

UNISYS

**V Series
Systems**

**Operations
Guide**

**Volume 2: System
Commands**

Copyright © 1990 Unisys Corporation
All rights reserved.

Unisys is a registered trademark of Unisys Corporation

Release 3.1

Priced Item

May 1990
Distribution Code SD
Printed in U S America
4127 4994-000

The names, places, and/or events used in this publication are not intended to correspond to any individual, group, or association existing, living, or otherwise. Any similarity or likeness of the names, places, and/or events with the names of any individual, living or otherwise, or that of any group or association is purely coincidental and unintentional.

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 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, Technical Publications, 460 Sierra Madre Villa, Pasadena, CA 91109.

Page Status

Page	Issue
v through xxii	Original
1-1 through 1-14	Original
2-1 through 2-151	Original
2-152	Blank
2-153 through 2-185	Original
2-186	Blank
2-187 through 2-199	Original
2-200	Blank
2-201 through 2-305	Original
2-306	Blank
2-307 through 2-439	Original
2-440	Blank
A-1 through A-10	Original
B-1 through B-11	Original
B-12	Blank
Glossary-1 through Glossary-34	Original
Index-1 through 47	Original
Index-48	Blank

()

()

()

About This Guide

The *V-Series Systems Operations Guide Volume 2: System Commands* describes the commands that enable you to communicate with the Master Control Program (MCP) for the V Series computer systems. The various system commands allow inquiry or change of system, program, and device status; initiation and modification of system utilities; and initiation of programs from the user program library.

Purpose

The *V-Series Systems Operations Guide Volume 2: System Commands* lists the system commands in alphabetical order and provides detailed information about the use and syntax for each command.

Scope

The system commands documented in this guide operate under MCP/VS 3.1 on any V Series system that has the correct firmware (microcode) and/or hardware.

- V300 systems require the correct firmware from floppy disk.
- V500 systems require the correct firmware and hardware (EPROMs) in the Data Transfer Module (DTM) and the System Maintenance Controller (SMC).

Audience

This manual provides information about system commands for data management personnel, systems analysts, and systems operators.

Prerequisites

Some familiarity with data processing and Unisys V Series systems is required. This guide provides some basic information for the first-time or occasional user as well as reference material for the experienced user.

How to Use This Document

The system commands are listed alphabetically in Section 2. For first-time or occasional users, Section 1 provides information on basic constructs and on

entering commands. A glossary is provided which defines all acronyms and unique terms used in this manual.

Organization

This guide consists of two sections, two appendixes, a glossary, and an index.

Section 1. Introduction to the System Commands

This section discusses the different ways system commands are entered into the system to communicate with the Master Control Program (MCP); the distinction between keyboard input messages and MCP control instructions; some details that pertain to the commands in general; and command restrictions.

Section 2. The System Commands

This alphabetical listing of the system commands describes and explains the syntax of each command.

Appendix A. Understanding Railroad Diagrams

For those who are unfamiliar with railroad syntax diagrams, this appendix provides information on how to follow these diagrams.

Appendix B. Commands for User-Coded Utilities

The ADD, CHECK, DUMP, LOAD and UNLOAD commands invoke the LOADMP and PACKUP utilities. The functions of these utilities have been replaced by the SYSTEM/COPY utility. The PB command invokes the PBTOUT or PBDOUT utilities, which have been replaced by PBDPRN. Unisys no longer provides the LOADMP, PACKUP, PBDOUT, and PBTOUT utilities, but user-written versions can be invoked by the applicable commands. This appendix contains the syntax of these commands.

Glossary

The glossary provides definitions of terms and concepts used throughout the V Series Systems Operation Guide Volumes 1 through 4.

Related Product Information

V Series BNA Installation and Operations Guide (5022999)

This guide discusses the V Series Network Architecture (BNA) software, which links V Series computer systems together and to other Unisys computer systems in a BNA data communications/distributed processing network configuration.

B 2000/B 3000/B 4000/V Series GEMCOS Implementation and Programming Reference Manual (5026313)

This manual describes the features, methods of application, and utilities of the Generalized Message Control System (GEMCOS).

B 2000/B 3000/B 4000/V Series PROPS Operations Services Operations Guide
(5025455)

This guide describes how to use the Programmer's Productivity System (PROPS) services. These services benefit both programmers and operators in providing electronic mail capabilities, system log search functions, and the ability to initiate a job and monitor all ODT messages relating to the job on the terminal.

B 2000/B 3000/B 4000/V Series Remote Job Entry (RJE) Operations Guide
(5026743)

This guide explains how to use Remote Job Entry (RJE), which is a software package that allows jobs, data, and control commands to be sent from a remote site to a central system, and that allows output of data from the central system to be sent to remote peripherals.

B 2000/B 3000/B 4000/V Series WFL Compiler Programming Reference Manual (5023229)

This manual provides information on the Work Flow Language (WFL). WFL is a Unisys language used for constructing jobs that compile and run programs on Unisys computer systems. WFL includes variables, expressions, and flow-of-control statements that offer a range of task control capabilities.

V Series CANDE Installation and Operations Guide (5030141)

This guide discusses the Command And Edit (CANDE) timesharing system, which provides interactive generalized file preparation and updating capabilities, and task control from your terminal.

V Series COBOL ANSI-74 Compiler Programming Reference Manual, Volume 1: Basic Implementation (4127 4945-000)

This manual discusses the basic implementation of the COBOL-74 (Common Business Oriented Language ANSI-74 Standard) language on Unisys V Series systems.

V Series COBOL ANSI-74 Compiler Programming Reference Manual, Volume 2: Product Interfaces (4127 4952-000)

This manual discusses the Unisys extensions to the COBOL-74 (Common Business Oriented Language ANSI-74 Standard) language that provide access to various V Series products.

V Series DMSII Operations Reference Manual Volume 1: Database Creation
(5031412)

Volume 1 contains information on how to set up a database with the Data and Structure Definition Language (DASDL). The manual includes sections about creating a database, creating and compiling a DASDL source file, logical data structures within Data Management System II (DMSII), DASDL syntax, and examples of DASDL specifications and DASDL database definitions.

V Series DMSII Operations Reference Manual Volume 2: Database Administration (5031404)

Volume 2 contains information the Database Administrator can use to customize or streamline the DMSII product for a particular environment. The manual includes sections about updating and reorganizing a database, audit and recovery functions, production data programs, an MCP interface to DMSII, database dumping and recovery, the database program, exception condition categories, and output messages.

V Series DMSII Operations Reference Manual Volume 3: Utilities (5031420)

Volume 3 contains all of the information for the following DMSII utilities: DMSUTL, DBPDMP, DMSTAT, DMSLST, DMSCTL, DMSDAP, DDFLIS, DMSAUD, DMSDLR. Each section covers one utility and the last section covers all utility messages.

V Series MCP/VS Programming Reference Manual (4127 0026-000)

This document provides advanced-level information a programmer/analyst might need to know about MCP/VS, such as how to read a dump or write a user-coded utility program.

V Series Program Interfaces Programming Reference Manual (4127 4531-000)

This manual describes the various programmatic interface areas available with MCP/VS.

V Series System Software Security Installation and Operations Guide (4127 4820-000)

This guide explains how to install and use the system security facilities for the Unisys V Series systems to control access to the system and to files.

V Series System Software Logging Operations Reference Manual (4127 4812-000)

This manual describes the components of the system ODT log (SLOG), Run log (RLOG), and Maintenance log (MLOG). These logs are produced by the Master Control Program (MCP) and can be analyzed by system or user-written utilities.

V Series Systems Operations Guide Volume 1: Installation (4127 4804-000)

This volume of the Systems Operations Guide contains introductory material about the MCP/VS operating system and details the procedures and options available for installing and initializing MCP/VS on V Series systems.

V Series Systems Operations Guide Volume 3: System Utilities (4127 0000-000)

This volume of the Systems Operations Guide describes the MCP/VS utilities, which act as extensions of the Master Control Program (MCP) for such functions as converting files from one medium to another, initializing diskpack units, or creating pseudo card decks.

V Series Systems Operations Guide Volume 4: System Messages and Recovery
(4127 0018-000)

This volume of the Systems Operations Guide is a reference to the system messages that appear on the Operator Display Terminal (ODT) to either inform you of a process taking place or of an error condition. Recovery procedures are also included.

()

()

()

Contents

About This Document	v
Section 1. Introduction to the System Commands	
System Command Overview	1-1
Keyboard Input Messages Versus MCP Control	
Instructions	1-1
Entering Commands through the ODT	1-2
MCP Response to ODT Input	1-3
Entering Commands through a Programmatic ZIP	
or ZIPSPO	1-4
Entering Commands through a ZIPSPO	1-5
Entering Commands through a SPOMESSAGE BCT	1-5
Entering Commands through Punched Cards	1-5
Entering Commands through CANDE, PROPS, or WFL	1-6
Entering Commands through RJE	1-7
Entering Commands through BNA	1-7
Command Details	1-7
File Identifiers	1-8
Special Characters	1-10
Command Restrictions	1-11
Access Level	1-12
Section 2. The System Commands	
AC (Purge and Place Tape Number in Label)	2-2
AD (Specify ODT/OCS Screen Parameters)	2-4
ADD (Copy Files from Library Tape to Disk or Diskpack) ...	2-11
AFTER (Hold Program)	2-12
AJ (Display Active Jobs)	2-14
ALLOCATE (Create Disk Space and Directory Entries)	2-17
ALTER (Change System Parameters)	2-21
BACKUP Option	2-22
EXCHANGE Option	2-24
FAMILY Option	2-25
Purging a Diskpack Family (PURGE)	2-25
Rebuilding a Diskpack Family (REBUILD)	2-26
Renaming a Diskpack Family (RENAME)	2-27
FILE Option	2-28
PACK Option	2-29
PROC Option	2-31

Setting the Processor Online	2-31
Setting the Processor Offline	2-32
SYSTEM Option	2-33
TRAKBUFFER Option	2-34
VCS Option	2-35
AT (Initiate BNA Job Transfer)	2-36
AT (Pass Message to Remote Host)	2-38
AX (Pre-Respond to an ACCEPT)	2-39
AX (Respond to an ACCEPT)	2-41
BCL (Mark Beginning of Card Data File)	2-43
BEGINUSER (Mark Beginning of Job Stream)	2-45
BF, BFP (List Backup Files on Disk or Pack)	2-47
BFRF (Search, List, Remove Backup Files)	2-49
Executing the BFRF Intrinsic	2-55
BINARY (Read Binary Card Deck)	2-57
BO (Blackout Access Code)	2-58
BT (Begin TRAK)	2-59
BUFFER (Set MCS Result Queue Size)	2-63
CA (List Port File Candidates)	2-65
CC (Enter Control Instructions)	2-66
CD, CDP (List Inactive Pseudo Decks)	2-68
CHANGE (Change File ID)	2-70
CHARGE (Insert Charge Number)	2-74
CHECK (Verify Library Tape)	2-76
CK (Test Peripheral Device)	2-77
CN (Display Tape Number)	2-78
COMPARE (Compare Files on Different Media)	2-79
COMPILE (Compile Program)	2-80
Getting Programs in the Correct Format	2-84
COPY (Invoke File Transfer Utility)	2-86
COPYADD (Transfer Files)	2-99
CV, CVP (Convert Punch Backup File to Pseudo Card Deck)	
.....	2-100
DA, DAP (Deactivate Pseudo Reader)	2-103
DATA/DATAB (Mark Beginning of Card Data File)	2-105
DBCA (Clear SSP Address Contention)	2-107
DBCL (Cancel DLP)	2-109
DBDR (Produce SSP Content Report)	2-110
DBIC (Initialize Controller Parameters)	2-111
DBMM (Set/Reset Maintenance Mode)	2-112
DBSR (Produce Block Lockout Table Report)	2-114
DBUA (Unlock Address)	2-115
DBUL (Clear Locked Files on Shared Systems)	2-117
DC (Display Compilation Data)	2-118
Displaying ICM Compilation Data	2-119
DD (Delete DLP)	2-120
DEBUG (Attach Program to DEBUG Session)	2-121
DIR (List File Directory)	2-125
DISK (Add a Disk)	2-128
DISPLAY/WHATS (Display Information)	2-131
Displaying a Program Code File, MCP Code File or	
MCP Intrinsic Code File	2-131

