array} \right\}$

$\left[ , \text{BKN}=\left\{ \begin{array}{l} \text{YES} \\ \text{NO} \end{array} \right\} \right] \Delta \text{TOA}$

$\left[ \begin{array}{l} \text{FILENAME=} \\ \text{'filename'} \\ \text{"filename"} \end{array} \right] \left[ , \text{WRPASS}=\text{password} \right], \text{VSN}=\text{volume}, \text{DEVICE}=\left\{ \begin{array}{l} \text{did} \\ \text{TAPE} \end{array} \right\}$

$\left[ , \text{INIT}=\left\{ \begin{array}{l} \text{YES} \\ \text{NO} \end{array} \right\} \right] \left[ , \text{EXTEND}=\left\{ \begin{array}{l} \text{YES} \\ \text{NO} \end{array} \right\} \right]$

$\left[ , \text{BFSZ}=\text{n} \right] \left[ , \text{BKN}=\left\{ \begin{array}{l} \text{YES} \\ \text{NO} \end{array} \right\} \right] \left[ , \text{RCFM}=\left\{ \begin{array}{l} \text{FIXUNB} \\ \text{FIXBLK} \\ \text{VARUNB} \\ \text{VARBLK} \\ \text{UNDEF} \end{array} \right\} \right] \left[ , \text{RCSZ}=\text{n} \right] \Delta \left[ \text{NUMBER} \right] \left[ , \text{HEX} \right] \left[ , \text{WAIT} \right]$

## Utility Commands

### Copying MIRAM Data Files

$\text{COPY}\Delta\text{FILENAME} = \left\{ \begin{array}{l} \text{filename} \\ \text{'filename'} \\ \text{"filename"} \end{array} \right\} [ , \text{RDPASS} = \text{password} ] , \text{VSN} = \text{volume} \left[ , \text{KEYNO} = \left\{ \begin{array}{l} n \\ 0 \end{array} \right\} \right]$

$\left[ , \text{DEVICE} = \left\{ \begin{array}{l} \text{did} \\ \text{DISK} \\ \text{DISKETTE} \end{array} \right\} \right] \Delta\text{TO}\Delta\text{FILENAME} = \left\{ \begin{array}{l} \text{filename} \\ \text{'filename'} \\ \text{"filename"} \end{array} \right\} [ , \text{WRPASS} = \text{password} ]$

$, \text{VSN} = \text{volume} \left[ , \text{CONTIG} = \left\{ \begin{array}{l} \text{YES} \\ \text{NO} \end{array} \right\} \right] \left[ , \text{INC} = \left\{ \begin{array}{l} n \\ 1 \end{array} \right\} \right] \left[ , \text{KEYNO} = \left\{ \begin{array}{l} n \\ 0 \end{array} \right\} \right]$

$\left[ , \text{KEY}i = \left\{ \begin{array}{l} n:m \\ \left( n:m, \left\{ \begin{array}{l} \text{DUP} \\ \text{NDUP} \end{array} \right\}, \left\{ \begin{array}{l} \text{CHG} \\ \text{NCHG} \end{array} \right\} \right) \end{array} \right\} \right]$

$[ , \text{SIZE} = n ] \left[ , \text{INIT} = \left\{ \begin{array}{l} \text{YES} \\ \text{NO} \end{array} \right\} \right] \left[ , \text{RCB} = \left\{ \begin{array}{l} \text{YES} \\ \text{NO} \end{array} \right\} \right] \left[ , \text{RCFM} = \left\{ \begin{array}{l} \text{FIX} \\ \text{VAR} \end{array} \right\} \right]$

$[ , \text{RCSZ} = n ] \left[ , \text{EXTEND} = \left\{ \begin{array}{l} \text{YES} \\ \text{NO} \end{array} \right\} \right] [ , \text{BFSZ} = n ] \left[ , \text{SCSZ} = \left\{ \begin{array}{l} n \\ 256 \end{array} \right\} \right]$

$\left[ , \text{DEVICE} = \left\{ \begin{array}{l} \text{did} \\ \text{DISK} \\ \text{DISKETTE} \end{array} \right\} \right] \Delta [ \text{NUMBER} ] [ , \text{HEX} ] [ , \text{WAIT} ]$

## Utility Commands

---

### ASK Command

The ASK command enables you to ask questions of other workstation users or the system operator. The command displays your question to the other user, accepts the reply, and returns the reply to you.

`ASKΔ[user-id], 'text'`

### BRKPT LOG Command

The BRKPT LOG command enables you to close a workstation log file and make it available to the output writer for printing before you log off the workstation.

`BRKPTΔLOG`  $\left[ \text{,OUT=} \begin{cases} \text{T A P E} \\ \text{D I S K} \\ \text{D I S K E T T E} \end{cases} \right] \left[ \text{,HOLD} \right]$

### COMMENT Command

The COMMENT command enables you to put a comment in a library module header or replace an existing comment on a library module header. Comments may be up to 30 characters long.

`COMMENTΔMODULE=modulename`  $\left[ \text{,TYPE=} \begin{cases} \text{module-type} \\ \$ \end{cases} \right] \text{,FILENAME=} \begin{cases} \text{filename} \\ \text{'filename'} \\ \text{"filename"} \end{cases}$   
`[,RDPASS=password][,WRPASS=password],VSN=volumeΔtext`

### RP Command

The RP command enables you to manually load an output writer to print to an auxiliary printer. *To use auxiliary printers, you must have generated your system to use auxiliary printers and directed your print files to an auxiliary printer prior to the time your system prints your print files.* RP alone will not direct printing to an auxiliary printer.

```
RP[function-code][,ACCT=acctno][,BNUMB=binary jobno]
[,CART=cartridge-name][,FILE=filename]
[,FORM=formname][,JOB=jobname]
```

## Controlling Spooling

---

### CHANGE SPL Command

The CHANGE SPL command permits you to change the device type and/or the number of copies of a spool file.

CHANGESPL, { ALL  
LOG  
PRINT }

[,modifier-1 ... modifier-n]

[,COPIES=nnn]

[ [ ,DVC= { 770  
776  
PPC  
ANY  
CLASS1  
CLASS2  
CLASS3 } ]  
[ ,DVC= AUX, ID= { \*  
user-id } ]  
[ ,ID= { \*  
user-id } ] ] ]

## BEGIN SPL Command

The BEGIN SPL command releases spool files held by a HOLD SPL command. Entering this command also loads an output writer to print the file as soon as a printer becomes available.

```
BEGINASPL, { ALL  
            LOG  
            PRINT  
            PUNCH  
            RDR } [,ACCT=acctno][,BNUMB=binary jobno]
```

```
[,CART=cartridge-name] [ ,DEV= { 770  
                                776  
                                789 } ]
```

```
[,FILE=filename][,FORM=formname][,JOB=jobname][,STEP=stepno] [ ,OUT= { did  
                                NO } ]
```

**Note:** *Global control users will begin every qualifying spool file in the system, not just the spool files initiated under their user-ids.*

## Controlling Spooling

---

### DISPLAY Command

#### Obtaining Information about Active Spool Files

The DISPLAY ACT command permits you to obtain information about the spool files being created.

```
DISPLAYACT [ , { ALL  
PRINT  
PUNCH } ] [ , ACCT=acctno ]
```

```
[ , CART=cartridge-name ] [ , DEV= { 770  
776  
789 } ]
```

```
[ , FILE=filename ] [ , FORM=formname ] [ , JOB=jobname ] [ , STEP=stepno ]
```

#### Obtaining Information about Completed Spool Files

The DISPLAY SPL command permits you to obtain information about the completed spool files.

```
DISPLAYSPL [ , { ALL  
LOG  
PRINT  
PUNCH  
RDR } ] [ , ACCT=acctno ]
```

```
[ , CART=cartridge-name ] [ , DEV= { 770  
776  
789 } ]
```

```
[ , FILE=filename ] [ , FORM=formname ] [ , JOB=jobname ] [ , STEP=stepno ]
```



### CONNECT Command

The **CONNECT** command is used to connect the **WORKSTATION** mode of your workstation to a job running on the system. You connect to a job at any time while it is running.

```
CONNECTAjob[,filename]
```