Displaying a Disk or Diskpack File	2-133
Displaying an Absolute Disk Sector	2-134
DL (Delete Peripheral Devices)	2-135
DLP (Add DLPs)	2-137
DM (Dump and Continue)	2-142
Memory Dumps of a User Task	2-142
Memory Dumps of the Operating System	2-142
DP (Dump and Discontinue)	2-144
DQ (Display System Messages)	2-145
DR (Date Reset)	2-147
DS (Discontinue Program)	2-148
DUMP (Copy Files to Library Tape)	2-150
DX (Display DLPs Attached to Exchange)	2-151
EBCDIC (Mark Beginning of Card Data File)	2-153
ED (Enter DEBUG Session)	2-155
END/ENDCTL (Mark End of Card File or Control	
Instructions)	2-156
ENDAT (End Job Transfer)	2-157
ENDUSER (Mark End of Job Stream)	2-158
ES (End Screen Display)	2-160
ET (End TRAK)	2-161
EXECUTE (Execute Library Program)	2-162
FI (List Files in Use by a Program)	2-166
FILE (Equate File-Name with File-Identifier or Medium)	2-167
Automatic Printer Assignment	2-171
FM (Direct File to Device—Special Forms Response)	2-175
FN (List File Names)	2-177
FP (Flush Program)	2-178
FR (Designate Final Reel)	2-179
GENERATE (Schedule DMPALL for Execution)	2-180
GO (Restart a Stopped Program)	2-182
GT (Initiate Trace)	2-183
HL (Request a Halt/Load)	2-187
HN (Identify or Change Hostname)	2-189
ID (Invoke DEBUG Session)	2-191
IL (Assign Labeled File)	2-192
IN (Insert Data into Program)	2-195
INSERT (Insert Data into Program)	2-197
IR (Initiate DMSII Recovery)	2-199
KA (Analyze Disk Directory)	2-201
KP (Print Disk Segments)	2-205
KS (Analyze Disk Space)	2-207
KX (Obtain User Disk Space)	2-210
Responses to KX	2-210
AUTO KX	2-211
LABEL1 (Change Tape Label Format)	2-212
LC (Insert Log Comment)	2-215
LD, LDP (Create Pseudo Card Deck)	2-217
LH (Load Firmware to Channel Controller/DCP)	2-218
LI (Log In)	2-221
LN (Transfer and Print Log)	2-223
LO (Log Out)	2-225

Contents

LOAD (Move Files from Library Tape to Disk or Diskpack) ..	2-226
LOCK (Lock Program)	2-227
LP (Lock Program)	2-228
MEMDUMP (Abnormal EOJ Memory Dump)	2-229
MEMORY (Allocate Memory for Program Execution)	2-230
MERGE (Transfer Files from Multiple Sources, Resolving Duplicates)	2-232
MOVE (Move Files to New Media)	2-233
MR (Remove Duplicate File)	2-234
MX (Display Jobs in Mix)	2-236
NET (Network Control Command)	2-239
NL (Resume at New Location)	2-242
NT (Terminate Trace)	2-243
NW (Networking Command Prefix)	2-244
OF (Optional File Indication)	2-245
OK (Continue Processing Suspended Program)	2-246
OL (Display Peripheral Status)	2-248
Channels	2-248
Peripheral Units	2-249
Magnetic Tape	2-250
Disk and Diskpack	2-251
Network Communication Ports	2-253
Pseudo Card Readers	2-254
OT (Display Program Data)	2-256
OU (Direct File to Device)	2-257
PA (Analyze Pack Directory)	2-259
PACK (Add a Pack)	2-262
PATCH (Patch File)	2-264
Patching a Running MCP Code File	2-264
Patching a Program or a Non-Running Code File ..	2-265
Patching Disk or Diskpack Files	2-266
Patching Disk Sectors	2-268
Verifying Patch Records	2-269
Executing the PATCHK Program	2-270
PATCHK Program Error Messages	2-271
PB (Print Backup File)	2-274
PC, PCP (Create Punch Backup File)	2-275
PD (Print Directory)	2-278
PERFORM (Schedule DMPALL)	2-280
PG (Purge Magnetic Tape)	2-282
PM (Print Memory Dump)	2-284
PO (Power Off Diskpack)	2-286
PP (Print Pack Sectors)	2-287
PR (Change Priority)	2-289
PRINT (Print Files)	2-291
Using AX to Enter Print Specifications	2-295
Executing PBDPRN	2-299
Auto Printing	2-299
PRIORITY (Set Priority)	2-301
PS (Analyze Pack Space)	2-303
QD (Query DEBUG Sessions)	2-307
QT (Quit Program Operation)	2-308

QWIKPOOL (Set Program Overlay Buffer Pool)	2-310
QWKMEM (Set MCP Overlay Buffer Pool)	2-311
RA (Remove After-Linkage)	2-312
RD, RDP (Remove Pseudo Decks)	2-313
REMOVE (Delete Files)	2-315
RERUN (Log a Rerun Operation)	2-319
RF, RFP (Remove Backup File from Disk or Pack)	2-320
RM (Remove Duplicate File)	2-322
RN, RNP (Activate Pseudo Reader)	2-324
RO (Reset System Options)	2-327
RP (Ready and Purge Magnetic Tape)	2-329
RQ (Remove Storage Queue Entries)	2-331
RS (Remove Job from Schedule)	2-333
RUN (Initiate Execution of Program)	2-334
RW (Rewind and Unload Tape)	2-335
RX (Return Deleted Space)	2-336
RXM (Return Memory)	2-336
RXD (Return Disk Space)	2-336
RXP (Return Diskpack Space)	2-337
RY (Ready Peripheral Device)	2-338
SB (Interrogate Database Activity)	2-340
SD (Set Deck Limit)	2-342
SHOW (Display System Information)	2-343
SI (Set Remote Device Identification)	2-347
SK (Skip Backup Records)	2-348
SM (Send System Message)	2-349
SN (Purge and Place Tape Number in Label)	2-350
SO (Set System Options)	2-352
SP (Display/Set/Change Database Parameters)	2-356
SPO (Simulate SPO through Card Reader)	2-357
SQ (Squash Disk)	2-359
SQP (Squash Pack)	2-360
SS (Send Message)	2-363
ST (Suspend Program Processing)	2-364
START (Restart a Stopped Program)	2-365
STOP (Suspend Program Processing)	2-367
SV (Save a Peripheral Unit)	2-368
SW (Set Programmatic Switches)	2-369
TEST (Mark a Test Operation)	2-371
TI (Display Time Statistics)	2-372
TIME (Discontinue Program After a Specified Amount of Time)	2-374
TL (Transfer Log)	2-375
TM (Write Tape Mark)	2-377
TO (Display System Options Status)	2-379
TR (Time Reset)	2-380
TRACE (Initiate Trace)	2-381
UL (Assign Unlabeled File)	2-383
UNIT (Add Peripheral Devices)	2-385
Adding an 80-Column Card Reader or Card Punch	2-385
Adding a Data Communications Processor	2-386

Contents

Adding an Image Page Printer (IPP)	2-387
Adding an Inter-System Connect (ISC)	2-388
Adding MPE, GCR, MTC and MT9 Tape Drives	2-389
Adding a Network Communications Port	2-391
Adding a Non-Status Device (NST)	2-392
Adding an ODT or OCS	2-392
Adding a Buffered Printer or Train Printer	2-393
Adding Data Communications Devices (RJE, TC5, VDD, TWX)	2-394
Adding Check Reader Sorters	2-395
UNLOAD (Move Library Files between Media, Remove Old Files)	2-397
UP (Unlock Program)	2-398
UR (Inhibit or Uninhibit Unit)	2-399
USER (Assign Access Code)	2-401
VALUE (Insert Value into a Program)	2-403
WB (Display Bound Programs)	2-405
WC (Display Available Memory)	2-406
WD (Display Date)	2-408
WJ (Display Waiting Jobs)	2-411
WM (Display MCP Version)	2-416
WO (Display Active User QWIK Operation Counts)	2-418
WQ (Display STOQUE Count)	2-420
WRU (Display ODT Access Code)	2-422
WS (Display Jobs in Schedule)	2-423
WT (Display Time)	2-424
WX (Display Removed Space)	2-425
WY (Display Job Status)	2-427
XA (Add/Remove a Channel from an Exchange)	2-432
XC (Inhibit/Uninhibit or Display Status of a Channel)	2-434
XD (Remove Disk Segments)	2-436
XM (Remove Memory from System)	2-437
XP (Remove Pack Sectors)	2-438

Appendix A. Understanding Railroad Diagrams

Introduction	A-1
Constants and Variables	A-2
Constraints	A-2
Vertical Bar	A-3
Right Arrow	A-3
Required Items	A-3
User-Selected Items	A-3
Loop	A-4
Bridge	A-4
Following the Paths of a Railroad Diagram	A-5
Railroad Diagram Examples	A-6

Appendix B. Commands for User-Coded Utilities

ADD (Copy Files)	B-2
Copy Tape to Disk	B-2
Copy Diskpack to Disk	B-2
Copy Diskpack Library Tape to Diskpack	B-2
Copy Disk to Diskpack	B-2
Copy Diskpack to Diskpack	B-3
Copy Disk Library Tape to Diskpack	B-3
Copy Diskpack Library Tape to Disk	B-3
CHECK (Verify)	B-4
DUMP (Copy to Library Tape)	B-5
Dump Disk to Library Tape	B-5
Dump Disk to Diskpack	B-5
Dump Diskpack to Library Tape	B-5
Dump Diskpack to Disk	B-5
Dump Diskpack to Diskpack	B-6
LOAD	B-7
Copy Library Tape to Disk	B-7
Copy Diskpack Library Tape to Diskpack	B-7
Copy Diskpack to Disk	B-7
Copy Disk to Diskpack	B-7
Copy Diskpack to Diskpack	B-8
Copy Disk Library Tape to Diskpack	B-8
Copy Diskpack Library Tape to Disk	B-8
UNLOAD (Move Files)	B-9
UNLOAD Disk or Diskpack to Tape	B-9
UNLOAD Diskpack to Tape	B-9
UNLOAD Disk to Diskpack	B-9
UNLOAD Diskpack to Disk	B-9
UNLOAD Diskpack to Diskpack	B-10
PBDOUT and PBTOUT (Print Backup Files)	B-11