### DISPLAY JS Command

The **DISPLAY JS** command enables you to obtain information about jobs. You receive information about the status of a job, whether it is on a scheduling queue, or executing, and, if not executing, why it is not. You can also display information on *all* jobs initiated or running under your user-id.

```
DISPLAYAJS[,jobname][,ALL]
```

### FREE Command

The **FREE** command permits you to manually disconnect a workstation from a job. When this command is issued, the **WORKSTATION** mode of your workstation is disconnected from the program to which it was connected.

```
FREE
```

### PAUSE Command

The **PAUSE** command permits you to suspend processing of a job. The job may be suspended immediately or when a specified job step is completed. Any job suspended with the the job-step-no option cannot be cancelled until the job is resumed. You may enter the **PAUSE** command at any time. Use the **GO** command to restart job processing.

```
PAUSEAjobname[,job-step-no]
```



## HOLD Command

The HOLD command permits you to defer the scheduling of jobs. You may hold individual jobs initiated under your user-id, all your jobs present in a particular queue, or all of your jobs (all those initiated under your user-id). You may hold jobs other than those initiated under your user-id if you have been given global control privileges by the system administrator.

### Holding All Jobs or All Jobs on a Particular Job Queue

HOLDΔJBQ ,  $\left( \begin{array}{c} \text{A} \\ \text{H} \\ \text{N} \\ \text{P} \\ \text{L} \end{array} \right)$

### Holding an Individual Job

HOLDΔjobname

## CHANGE Command

The CHANGE command permits you to change the scheduling queue in which a job is residing.

CHANGEΔjobname ,  $\left( \begin{array}{c} \text{H} \\ \text{N} \\ \text{P} \\ \text{L} \end{array} \right)$

## EXECUTE Command

The EXECUTE command enables you to run programs in an interactive job environment. To use EXECUTE, you must first run the super-set job control stream that incorporates the program you want to use.

`EXECUTEΔprogram-name`

## ULD Command

The upline dump command enables a UTS 400 user to get a dump of the terminal's memory. A user can choose to print or save the dump file.

`ULDΔ, filename, vsn, SIZE=nΔ`  $\left[ \begin{array}{l} \left\{ \begin{array}{l} \text{PRINT} \\ \text{NOPRINT} \end{array} \right\} \Delta \left\{ \begin{array}{l} \text{SCRATCH} \\ \text{SAVE} \end{array} \right\} \end{array} \right]$

## DLOAD Command

The DLOAD command allows a UTS 400 terminal user or a UTS 40/40D workstation user to downline load a program to terminal memory. Any program you load using the DLOAD command must reside in the \$Y\$LOD library.

`DLOADΔ`  $\left\{ \begin{array}{l} \text{program-name} \\ \text{/OFFLINE} \end{array} \right\}$

## SI Command

The SI command enables you to run jobs previously saved in the expanded state. The SI command is used when embedded data in the job control stream is to be replaced.

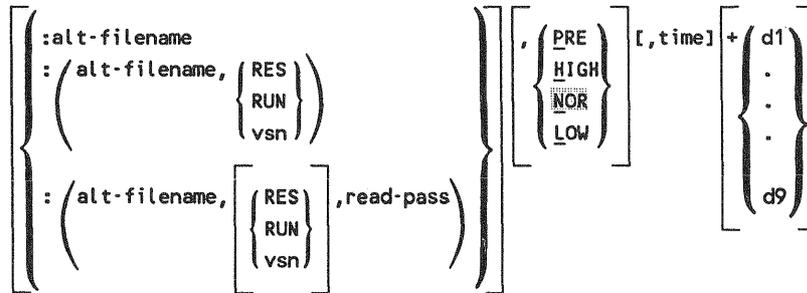
SI { ([did],label) } Δ[ jobname][ (new-name) ]  
 { (RDR,label) }

$$\left[ \begin{array}{l} :alt-filename \\ : (alt-filename, \begin{array}{l} \{ RES \} \\ RUN \\ vsn \end{array} ) \\ : (alt-filename, \begin{array}{l} \{ RES \} \\ RUN \\ vsn \end{array} ), read-pass \end{array} \right] \left[ \begin{array}{l} , \begin{array}{l} \{ PRE \} \\ \{ HIGH \} \\ \{ NOR \} \\ \{ LOW \} \end{array} \end{array} \right] [ , time ] + \left[ \begin{array}{l} d1 \\ . \\ . \\ . \\ d9 \end{array} \right]$$