Glossary

Index

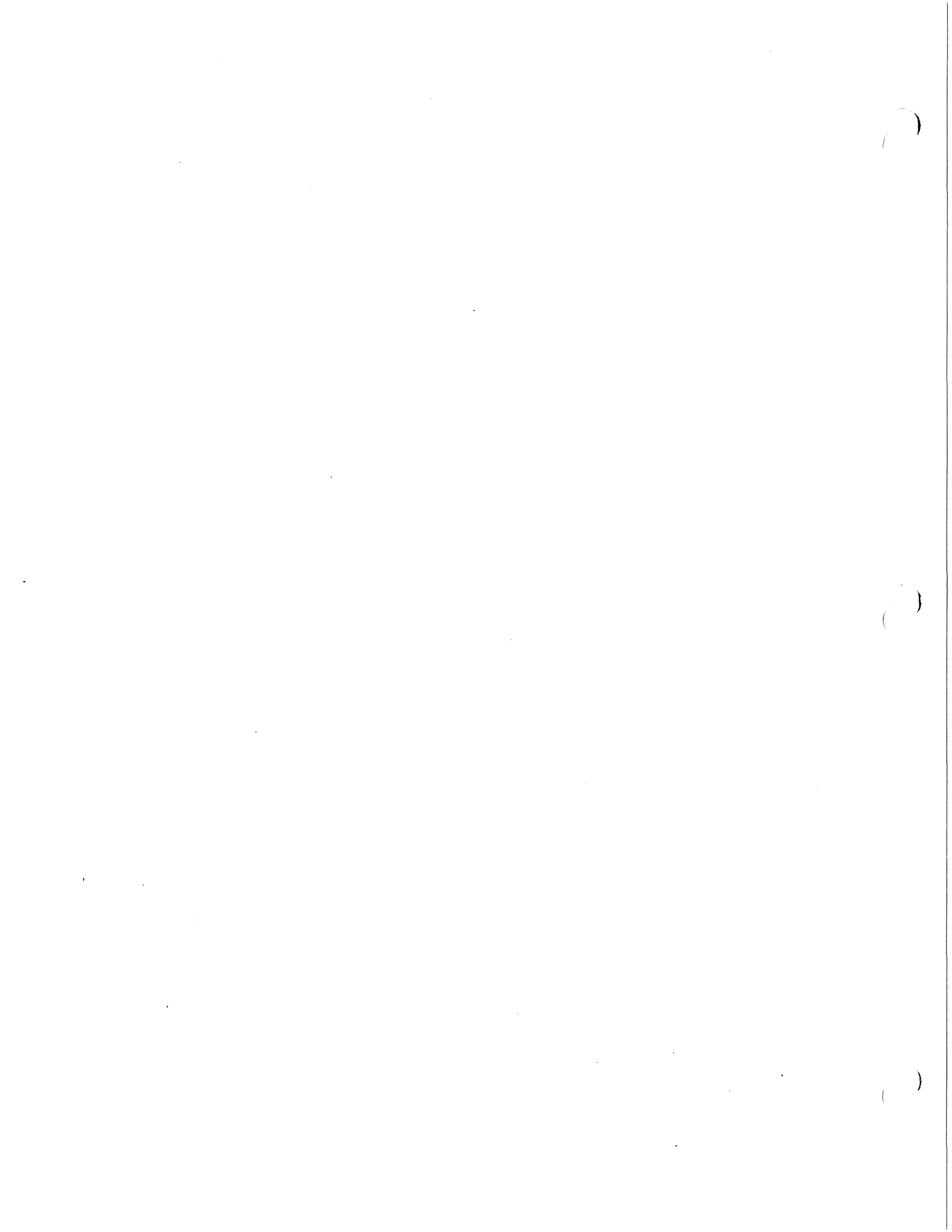
()

()

()

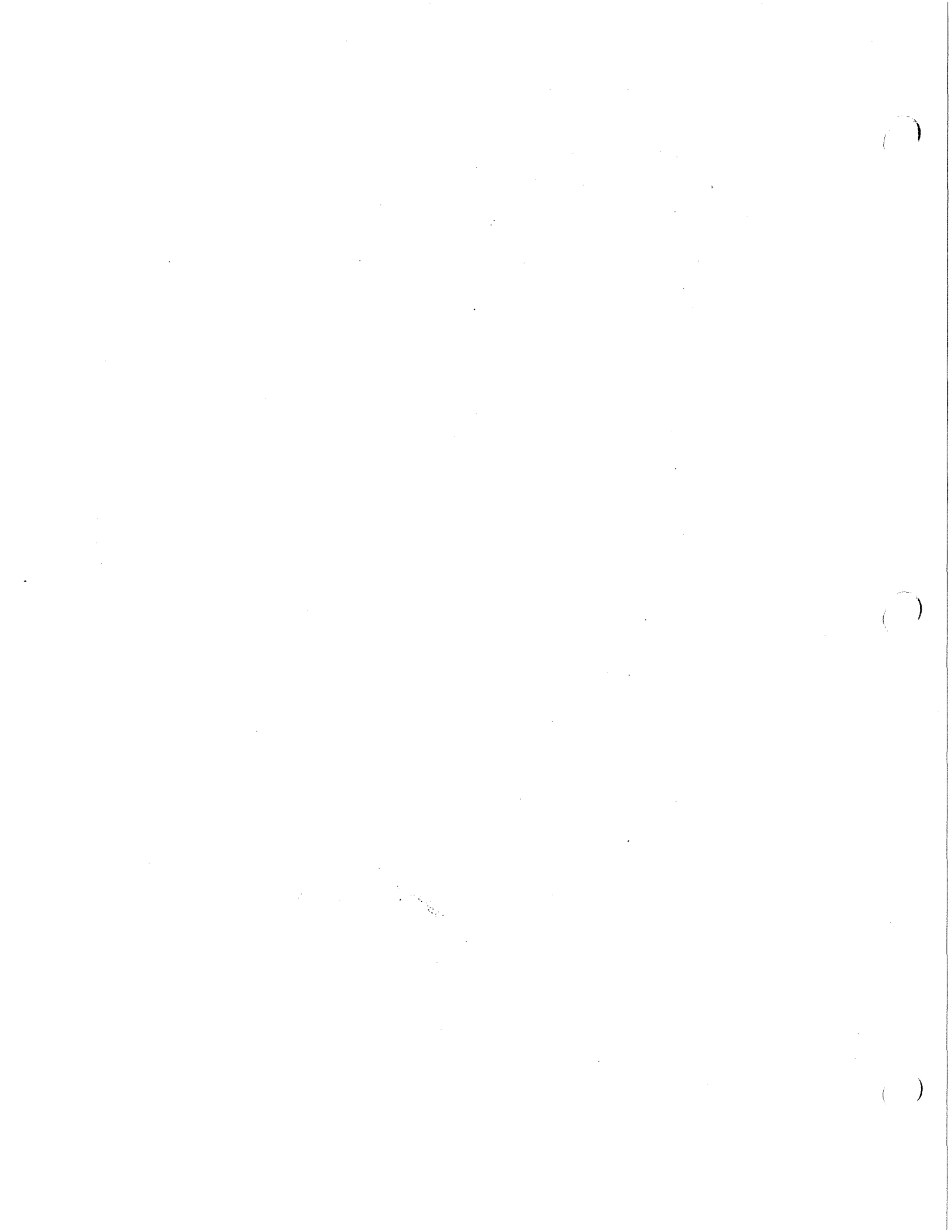
Figures

2-1.	System Response to the OL NCP Command	2-253
2-2.	System Response to a TI Command	2-372
2-3.	System Response to the WD Command	2-409
2-4.	Sample Response to the WJ Command	2-414
2-5.	Sample Response to the WM Command	2-416



Tables

1-1.	Examples of File Identifiers	1-9
1-2.	Reserved File Identifiers	1-9
1-3.	Invalid File Identifier Characters	1-10
1-4.	Commands and Their Access Level Classifications	1-13
2-1.	AD Command Definition	2-5
2-2.	Available Table Displays	2-7
2-3.	Valid Attributes for Source and Destination Volume Specifications	2-93
2-4.	DISK/LAK Diskpack Maximum Address Ranges	2-129
2-5.	Allowable File Equations Across Hardware Types	2-169
2-6.	Tape Label Record Format	2-212
2-7.	Host Load Files and Their Usages	2-219
2-8.	System Options Available For Setting with SO	2-352
2-9.	Memory Locations of Programmatic Switches	2-370



Section 1

Introduction to the System Commands

System Command Overview

Systems operators and programmers can direct the MCP to perform particular actions through the use of Master Control Program (MCP) system commands. These commands have many functions, including display or change of system, program, or device status and initiation and control of various system utilities. This guide lists the system commands in alphabetical order and discusses the use and syntax of each command.

System commands can be entered in the following ways:

- At an Operator Display Terminal (ODT) or Operator Control Station (OCS).
- Through the programmatic ZIP and ZIPSPO facilities in the syntax of the different programming languages. Most programs can contain ZIP statements that pass system commands to the MCP.
- Through programmatic SPOMESSAGE Branch Communicates (SPOM BCTs) that request information from the MCP.
- Through a card reader or pseudo card reader.
- Through the command capabilities provided with some programming tool packages, such as the Command and Edit (CANDE) system and the Programmer Productivity System (PROPS).
- Through the Work Flow Language (WFL).
- Through the Remote Job Entry (RJE) system, which enables communication between a remote site and a central system.
- Through the BNA network architecture system, which enables communications between remote systems.

Keyboard Input Messages Versus MCP Control Instructions

There are two types of MCP system commands: keyboard input messages and MCP control instructions. Each of the commands described in this guide specifies whether it is a keyboard input message or an MCP control instruction.

Keyboard input messages have a wide range of functions, such as inquiry and system maintenance. For the most part, keyboard input messages are 2- or 4-character abbreviations, such as DD (Delete DLP) or TR (Time Reset).

The MCP control instructions generally affect a program in the mix by either initiating a program or providing special parameters for the execution of a program. These control instructions are mostly full-word commands, such as EXECUTE (Execute Library Program) or PERFORM (Schedule DMPALL).

The precise differences in how the two are entered are as follows:

- The SPOMESSAGE BCT passes only keyboard input messages to the MCP, not MCP control instructions.
- A keyboard input message entered from a card reader or in a programmatic ZIP must be prefaced by SPO.
- When you enter commands through a ZIP statement within a program, the syntax among the different programming languages does vary, but MCP control instructions are usually preceded by ZIP, and keyboard input messages by ZIP SPO.
- A keyboard input message entered through CANDE or a programmatic SPOMESSAGE BCT is displayed on the ODT as "==> SPO" followed by the text that was sent. An MCP control instruction entered through CANDE or RJE is displayed as "==>" followed by the text you sent.
- All MCP control instructions supplied through the keyboard can optionally be prefaced with the CC (Enter Control Instructions) command (for example, CC EXECUTE MYPROG).
- If a system running MCP/VS release 2.0 or later is communicating with a system running MCP/VS 1.0 through BNA, the CC is mandatory when using MCP control instructions after the AT (Initiate BNA Job Transfer) command, which initiates a BNA job transfer between a local host and a remote host.
- The keyboard input messages that start a job must be prefaced by the word SPO after the AT command. This includes the CV (Convert Punch Backup File to Pseudo Card Deck), KA (Analyze Disk Directory), KP (Print Disk Segments), KS (Analyze Disk Space), PA (Analyze Pack Directory), PB (Print Backup File), PC (Create Punch Backup File), PM (Print Memory Dump), PP (Print Pack Sectors), and PS (Analyze Pack Space) commands.

Entering Commands through the ODT

System commands are entered at the top line of the ODT, and they can continue to subsequent lines.

To enter commands through the ODT, do the following:

1. Position the ODT cursor at the top of the screen by pressing the CLEAR/HOME key, if necessary.
2. Type the desired command.
3. If the ODT is a B 20, ET 1100, TD 830, or MT 983 device, you can transmit the command by pressing the transmit key. All data preceding the cursor will be transmitted to the MCP. For other devices, press the End-Of-Text

(ETX) key, which places an ETX character on the screen and returns the cursor to the Home position. Then press the transmit key.

Optionally, the command can be terminated with a period (.) character prior to the ETX, in which case all text following the period is ignored. When the MCP acknowledges the command, the ODT is set in receive mode.

If you have multi-line input, put an ETX character at the end of the message, put the cursor at the beginning of the message, and then press the transmit key.

Note: Pressing the transmit key causes everything from the ODT cursor to the ETX character to be sent to the MCP. It is important, therefore, that the cursor be positioned at the beginning of the message before the transmit key is pressed and that no spurious ETX characters be on the screen.

If you make an error while typing in a command, press the CLEAR/HOME key and then retype the command.

If a command is entered incorrectly, the MCP returns the following message:

```
**KBD IGNORED: INV SYNTAX
```

MCP Response to ODT Input

The MCP responds to ODT keyboard input in the following way:

A single-line response to an input request will be displayed on the second line of the screen (the remainder of the screen will not be cleared). For a multiple-line (or no-line) response, the second line of the screen will be cleared. A multiple-line response will be displayed following the header and any skip-lines.

System commands are indented one additional space when they are displayed on the ODT "messages" screen and in the ODT log listing to make it easier to identify input requests. However, this does not affect keyboard output messages returned to a SPOMESSAGE BCT.

The system message display is indented four spaces. The lines that have not previously appeared in a display will be highlighted, independently of system message displays on any other ODT terminal. It will therefore be possible to see at a glance which messages are new.

When input messages are entered from an ODT, they appear in a DQ (Display System Messages) command response and/or from the AD MSG display. All input messages are flagged with "=>" to the left of the display line.

If the command requires the MCP to display multiple lines, all available lines except the communication interface area and lines reserved by the ODT SK option are used, and lines currently being displayed are overwritten or erased. See UNIT Record in the *V Series Systems Operations Guide Volume 1: Installation*, and the

AD (Specify ODT Screen Parameters) command for a description of the ODT SK, HDR, and MSG options.

Replies that exceed one line remain on the screen for 30 seconds or until the system operator enters an ETX (or transmits blanks) for a new screen display, an ES (End Screen - ODT Only) command to end screen display, or another input request that would cause the current display to be overwritten or erased.

If the reply to the command requires more lines than are available on the ODT screen, the MCP uses all available lines except the communication interface area lines and the lines reserved by the ODT SK option. The MCP then displays "MORE DATA" in the communication interface area. No more data is written on the screen for 30 seconds or until the system operator enters an ETX, one or more blanks, ES, or another keyboard input request. If, after 30 seconds, the system operator has not performed any of these operations, the MCP displays a new screen containing the overflow data.

Entering Commands through a Programmatic ZIP or ZIPSPO

The commands sent with the ZIP or ZIPSPO facilities are acted on exactly as if they had been entered at the ODT. The ZIP and ZIPSPO facilities are invoked by constructs in the various programming languages, which are discussed in the various language manuals.

If a program under execution contains a ZIP or ZIPSPO statement or equivalent syntax in a programming language that performs communication with the MCP, the statement must refer to a data area within the program. The information in that data area must conform to the following rules:

- Unless the first character of the command is a period, the data area is assumed to be 73 characters long. The text can be up to 72 characters long, followed by a period. If a period is not present, the MCP will insert one in the 73rd character position.
- If the first character of the command is a period, the data area can be of any length, and the MCP will not insert a period. The end period must therefore be present in the program.
- The hyphen (not enclosed in parentheses) cannot be used as a continuation character in a ZIP statement.
- The program must be able to handle 4-digit mix numbers.

Note that MCP/VS commands not supported by MCP/VS cannot be used in a ZIP statement.

Entering Commands through a ZIPSPO

The ZIPSPO function gives a user program the ability to pass control information to the MCP and to determine if any errors occurred during the processing. If there is an error, a response will be returned to the program.

The ZIPSPO response area is 82 characters long. The first two characters contain a numeric error code; the remainder contains the text.

If no errors are detected, the error code will contain "00" followed by a 4-character value containing the Run log number of the last job initiated by the ZIPSPO or "0000" if no jobs were initiated. The remaining 76 characters are undefined. If an error is detected, the first two characters are non-zero and the following 80 characters will be a "***ZIP IGNORED: ..." keyboard response terminated by ETX.

Entering Commands through a SPOMESSAGE BCT

The SPOMESSAGE BCT passes keyboard input messages to the MCP and requests that the response be returned to the requestor. The input message must be terminated by either a period (.) or an ETX character.

Note that you cannot enter MCP control instructions through a SPOMESSAGE BCT.

The length of the response buffer must be at least 80 characters and must not overlap the input area. The output buffer should be cleared before calling this BCT.

If you enter a command not valid through a SPOMESSAGE BCT, a message stating that the request is not allowed appears on the ODT.

See the *V Series Program Interfaces Programming Reference Manual* for more information.

Entering Commands through Punched Cards

If punched cards are used to communicate a command, the following rules apply:

- The first column must contain an invalid character, known as the Punched Cards Control Character. This control character is a punch combination that does not represent any valid character. For example, multi-punched 1-2-3 is most commonly used. Invalid characters are not allowed in any other column. The first word following the invalid character must be a valid command. When a command is contained in a pseudo-card file, the first column must contain a question mark character ("?", EBCDIC 6F).

Only columns 2 through 72 can contain commands; the MCP ignores information after column 72. Commands can start in any position between columns 2 through 72.

Introduction to the System Commands

- If the special character period (.) appears in a control record, other than as part of a file name in parentheses, such as (B974.1), all information following the period is ignored for control purposes. This enables comments to be present in control records.
- The special character semicolon (;) must be used to designate the end of a series of commands if a new command is to start on the same punched card or if the MCP would not otherwise recognize the new control data. This is used primarily to enter control data such as CHARGE, PRIORITY, MEMORY, and so forth, to a system intrinsic such as DMPALL. The semicolon can be used on a ?SPO record to designate multiple commands (for example, ?SPO KA =; MEM + 20 PR 5).
- Commands can be contained on more than one record, except for the ?SPO commands, which must always be contained on a single record. The continuation character hyphen (-) can be used to continue information from one record to another. The hyphen must not be used to divide any word. Any information following the hyphen (on the same record) is ignored. The record on which the information is continued must not contain an invalid character in column 1.
- All commands are described in Section 2 under headings that appear to indicate that each of them must consist of a separate record (a COMPILE record, a PRIORITY record, and so forth). However, if the text of a command is delimited by a semicolon, indicating the logical end of that command, it can then be followed by another command on the same record.
- Any program name or file name that begins with a number or contains special characters must be enclosed in parentheses.
- The MCP treats every program name and file identifier as if it had six characters, and shorter identifiers are assumed to have an appropriate number of trailing blanks. Consequently, any program name or file identifier that contains special characters or embedded blanks must be enclosed in parentheses. It is syntactically correct to enclose only the actual identifier within parentheses. For example, (X-X) and (X-X) are both correct.

Entering Commands through CANDE, PROPS, or WFL

The Command And Edit (CANDE) system is a multi-user timesharing facility that provides several levels of system and file access control. If you have CANDE installed at your site and want to enter commands through CANDE, see the *V Series CANDE Installation and Operations Guide* for detailed information.

The Programmer's Productivity System (PROPS) is an interactive program development environment. If you have PROPS installed at your site and want to enter commands through PROPS, see the *B 2000/B 3000/B 4000/V Series PROPS Operations Services Operations Guide* for detailed information.

The Work Flow Language (WFL) is a Unisys language used for constructing jobs that compile and run programs on Unisys computer systems. WFL includes

variables, expressions, and flow-of-control statements that offer a range of task control capabilities. See the *B 2000/B 3000/B 4000/V Series WFL Compiler Programming Reference Manual* for detailed information about WFL.

Entering Commands through RJE

The Remote Job Entry (RJE) system is a software package that permits commands to be sent from a remote site to a central system, and that enables output of data from the central system to be sent to remote peripherals.

See the discussion of RJE in the *V Series Systems Operations Guide Volume 3: System Utilities*.

For more information on the operation of RJE, see the *B 2000/B 3000/B 4000/V Series Remote Job Entry (RJE) Operations Guide*.

Entering Commands through BNA

BNA network architecture is a software system used to link Unisys computer systems together in a data communications network. The BNA network supports full data communications, distributed processing, and virtual station transfer, as well as remote command entry.

For more information on the operation of BNA, see the *V Series BNA Installation and Operations Guide*.

Command Details

The following comments apply generally to all the system commands:

- If the security option is set, any input device (ODT, card, reader, CANDE station) must be logged in with a valid access code (which can include a user code, password, and charge number) before commands can be entered through that device. Pseudo-card files must contain LI, USER, or BEGINUSER commands. See the *V Series System Software Security Installation and Operations Guide* for more information.
- Leading zeros are generally optional unless otherwise specified.
- The words and numbers in a system command are separated by delimiting characters, which are either blank spaces, commas, or semicolons. A space is sufficient in most cases.
- Any number of spaces may be placed between words without affecting the meaning. For example, the following are equivalent:

```
COPY A,B TO TAPE
COPY A , B TO TAPE
KS =; MEM +10
KS = ; MEM + 10
```

Introduction to the System Commands

- Parentheses must be used to enclose identifiers containing non-alphanumeric characters.
- Up to 400 bytes of input text can be entered from an ODT. It is not possible for a word to wrap around from one line to the next because the terminal automatically inserts a CR/LF at the end of each line transmitted.
- If a system running MCP/VS release 2.0 or later is communicating with a system running MCP/VS 1.0, system commands must be prefaced with the characters CC if they are entered through BNA at an ODT and if they start a job (that is, ADD, CHECK, COMPILE, COPY, CV, DIR, DUMP, EXECUTE, GENERATE, KA, KP, KS, LOAD, MERGE, MOVE, PB, PC, PERFORM, PM, PP, PRINT, PS, and UNLOAD).
- You cannot use a mix number of 0 or 1 in a system command, except for the DM (Dump and Continue) command, which allows 0 DM. Mix numbers 0 and 1 are reserved for the use of MCP/VS. If you do enter a mix number of 0 or 1, an error message will be displayed.

File Identifiers

A file identifier is the name contained in the directory for the referenced device, such as a tape, disk, or pack. A file identifier can be up to six characters long, but cannot be made up of all nulls, all blanks, or all zeros.

All library maintenance commands use an expanded method of identifying a file identifier. A file identifier can contain any combination of EBCDIC characters with the exception of left and right parentheses, period, comma, semicolon, hyphen, slash, and space; however, these characters (except the right parenthesis) can be used if they are enclosed in parentheses.

An equal sign not enclosed in parentheses, used in any position of a file identifier, indicates that any character is acceptable in that character position. An equal sign used in this fashion is also called a masking character.

A file identifier that contains masking characters is referred to as a file-group identifier. Trailing equal signs are assumed if the file identifier ends with an equal sign.

The following examples, as shown in Table 1-1, are specific file identifiers that would be syntactically equal to the file-group identifier.

Table 1-1. Examples of File Identifiers

File-Group Identifier	Matching File Identifiers	Remarks
A=	A ABC ASMBLR	All file identifiers starting with the letter A.
A	A	Only the file identifier A.
=A=	AA BA ZAYZ	All file identifiers with an A in the second character position.
A====F	ABCDEF AWXYZF	All file identifiers with an A in the first character position and an F in the sixth.

An equal sign must not be separated by a blank space from the letters of a file-group identifier. A single equal sign normally causes all file identifiers to be specified. Exceptions are detailed under each applicable system command.

The file identifiers listed in Table 1-2 are reserved for use by the MCP and cannot be used for user-created disk files.

Table 1-2. Reserved File Identifiers

Reserved File Identifiers	Reason
ADFL*W	AD rule overflow file (* is a system number and can be 0 through 3)
CONFIG*	System Configuration File (* is a system number and can be 0 through 3)
CTLDCK	LDCNTL card deck
DSKAVL	Disk available table
DSKDIR	Disk directory
hlparm	Halt/Load Parameters
MCPMCP	Operating MCP area
mstavl	Master Available Tables
nLOG*w	MCP Logs (system and option dependent) (n denotes the name of the log and can be M, R, or S) (* is a system number and can be 0 through 3) (w denotes a rotating log file number and can be 0 through 9 or X. 0 is the current log file and X is a reserved log file.)
PCRXRF	Pseudo reader dictionary

Table 1-2. Reserved File Identifiers (cont.)

Reserved File Identifiers	Reason
pntp*t	Program Name Table (* is a system number and can be 0 through 3)
SQARCY	SQUASH Recovery file
USERFL	System Access Control file
USRCOM	System Access Control file
\$*0001	MCP memory dump area (* is a system number and can be 0 through 3)

Special Characters

Table 1-3 lists the special characters that cannot be used in a file identifier except when enclosed in parentheses.

Table 1-3. Invalid File Identifier Characters

Reserved Character	Reason
Period (.)	A period delimits an ODT message line or a system command on one record.
Semicolon (;)	A semicolon denotes another system command to immediately follow.
Comma (,)	A comma delimits a file identifier.
Hyphen (-)	A hyphen indicates command continuation.
Equal (=)	An equal sign indicates a masking character position.