## RUN Command

The RUN command enables you to execute user jobs from the workstation. The command causes the job control stream associated with your job to be read, expanded, and scheduled for execution. The RUN command is used when an input device, either a data set label diskette or the input spool file (RDR), is required.

**RUN** { [(did), label] } Δ [jobname] [(new-name)]  
 { (RDR, label) }



[, key-1=val-1, ..., key-n=val-n]

## LOGON Procedure

---

To log off the workstation, enter the LOGOFF command:

```
LOGOFF
```

There are no parameters entered with this command.

**Note:** *The BULLETIN and LOG keyword defaults can be changed during SYSGEN (see the Models 3-6 and 8-20 Installation Guide, (UP-8839) or the Model 7E Installation Guide, (7002 3858)) or through a SET IS command.*

## Standard Terminal Dialog for ICAM Terminals

If you are using a remote terminal as a workstation, you must connect the terminal to ICAM (Integrated Communications Access Method) before you log on. You connect the terminal to ICAM with the following command:

```
$$SONΔxxxxyyyy
```

### Explanation

xxxx

Specifies the logical name of your terminal.

yyyy

Specifies the logical name of the program you are signing on to use.

After entering the \$\$SON command, you will receive a message informing you whether your sign on attempt was successful.

After finishing your workstation session and logging off, you terminate your communications link with ICAM by issuing the following command:

```
$$SOFF
```

There are no parameters entered with this command.

## Entering Information on a Workstation

---

### Example

1. COPY MODULE=MYMODULE,FILENAME=MYFILE,VSN=REL071 TO MODULE=NEWMODUL,  
FILENAME--
2. 11?USERID IS93 ENTER CONTINUATION?
3. 11 NEWFILE,VSN=NEWVOL,SIZE=10,CONTIG=NO

1. After keying in almost a full line, add a dash as the last character on the line. Press the XMIT key.
2. The system asks you for a continuation.
3. To answer, key in the system message number (11), enter a space, and enter the rest of the command. Press the XMIT key.

## What You Can Control

If you are a normal user, you can control and display all jobs (including spooled output) run under your user-id. If you have been given global control privileges by the system administrator, you can control and display *all* jobs. If you have been given global status privileges, you can display *all* jobs.

## Contents

---

### Section 9. RPG II Editor and COBOL Editor

RPG II Editor .....	9-1
COBOL Editor .....	9-3

### Section 10. DDP Commands

### Section 11. Interactive Terminals

### Section 12. Function Key Summary

### User Comments Form

## Contents

---

### Section 4. Controlling Jobs

CANCEL Command .....	4-1
CJ Command .....	4-1
CONNECT Command .....	4-2
DISPLAY JS Command .....	4-2
FREE Command .....	4-2
PAUSE Command .....	4-2
GO Command .....	4-3
STOP Command .....	4-3

### Section 5. Controlling Spooling

DISPLAY Command .....	5-2
HOLD SPL Command .....	5-3
BEGIN SPL Command .....	5-4
DELETE SPL Command .....	5-5
CHANGE SPL Command .....	5-6
BRKPT Command .....	5-7
PR/PU Command .....	5-7
RP Command .....	5-8

### Section 6. Utility Commands

ALLOCATE Command .....	6-1
ASK Command .....	6-2
BRKPT LOG Command .....	6-2
COMMENT Command .....	6-2
COPY Command .....	6-3
DEFKEY Command .....	6-7
DEFKEY (Delete) Command .....	6-7
DEFKEY DISPLAY Command .....	6-8
DISPLAY LOG Command .....	6-8
EDT Command .....	6-8
ENTER Command .....	6-8
ERASE Command .....	6-10
FLUSH Command .....	6-10

## About This Guide

---

- Default values are values automatically generated by the system when you do not specify a value for a parameter. Default values are shown shaded in each command format:

`,COPIES= { n }`

- Spaces are indicated by a delta ( $\Delta$ ) symbol:

`COPY $\Delta$ MODULE`

## Related Product Information

The following Unisys documents may be useful in understanding and implementing OS/3's interactive services.