Be careful when using file identifiers that begin with special characters that are also used by the MCP, such as the following:

- @ which identifies a printer backup file
- # which identifies a pseudo card deck
- * which identifies a punch backup file
- \$ which identifies a dump file

Note: Files named ":" followed by five numbers (for example, ":03F01") will be removed from disk after every halt/load.

Command Restrictions

The following restrictions apply to the various system commands:

- The following commands are invalid from a programmatic ZIP or ZIPSPO:

ALLOCATE	DL	LO
ALTER	DLP	PACK
BCL	DQ	PATCH
BINARY	EBCDIC	QD
BO	ED	QWIKPOOL
DATA/DATAB	ES	QWKMEM
DD	HL	SHOW
DEBUG	ID	UNIT
DISK	KX	XA
DISPLAY/WHATS	LABEL1	

The REMOVE === (remove all files) command cannot be issued using a programmatic ZIP since it requires a null mix.

- The following commands are invalid from program for a SPOMESSAGE BCT:

AC	KX	RN
AD	LD	RP
AT	LH	RQ
BO	LN	RS
BT	LO	RW
CV	MR	RX
DD	NET	RY
DL	NW	SD
DM	OF	SI
DP	OK	SM
DR	OU	SN
DS	PA	SP
ED	PB	ST
ES	PC	SV
ET	PG	TL
FM	PM	TM
FP	PO	TR
FR	PP	UL
HL	PR	XA
ID	PS	XD
IL	QD	XM
KA	QT	XP
KP	RA	
KS	RM	

- The following commands are invalid through a Card Reader (SPO commands):

BO	ES	LO
BT	ET	QD
DEBUG	ID	XA
DQ	KX	
ED	LI	

Introduction to the System Commands

- The following commands are invalid from a WFL handler:

ADD	PRIORITY	PRS
CHECK	PR	UNLOAD
DUMP	PRM	
LOAD	PRP	

- The following commands are invalid through Remote Job Entry (RJE):

AC	KX	RX
AD	LABEL1	RY
ADD	LD	SD
ALLOCATE	LH	SHOW
ALTER	LN	SI
AT	LOAD	SN
BT	NET	SO
CHECK	NW	SP
CK	PACK	SQ
DB	PATCH	SV
DD	PG	TL
DISK	PO	TM
DISPLAY/WHATS	PR	TR
DL	PRIORITY	UNIT
DLP	PRM	UNLOAD
DR	PRP	UR
DUMP	PRS	XA
DX	QD	XC
ED	QWIKPOOL	XD
ES	QWKMEM	XM
ET	RO	XP
HL	RP	
ID	RQ	
IR	RW	

- The DM command is valid for MCP mix number 0.

Access Level

A terminal can be restricted to permit input of only certain types of system commands. These restrictions can be set in the LEVEL parameter of the UNIT ODT record of the system configuration file, which declares the terminals that communicate with the MCP. See the *V Series Systems Operations Guide Volume 1: Installation* for more information.

Table 1-4 shows the type and access level of each command.

Note that all commands permitted to lower-level ODTs are also permitted to ODTs with higher classifications. Except for level 8 or 9 ODTs, commands requiring a <mix no> can be used only if the <mix no> refers to a program originating from the station that also originated the command.

Table 1-4. Commands and Their Access Level Classifications

Access Level	0	1	3	5	7	8
General Classification	Inquiry	Job Initiation and Termination	Job Maintenance	Disk Maintenance	Peripheral Maintenance	System Maintenance
Commands Allowed	AJ	AFTER	FILE	ADD	AC	ALLOCATE
	AT	AX	FN	CHANGE	AD	ALTER
	AX	CD,CDP	GO	CHECK	CK	BT
	BCL	CHARGE	GT	COMPARE	CN	DBCA
	BEGINUSER	COMPILE	IN	COPY	FM	DBCL
	BF	DC	MEMORY	COPYADD	LD	DBDR
	BFRF	DEBUG	NL	CV	PG	DBIC
	BINARY	DM	NT	DA	PO	DBMM
	BO	DP	OT	DIR	RP	DBSR
	BUFFER	DS	PR	DISPLAY/	RW	DBUA
	CA	EXECUTE	PRIORITY	WHATS	RY	DBUL
	CC	FP	PRM	DUMP	SN	DD
	DATA/B	FR	PRP	IL	SV	DISK
	DQ	GENERATE	PRS	KA	TM	DL
	EBCDIC	INSERT	SHOW	KP	UL	DLP
	END/	LC	SK	KS		DR
	ENDCTL	LOCK	ST	LOAD		DX
	ENDAT	LP	START	MERGE		ED
	ENDUSER	MEMDUMP	STOP	MOVE		ET
	ES	OF	TRACE	MR		HL
	FI	OK		OU		ID
	HN	PB		PA		IR
	KX	PC		PATCH		LH
	LABEL1	PERFORM		PP		LN
	LI	PM		PS		NET
	LO	PRINT		RD		NW
OL	RA		RF		QD	

Introduction to the System Commands

Table 1-4. Commands and Their Access Level Classifications (cont.)

Access Level	0	1	3	5	7	8
General Classification	Inquiry	Job Initiation and Termination	Job Maintenance	Disk Maintenance	Peripheral Maintenance	System Maintenance
	PD	RERUN		RM		QWIKPOOL
	SM	RN		UNLOAD		QWKMEM
	SPO	RS				RO
	SS	RUN				RQ
	TI	SW				RX
	TO	TEST				SB
	USER	TIME				SD
	WB	UP				SI
	WC	VALUE				SO
	WD					SP
	WJ					SQ
	WM					SQP
	WO					TL
	WQ					TR
	WS					UNIT
	WT					UR
	WY					WX
						XA
						XC
						XD
						XM
						XP

Section 2

The System Commands

This section contains the system commands listed in alphabetical order. The description for the commands, in most cases, includes the following:

- Whether the command is a keyboard input message or an MCP control instruction
- The ODT access level of the command
- Introductory material stating the function of the command
- A railroad syntax diagram of the command. If you do not know how to interpret railroad syntax diagrams, see Appendix A.
- An explanation of each element in the diagram, followed by command details or restrictions
- An example of the command
- References to other relevant commands and documents

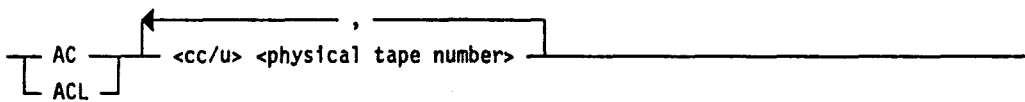
AC (Purge and Place Tape Number in Label)

Keyboard Input Message

Access Level: 7

The AC command purges a tape and puts a permanent identifying external physical tape number into a magnetic tape label. The AC command also has additional capabilities that allow you to assign the tape to a mix number when the tape is purged and/or to override a program's tape type request.

Purge a Tape



Explanation

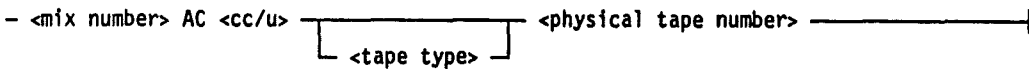
ACL This locks the tape after purging it.

<cc/u> This is the channel number and unit number of the appropriate device. The tape must be ready and contain a write ring.

<physical tape number> This is a number that is placed into the label of the designated tape reel. The number can be any integer up to five digits in length.

The **<physical tape number>** remains in the label throughout the life of the tape or until you perform another AC or SN (Purge and Place Tape Number in Label) operation on that tape.

Purge and Assign a Tape to a Mix Number



Explanation

<mix number> This is the mix number of a job. Inclusion of **<mix number>** causes the tape on the drive designated by **<cc/u>** to be purged and assigned to the job with this mix number.

If the tape has GCR capability and no tape type is specified, GCR mode will be used.

<tape type>

The <tape type> enables the operator to override the tape type request of a program (allowed only when the <mix number> is specified). The <tape type> can be MT9, MPE, GCR or MTC.

Details

If you do not specify the <mix number> and <tape type> *and* you are using a GCR-capable drive, the tape will be purged and made available in GCR mode.

Examples

ACL 4/3 4

This purges the magnetic tape on channel number 4, unit number 3, places the number 4 in its label, and then locks the tape.

AC 4/3 12345, 4/2 54321

This purges the magnetic tape on channel number 4, unit number 3 and places the number 12345 in its label. This also purges the magnetic tape on channel number 4, unit number 2, and places the number 54321 in its label.

14 AC 6/3 GCR 84127

This purges the GCR magnetic tape on channel number 6, unit number 3, assigns it to mix number 014, and places the number 84127 in its label.

See also

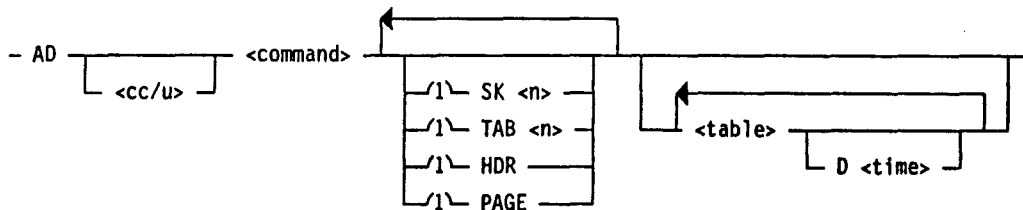
- The CN (Display Tape Number) command on Page 2-78.
- The LABEL1 (Change Tape Label Format) command on Page 2-212.
- The SN (Purge and Place Tape Number In Label) command on Page 2-350.
- The TM (Write Tape Mark) command on Page 2-377.

AD (Specify ODT/OCS Screen Parameters)

Keyboard Input Message

Access Level: 7

The AD command controls the automatic display of system status information on an ODT or OCS by adding or deleting parameters.



Explanation

<cc/u>

This specifies the channel number and unit number of the ODT or OCS device to which the designated parameters apply. If a <cc/u> is not specified, the AD command applies to the current device.

<command>

This specifies the action to be performed by this command. Table 2-1 lists the valid actions of the AD command.

Table 2-1. AD Command Definition

<command>	Meaning
ADD	Adds the table displays and display options immediately following this keyword to the current table display parameters. Any duplicate table entries are replaced with the new parameters.
DEF	<p>Applies default automatic display parameters to the device. The default display consists of three sets of tables.</p> <p>The first set of tables lists information about the jobs active in the mix, jobs waiting, and jobs in the schedule. These tables appear on the left side of the first screen of information.</p> <p>The second set of tables has four tables that list information regarding the status of the disk, diskpack, tape, and printer devices on the system. These tables appear on the right side of the first screen of information.</p> <p>The third set of tables lists the latest commands that have been input to the MCP and any system output messages. These tables are on the second screen of information.</p> <p>The displays are formatted so that the first and second lines are blank. The third line displays a heading with the system name, available memory and CPU, date, and time. The tables are displayed below that line for 10-second intervals. Also, the PAGE option is set, which allows any excess information that could not fit on the screen to be displayed when the user transmits a blank space.</p> <p>The following example represents the AD syntax that would be required to set the automatic display parameters represented by DEF (the default automatic display rules).</p> <pre>AD RPL SK 1 HDR PAGE (AM WM S) (DSK DPK MTP PRN) D 10 MSG D 10</pre>
DEL	<p>Deletes the table display or table displays immediately following this keyword from the current list of tables automatically display by the system.</p> <p>Multiple table names can be deleted. A group of table names enclosed in parentheses can be deleted provided the number of lines, n, and time, if specified, for the group are the same as those items to be deleted.</p>
DIS or blank	Displays the current automatic display parameters for the device.
GO	Resumes the automatic display.
RPL	Replaces the current display parameters with the table display options immediately following this keyword.
ST	Stops the automatic display.