*Note:* Throughout this guide, when we refer you to another manual, use the current version that applies to the software level at your site.

### ***Models 3-6 and 8-20 Installation Guide (UP-8839)***

This manual describes how to install the models 3-6 and 8-20 data processing systems and explains the software aids available to help the site administrator complete the installation process.

### ***Model 7E Installation Guide (7002 3858)***

This manual describes how to install the model 7E data processing system and explains the software aids available to help the site administrator complete the installation process.

### ***Interactive Services Operating Guide (UP-9972)***

This guide describes how to communicate interactively through a local workstation or remote terminal with the operating system.

### ***Report Program Generator II (RPG II) Operations Guide (UP-9981)***

This guide describes how the RPG II programmer can use the RPG II editor to create and maintain RPG II source programs interactively from a workstation.

### ***COBOL Editor (COBEDT) Programming Guide (UP-9974)***

This guide describes how to use the COBOL editor so the programmer can interactively create and modify COBOL source programs.

### ***General Editor (EDT) Operating Guide (UP-9976)***

This guide describes how to use the general editor (EDT) to create and update library modules and data files interactively from a workstation, and to copy and concatenate files.

## Audience

The intended audience is the system administrator and the workstation operator.

## How to Use This Guide

Read the entire guide to familiarize yourself with the basic concepts it presents; then use it for reference as needed.

## Organization

This guide is divided into 12 sections.

### **Section 1. Entering Information on a Workstation**

Explains how to enter commands and respond to messages at the System 80 workstation.

### **Section 2. LOGON Procedure**

Explains how LOGON to System 80 so you can begin your workstation session.

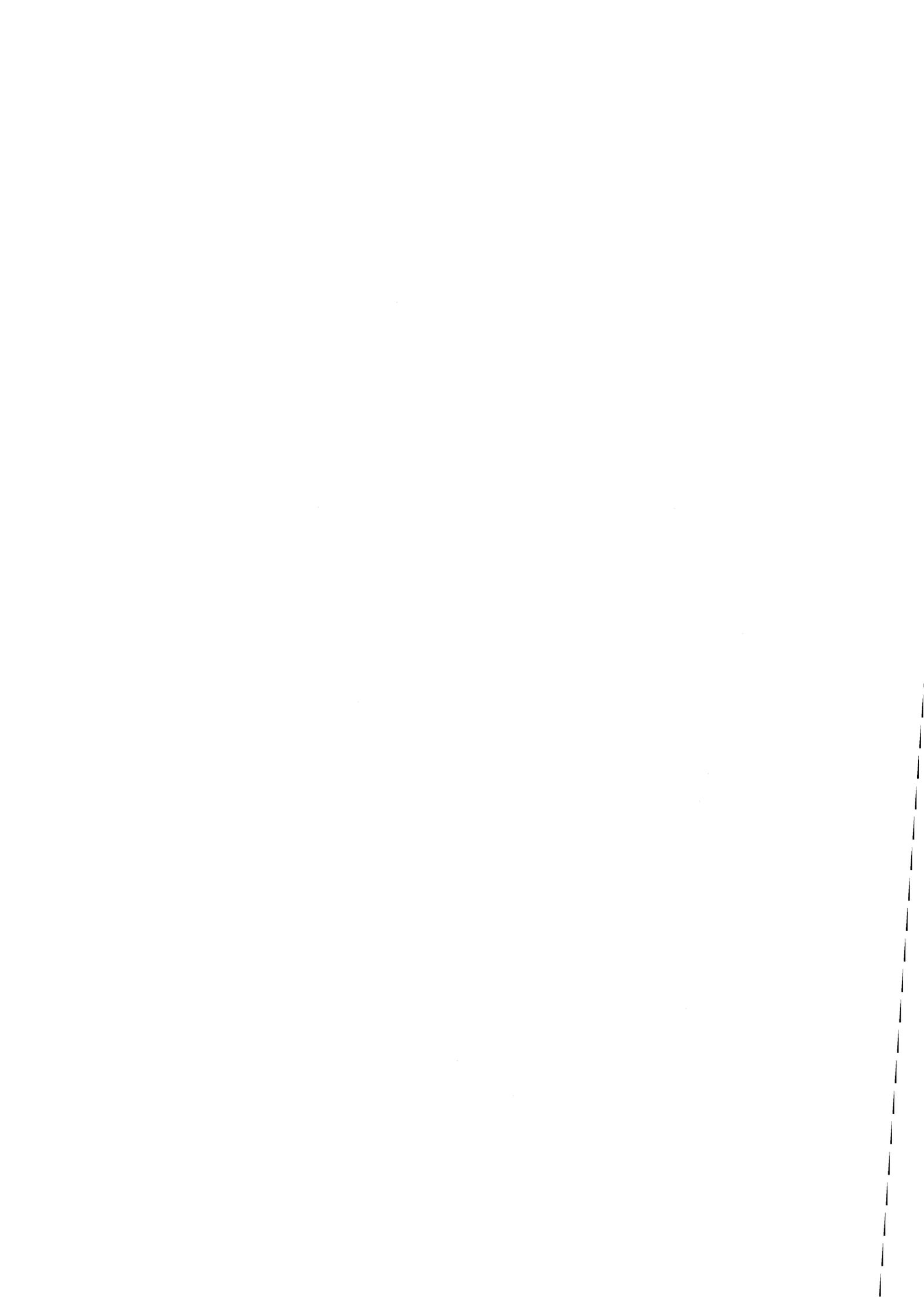
### **Section 3. Running Jobs and Changing Job Scheduling**