Note: *If the display has been previously stopped by an AD ST message, subsequent displays of the rules are appended by "...STOPPED."*

The options following <command> specify table display parameters and are the same as the System Status Tables options described in the *V Series Systems Operations Guide Volume 1: Installation*.

- SK <n>** This sets the number of lines skipped before text is displayed. The lines are skipped starting from the header, or from the top of the screen if there is no header text. The value of <n> can be any number from 1 to 9. If you do not specify SK, the display of text starts on line 3 or on the line immediately following the header line.
- TAB <n>** This sets the left margin of the table display, where <n> represents a column number and must be a multiple of 8. The maximum value for <n> is 40. The default value is 24.
- HDR** This causes the following information to be displayed on line 3:
- Host name
 - Available memory
 - Available CPU
 - Date
 - Time
- PAGE** This enables a display to exceed one screen if there are more lines to display than will fit on one screen.
- If PAGE is not specified, all table displays (except responses to operator inquiries) will be truncated to one screen.
- If you specify PAGE, you can display the extra information by transmitting a blank space to the system (pressing the space bar and then the transmit or SPCFY key). The message "More Data" indicates that more information is available.
- When two tables are displayed simultaneously, the table on the left can overflow to the right. This can happen whether or not this option is set. Once all the extra information has been displayed, normal table display resumes automatically.
- D <time>** This specifies the time interval in seconds between updates to the display, where <time> can be any value from 1 through 99. The default for <time> is 10 seconds. Time intervals less than 3 to 5 seconds are not recommended due to excessive system overhead. If entries on a page have conflicting delay values, the value of the last entry is used.

<table>

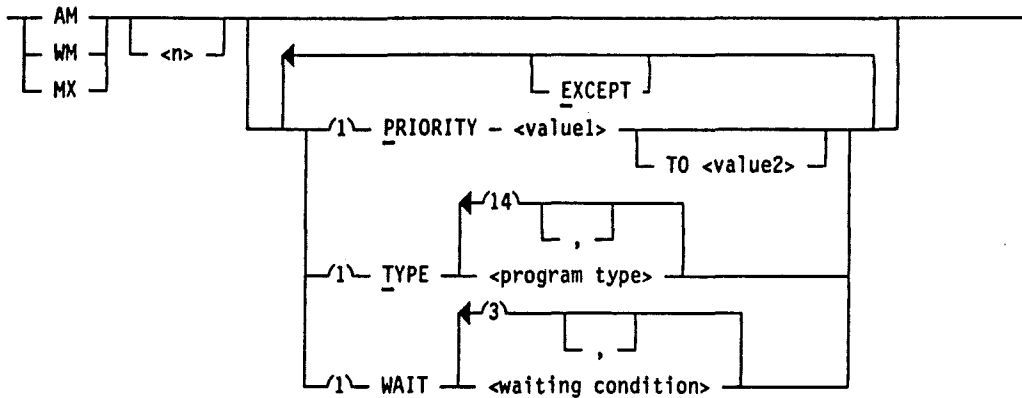
This determines which table or tables are to be displayed. If multiple tables are to be displayed together, they must be enclosed in parentheses. The available table displays are defined in Table 2-2. The optional parameter <n> indicates the number of lines of information to display.

Table 2-2. Available Table Displays

Option	Meaning
AM	This displays information on active jobs. The AM table selection is discussed further following this table.
DPK <n>	This displays the status for all or <n> diskpacks on the system.
DSK <n>	This displays the status for all or <n> disks on the system.
MSG <n>	<p>This displays of the most recent commands received by the MCP along with any message the MCP might have issued. No more than 22 commands or responses can be displayed at one time.</p> <p>If you want to display less than 22 lines of text, indicate the number of lines you want by entering the number following MSG.</p> <p>Because this table is 80 characters wide, it cannot be displayed simultaneously with any other table.</p>
MTP <n>	This displays the status for all or <n> magnetic tape drives on the system.
MX	This displays information on both active and waiting jobs. The MX table selection is discussed further following this table.
PRN <n>	This displays the status for all or <n> printers on the system:
S <n>	This displays information about scheduled jobs. This information includes the processor and memory priorities, the amount of memory in use, the length of time a job has been in the schedule queue, and the after-mix number for each job.
WM	This displays information on waiting jobs. The WM table selection is discussed further following this table.

AM, WM, MX Tables

Special syntax is used for the table selections AM, WM, and MX. These tables display information about jobs in the mix. The following syntax diagram shows the syntax for these table selections.



Explanation

AM
WM
MX

These three commands display information about jobs in the mix.

AM displays jobs in the active mix, including the task number, the processor priority, the memory priority, and the amount of memory in use for each job.

WM displays jobs in the mix that are waiting for a condition to be satisfied before processing resumes. The information displayed for each waiting job includes the processor and memory priorities and the amount of memory in use.

MX displays the status of the and at the same time.

<n>

Limits the lines of information, and consequently, the number of jobs displayed on the screen. The value of <n> can range from 1 to 22; the default is 19.

PRIORITY <value1>
TO <value2>

Selects jobs based on processor priority. You have the option of selecting a single priority or a range of priorities. The values of <value1> and the optional <value2> must be in the range 1 through 9. The value of <value1> must be less than or equal to <value2>.

TYPE <program type>

Selects jobs based on the type of program. You can enter a maximum of 14 type selections. The available program types are as follows:

<u>B</u> NAHNDL	Program is a BNA handler.
<u>C</u> OMPILER	Program generates object code.
<u>C</u> OMP <u>S</u> HAR	Program that generates object code, executed as a timesharing task.
<u>C</u> OPY	Program is SYSTEM/COPY intrinsic.
<u>D</u> MPALL	Program is DMPALL intrinsic.

<u>DMS</u> PROG	Program is a DMSII control program (for example, DBP).
<u>DSK</u> OUT	Program is DSKOUT intrinsic or diskpack SQUASH intrinsic.
<u>EXECUTE</u>	Program is executing.
<u>HANDLERS</u>	Program is any type of handler (BNA, WFL, time sharing).
<u>LOAD</u> MPP	Program is LOADMP intrinsic.
<u>MCS</u>	Program is Message Control System (MCS).
<u>PACK</u> UP	Program is PACKUP intrinsic.
<u>SHARE</u> HNDL	Program is a timesharing handler program (for example, CANDE).
<u>SORT</u>	Program is SORT: or SORT. intrinsic.
<u>TIMESHARE</u>	Program is executed as time sharing task.
<u>WFL</u> HNDL	Program is a Work Flow Language (WFL) handler.
<u>UTILITY</u>	Program is any type of intrinsic (DMPALL, LOADMP, PACKUP, DSKOUT, SYSTEM/COPY).

WAIT
<waiting condition>

Selects jobs that are waiting for a specific condition to be satisfied (for example, waiting hardware, waiting memory and so forth). You can enter as many as three waiting conditions and three waiting condition exceptions.

The available types of waiting conditions are as follows:

<u>DEVICE</u>	Waiting for hardware device to become available
<u>EVENT</u>	Waiting for a software event to be signaled (for example, STOQUE, CRCR, and so forth)
<u>OPERATOR</u>	Waiting for operator to perform an action
<u>SPACE</u>	Waiting for memory, disk space or diskpack space

EXCEPT

All text that follows the keyword EXCEPT is treated as exceptions to the selections entered previously. All of the exceptions must be grouped together after the keyword EXCEPT; you cannot enter a selection and an exception followed by another selection.

Details

All but one of the system status tables are 40 columns wide. Because of the table size, the screen can be split to display two or more of these tables simultaneously. One or more tables can appear on one side of the screen, while one or more tables appear on the other side of the screen. Since the tables displayed by the MSG option are 80 columns wide, they require the entire screen.

When setting up the table displays, it is important to consider the amount of information the system has to display. For an open, readable display, group together the tables that have a limited amount of information. You could also limit the amount of information in that table.

To display more than one table at a time on the same half of the screen, enclose the options for those tables in parentheses. If the options are not in parentheses, the status tables are displayed one at a time.

Example

```
AD 6/3 RPL HDR (AM WM) (DSK DPK MTP)
```

This replaces the current automatic display of the system status information for the OCS on channel number 6, unit number 3 with the following rules. There will be a header on line three. The active and waiting mix tables will be on one half of the screen, while the disk, diskpack, and tape status tables will appear on the other half.

See also

- The AJ (Display Active Jobs) command on Page 2-14.
- The ES (End Screen Display) command on Page 2-160.
- The MX (Display Jobs In Mix) command on Page 2-236.
- The OL (Display Peripheral Status) command on Page 2-248.
- The WJ (Display Waiting Jobs) command on Page 2-411.
- The WS (Display Jobs In Schedule) command on Page 2-423.
- The WY (Display Job Status) command on Page 2-427.
- The *V Series Systems Operations Guide Volume 1: Installation* for more information about system status tables.

ADD (Copy Files from Library Tape to Disk or Diskpack)

MCP Control Instruction

Access Level: 5

The ADD command copies files from a LOADMP or PACKUP library tape to disk or diskpack but does not replace an existing file.

It is recommended that you use the COPYADD option of the COPY command instead of the ADD command. The ADD command will become unavailable at a future release. See "COPY (Invoke File Transfer Utility)" for more information.

The ADD command starts a program called LOADMP, if the files are to be copied to disk, or a program called PACKUP, if the files are to be copied to diskpack. LOADMP and PACKUP must be written by the user; Unisys does not supply them. If the appropriate program is not present on disk, the ADD command is ignored.

See Appendix B for ADD syntax if you have user-coded LOADMP and/or PACKUP utilities.

See also

- The COPY (Invoke File Transfer Utility) command on Page 2-86.
- Appendix B.

AFTER (Hold Program)

MCP Control Instruction

Access Level: 1

The AFTER command holds execution of a program until the completion of a specified program.

```
- AFTER | <program name> _____ |
        | <program name>/<compiler name> |
        | (<backup file ID>)/<intrinsic ID> |
```

Explanation

<program name> This designates the name of an object code file. A program name can have up to six characters. The first character must be a letter of the alphabet.

<program name>/<compiler name> This can be used when a program is to be initiated AFTER a compilation and it is necessary to specify a particular compilation by including the compiler name.

<backup file ID>/<intrinsic ID> This specifies an intrinsic. A printer backup file number is represented by an at sign (@) followed by five digits.

If a program is to be initiated after a program that has used the PROGRAM CALL BCT to call SORT; or MERG;, the AFTER command should specify only the program name, not <backup file ID>/<intrinsic ID>.

Details

When the MCP encounters AFTER in the control string, the MCP locates the job with the highest processor priority in the mix and the requested <program name>, and attaches the job to be initiated to that program. If the specified job is not in the mix, the schedule is examined for the requested program (by schedule priority). If a match is not found, the first job entering the schedule with the requested name is used as the <program name>.

The program to be initiated is scheduled but not brought into the mix until <program name> has come to a normal End-Of-Job (EOJ).

There is no limitation on the length of time the program waits.

If <program name> or <program name>/<compiler name> does not go to a normal EOJ (if, for example, it has been discontinued by a DS (Discontinue

Program) command or the compilation of <program name> contains syntax errors), the MCP does one of the following:

- If the AFTR option is reset (off), the program is left in the schedule until a program of the specified name reaches a normal EOJ. The operator can remove the job from the schedule with the RS (Remove Job From Schedule) command or allow the job to enter the mix with the RA (Remove After-Linkage) command.
- If the AFTR option is set (on), the program is automatically removed from the schedule. In addition, any jobs linked to this job through the AFTER command are removed from the schedule.

Examples

```
COMPILE PROGX COBOL9; AFTER PROGY
```

This syntax compiles PROGX with the COBOL9 compiler after the program PROGY finishes execution.

```
PFM DSKMTP SOLT SOURCE/TAPE 80 9; AFTER PROGX/COBOL9
```

This initiates the DMPALL intrinsic to reproduce the data file called SOLT on disk to a tape called SOURCE and file named TAPE, with a record length of 80 and a block size of 9, after the program PROGX is compiled.

```
EX PROGX; AFTER PROGZ
```

This holds the execution of PROGX until PROGZ finishes executing.