Describes the commands used to run jobs from the workstation and change the scheduling for those jobs.

### **Section 4. Controlling Jobs**

Describes the commands that permit your workstation to act as a miniature system console to control those jobs initiated or running under your user-id. You can *only* control jobs initiated or running under your user-id unless you have been global control privileges by the system administrator.





NO WARRANTIES OF ANY NATURE ARE EXTENDED BY THE DOCUMENT. Any product and related material disclosed herein are only furnished pursuant and subject to the terms and conditions of a duly executed Program Product License or Agreement to purchase or lease equipment. The only warranties made by Unisys, if any, with respect to the products described in this document are set forth in such License or Agreement. Unisys cannot accept any financial or other responsibility that may be the result of your use of the information in this document or software material, including direct, indirect, special or consequential damages.

You should be very careful to ensure that the use of this information and/or software material complies with the laws, rules, and regulations of the jurisdictions with respect to which it is used.

The information contained herein is subject to change without notice. Revisions may be issued to advise of such changes and/or additions.

Correspondence regarding this publication should be forwarded, using the User Comments form at the back of this manual or remarks addressed directly to Unisys Corporation, to E/MSG Product Information, P.O. Box 500, M.S. E5-114, Blue Bell, PA 19424 U.S.A.



NO WARRANTIES OF ANY NATURE ARE EXTENDED BY THIS DOCUMENT. Any product and related material disclosed herein are only furnished pursuant and subject to the terms and conditions of a duly executed Program Product License or Agreement to purchase or lease equipment. The only warranties made by Unisys, if any, with respect to the products described in this document are set forth in such License or Agreement. Unisys cannot accept financial or other responsibility that may be the result of your use of the information in this document or software material, including direct, indirect, special, or consequential damages.

You should be very careful to ensure that the use of this information and/or software material complies with the laws, rules, and regulations of the jurisdictions with respect to which it is used.

The information contained herein is subject to change without notice. Revisions may be issued to advise of such changes and/or additions.

Correspondence regarding this publication should be forwarded to Unisys Corporation either by using the Business Reply Mail form at the back of this manual or by addressing remarks directly to Unisys Corporation, OS/3 Systems Product Information Development, P.O. Box 500, Mail Station E5-114, Blue Bell, Pennsylvania, 19424, U.S.A.