See also

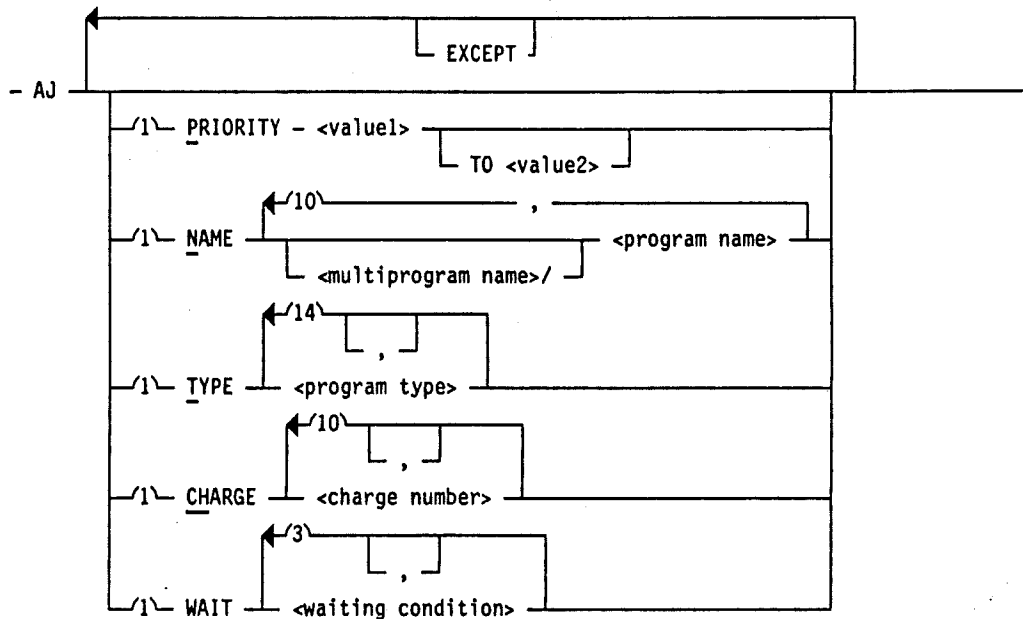
- The RA (Remove After-Linkage) command on Page 2-312.
- The RS (Remove Job from Schedule) command on Page 2-333.

AJ (Display Active Jobs)

Keyboard Input Message

Access Level: 0

The AJ command lists jobs that are currently executing. You can list a subset of the active jobs by using syntax options of the command.



Explanation

PRIORITY <value1>
TO <value2> Selects jobs based on processor priority. You have the option of selecting a single priority or a range of priorities. The values of <value1> and the optional <value2> must be integers in the range of 1 through 9, <value1> must be less than or equal to <value2>.

NAME
<multiprogram name> /
<program name> Selects jobs based on the program name and (optionally) on the multiprogram name. Separate the multiprogram name from the program name with a slash (/). You can enter a maximum of 10 name selections, separated by commas.

You can also select jobs based on part of a name by using the equal sign as a masking character. Use an equal sign for any unknown characters in the name. For example, the following command displays all programs in the mix beginning with the three

characters DMP, but does not display the program DMPOUT.

AJ NAME DMP= EXCEPT NAME DMPOUT

TYPE <program type> Selects jobs based on the type of program. You can enter a maximum of 14 type selections. The available program types are:

<u>B</u> NAHNDL	Program is a BNA handler.
<u>C</u> OMPILER	Program generates object code.
<u>C</u> OMPSHAR	Program that generates object code, executed as a timesharing task.
<u>C</u> OPY	Program is SYSTEM/COPY intrinsic.
<u>D</u> MPELL	Program is DMPALL intrinsic.
<u>D</u> MSPROG	Program is a DMSII control program (for example, DBP).
<u>D</u> SKOUT	Program is DSKOUT intrinsic or diskpack SQUASH intrinsic.
<u>E</u> XECUTE	Program is executing.
<u>H</u> ANDLERS	Program is any type of handler (BNA, WFL, time sharing).
<u>L</u> OADMP	Program is LOADMP intrinsic.
<u>M</u> CS	Program is Message Control System (MCS).
<u>P</u> ACKUP	Program is PACKUP intrinsic.
<u>S</u> HAREHNDL	Program is a timesharing handler program (for example, CANDE).
<u>S</u> ORT	Program is SORT: or SORT. intrinsic.
<u>T</u> IMESHARE	Program is executed as time sharing task.
<u>W</u> FLHNDL	Program is a Work Flow Language (WFL) handler.
<u>U</u> TILITY	Program is any type of intrinsic (DMPALL, LOADMP, PACKUP, DSKOUT, SYSTEM/COPY).

CHARGE
<charge number>

Selects jobs based on the charge number under which the job is executed. You can enter as many as 10 charge number selections and 10 charge number exceptions.

You can also select jobs based on part of a charge number by using the equal sign as a masking character. Use an equal sign for any unknown characters in the charge number. For example, the

following command displays jobs executed under a charge number beginning with the characters 66, but does not display jobs executed under charge numbers 660001 or 669999.

AJ CH 66= EXCEPT CH 660001, 669999

WAIT
<waiting condition>

Selects jobs that are waiting for a specific condition to be satisfied (for example, waiting for hardware, waiting for memory, and so forth). You can enter as many as three waiting conditions and three waiting condition exceptions.

The available types of waiting conditions are

<u>DEVICE</u>	Waiting for hardware device to become available
<u>EVENT</u>	Waiting for a software event to be signaled (for example, STOQUE, CRCR, and so forth)
<u>OPERATOR</u>	Waiting for operator to perform an action
<u>SPACE</u>	Waiting for memory, disk space or diskpack space

EXCEPT

All text following the keyword EXCEPT is treated as exceptions to the selections entered previously. All of the exceptions must be grouped together after the keyword EXCEPT; you cannot enter a selection and an exception followed by another selection.

Details

The AJ command corresponds to the AD (Specify ODT/OCS Screen Parameters) command <table> entry AM. The AM table selection displays the task number, processor priority, memory priority, and the amount of memory required for each job active in the mix.

If there are no active jobs, a "NULL MIX" response will be displayed.

See also

- The AD (Specify ODT/OCS Screen Parameters) command on Page 2-4.
- The MX (Display Jobs In Mix) command on Page 2-236.
- The WJ (Display Waiting Jobs) command on Page 2-411.
- The WS (Display Jobs In Schedule) command on Page 2-423.

ALLOCATE (Create Disk Space and Directory Entries)

MCP Control Instruction

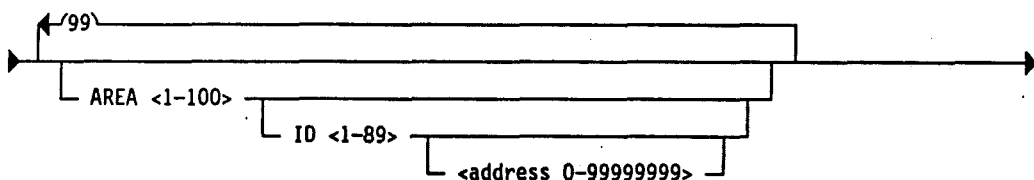
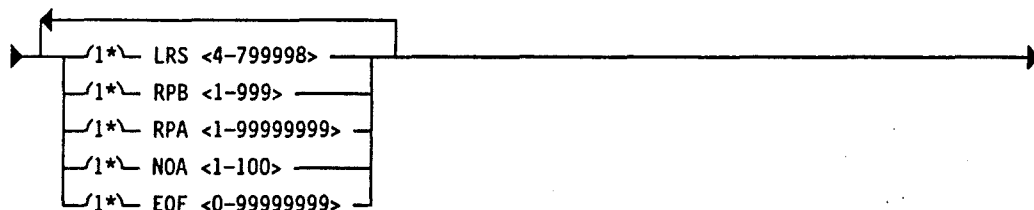
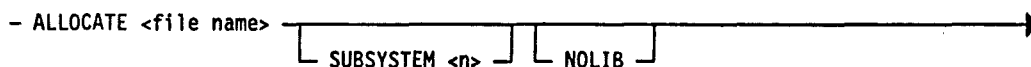
Access Level: 8

ALLOCATE creates disk directory entries for files that can be loaded using the ALLOCATE feature of SYSTEM/COPY, the file transfer utility.

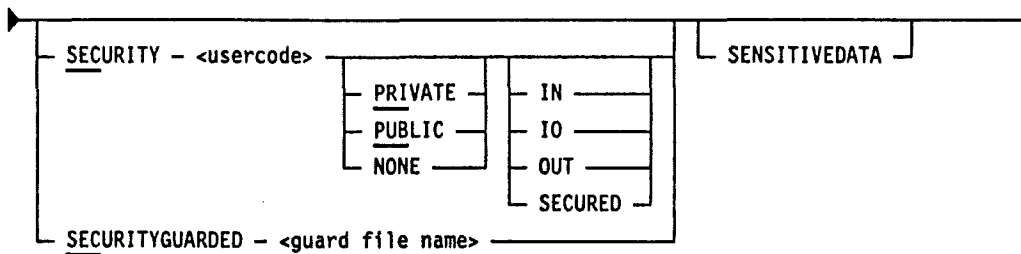
A number of optional elements permit considerable flexibility in the assignment of disk space for the file. You can also assign security attributes for the file.

The syntax for this command is identical to that for the ALLOCATE record of the system configuration file. See the *V Series Systems Operations Guide Volume 1: Installation* for more information.

You cannot use more than one ALLOCATE command for each file name.



ALLOCATE



Explanation

- <file name>** This indicates the name of the file.
- SUBSYSTEM <n>** This allocates the file to a specific logical disk subsystem. Logical subsystem numbers range from 1 through 8.
- If you omit this option, the file is allocated to the default logical subsystem.
- NOLIB** This marks the file NO LIBRARY MAINTENANCE to prevent unwanted removal or changing of the file.
- LRS <4-79998>** This indicates the logical record size in digits. The value can range from 4 through 79998 digits and must be an even number.
- RPB <1-999>** This is the number of records per block. The size of a block (the number of records per block times the logical record size) is restricted only by the amount of available memory. The value can range between 1 through 999.
- RPA <1-99999999>** This is the number of records per area. The value can range from 1 through 99,999,999.
- NOA <1-100>** This indicates the file's number of areas. The maximum file size (the number of areas times the number of records per area) cannot exceed 99,999,999. The value can range from 1 through 100.
- EOF <0-99999999>** This indicates the end of file pointer. The value must fall within one of the areas allocated through an AREA specification or it must be zero.
- AREA <1-100>** This assigns areas of the file to specific disk devices and addresses. If you specify AREA, the file can be changed or removed through system commands, but it cannot be programmatically closed with a PURGE or with a REMOVE. Files with AREA specifications that include actual addresses are not squashed.

Allocating a multiple-area file requires all AREA allocations to be on one ALLOCATE command.

The AREA specifications are optional if the EOF value is 0. If used they cause pre-allocation of disk space for the file. You can request a specific disk address, or a specific ID, or you can allow the MCP to perform the allocation (according to any DISK record constraints).

ID <1-89>	This is the disk device to which the area of the file is allocated. If an address is used with this option, the allocated areas start at that address.
<address 0-999999>	This the specific address on the designated disk drive at which the file or sector will begin.
SECURITY <usercode>	This allows access to this file only when the specified usercode is given.
PRIVATE	This allows only the creator access the file. This is the default security setting for ALLOCATED files.
PUBLIC	This allows any user to access the file.
NONE	This means that the file has no security assignment.
IN	This allows data to be read from the file but not written to it.
IO	This indicates that data can be read from and written to the file.
OUT	This indicates that data can be written to the file but not read from it.
SECURED	This means that the file is a program and can only be executed. Data cannot be read from or written to this file.
SECURITY GUARDED <guard file name>	This specifies that a special file must be accessed before this file can be accessed. The name of the guard file is <guard file name>. The guard file resides on disk.
SENSITIVEDATA	This indicates that the file contains sensitive data. When the file is removed, its data will be replaced with random data.

ALLOCATE

Example

```
ALLOCATE COBTST LRS 200 RPB 1 RPA 500 NOA 20 EOF 0 SECURITY 8196 PUBLIC IO
```

This allocates disk space for the file called COBTST. COBTST's physical and logical characteristics include a logical record size of 200, with 1 record per block. COBTST has a maximum of 20 areas and 500 records per area. Any user can access COBTST. Data can be read from and written to the file.

```
ALLOCATE FILEZ LRS 200 RPB 100 NOA 2 EOF 200 RPA 100 AREA 1 ID 8 100000  
AREA 2 ID 8 100100
```

This allocates disk space for the file FILEZ. FILEZ's physical and logical characteristics include a logical record size of 200, with 100 records per block. FILEZ has 2 areas and 100 records per area. Area 1 is assigned to disk ID 8 beginning at sector 100000; area 2 beginning at sector 100100.

See also

- The COPY (Invoke File Transfer Utility) command on Page 2-86.
- The ALLOCATE record in the *V Series Systems Operations Guide Volume 1: Installation*.

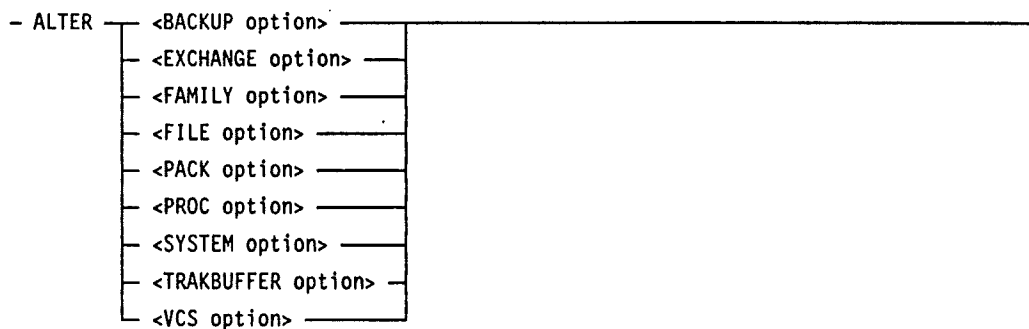
ALTER (Change System Parameters)

Keyboard Input Message

Access Level: 8

The ALTER command lets you change several parameters that deal with the system environment.

The following diagram shows the syntax of the ALTER command.



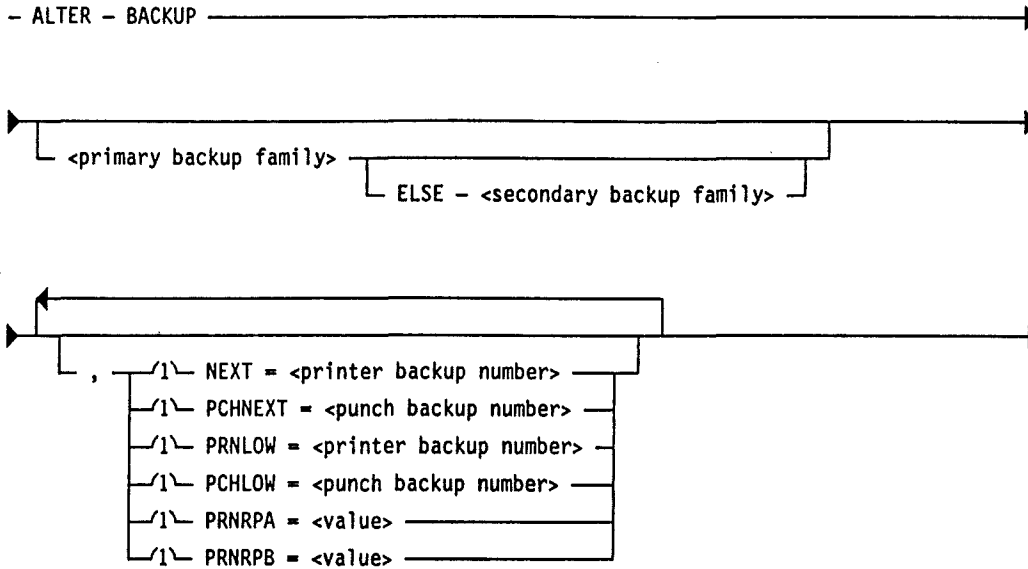
These options are described on the following pages.

BACKUP Option

The BACKUP option can be used to change the following:

- The families where backup files are directed by default
- The next number assigned to printer and punch backup files
- The lowest number that the operating system uses when searching for printer and punch backup files
- The number of records per area and number of records per block for printer backup files.

The following diagram shows the syntax for the BACKUP option of the ALTER command.



Explanation

<primary backup family> This is the family name of the diskpack family where the operating system stores printer backup files by default.

<secondary backup family>

This is the family name of the diskpack family where the operating system stores printer backup files if the primary backup family does not have enough space or is unavailable. If no secondary backup family is declared, the operating system stores backup files on any available system resource diskpack.

- NEXT** = <printer backup number> This sets the number that is assigned to the next printer backup file generated on the system. The <printer backup number> must be greater than 0.
- PCHNEXT** = <punch backup number> This sets the number that is assigned to the next punch backup file generated on the system. The <punch backup number> must be greater than 0.
- PRNLOW** = <printer backup number> This sets the lowest number that operating system intrinsics (for example, BFRF or PBDPRN) use when they search for printer backup files. The <printer backup number> must be greater than 0.
- PCHLOW** = <punch backup number> This sets the lowest number that operating system intrinsics (for example, BFRF or PBDPRN) use when they search for punch backup files. The <punch backup number> must be greater than 0.
- PRNRPA** = <value> This sets the number of records in each area of a printer backup file. This value must be a multiple of 10 and a multiple of the PRNRPB value. If it is not a multiple of PRNRPB, the operating system will adjust it upward to the nearest multiple. The default value is 3150.
- PRNRPB** = <value> This sets the number of records in each block of a printer backup file. This value must be a multiple of 10. The default value is 10.

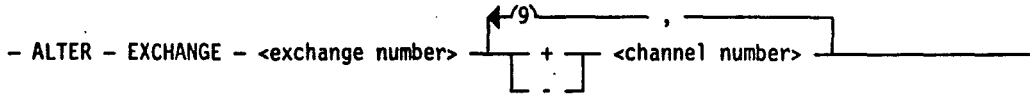
Example

```
ALTER BACKUP BKUP, NEXT = 0100
```

This changes the default backup family to BKUP, and assigns 0100 as the next backup file number.

EXCHANGE Option

The EXCHANGE option adds or removes a channel from an exchange. The following diagram shows the syntax for the EXCHANGE option of the ALTER command.



Explanation

<exchange number>

This is the number of the exchange from which you are adding or deleting a channel.

<channel number>

This is the number of the channel that you are adding or deleting.

Example

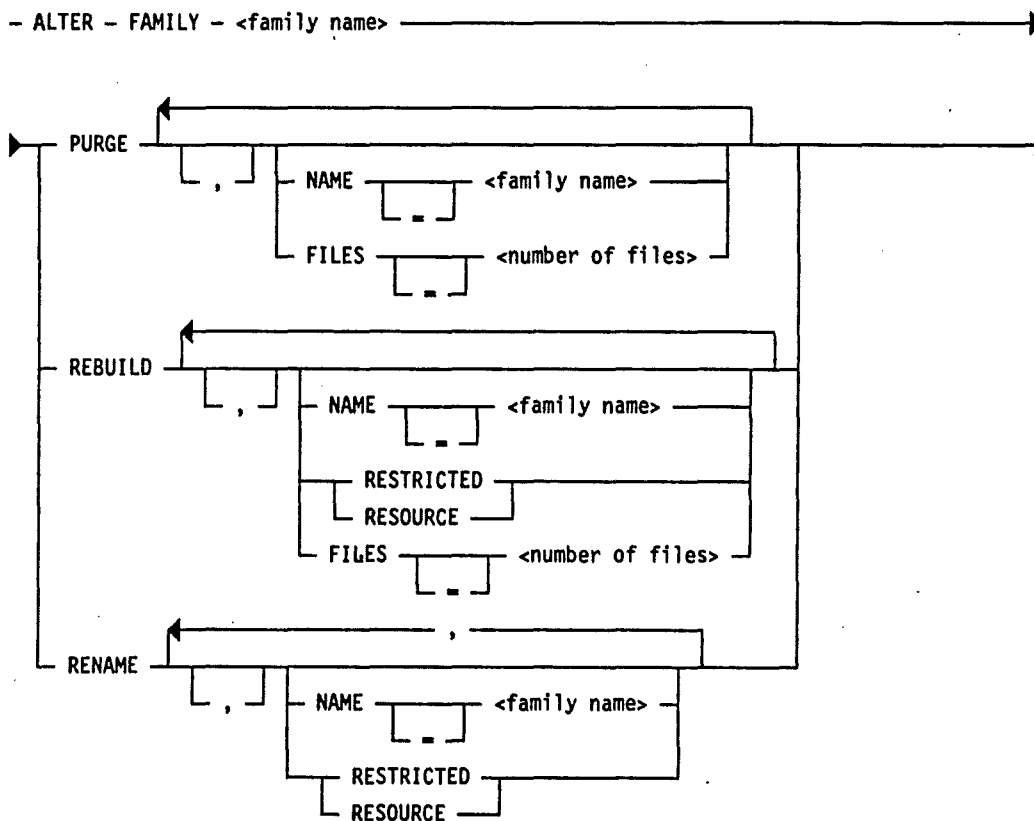
ALTER EXCHANGE 4 + 14

This adds channel 14 to exchange number 4.

FAMILY Option

The FAMILY option modifies pack structures for an entire family. Before this command can be used, all units must be saved. When the command is issued, it must be confirmed with an OK.

Note: The FAMILY option is valid only against Version 2 (or greater) pack families. For more information on pack family versions, refer to "Pack Subsystems" in the V Series Systems Operations Guide Volume 3.



Explanation

<family name>

This is the name of the diskpack family that is to be modified. The following describes the three actions that can be performed on a family.

Purging a Diskpack Family (PURGE)

Purging a diskpack family purges (removes) all files from the diskpack family, compacts the Master Available table, and creates a new Available table on each unit of the diskpack family. An empty directory is created on the base diskpack.

ALTER

If ALTER FAMILY PURGE is entered from an ODT/OCS, the system will issue the following message to the ODT/OCS:

```
ALL FILES ON FAMILY WILL BE LOST!  
PURGE FAMILY <family name>? (OK OR QUIT)
```

To proceed with the purge of the diskpack family, enter OK. To abort the purge, enter QUIT.

If any continuation diskpacks for the family are offline, their Available tables are not rebuilt. However, when the diskpack is placed online, all previously-assigned sectors will be unavailable. This is because the file headers for the diskpack, which were stored on the base diskpack, are no longer available. Should this condition arise, the only alternative is to use the ALTER NEW PACK command to rebuild the diskpack label, the Master Available table, and the Available table, and add the continuation diskpack into the family.

The following options are available.

NAME <family name> This causes the family name in the label to be updated.

FILES <number of files> This specifies the approximate number of files to be stored on each diskpack of the family. The <number of files> can range from 1 through 999,999. If <number of files> is not specified, the system reserves initial directory space for 1,100 files.

This value causes the appropriate number of contiguous directory entries to be constructed on the base pack of the family. This minimizes head movement during directory lookup operations. If required, the number of entries in the directory will be expanded.

Rebuilding a Diskpack Family (REBUILD)

Rebuilding a diskpack family purges (removes) all files and rebuilds all pack structures for the diskpack family. A label, a Master Available table, and an Available table are built for all diskpacks. All previously removed sectors (removed with the XP command) are ignored. An empty directory is created on the base diskpack.

If ALTER FAMILY REBUILD is entered from an ODT/OCS, the system will issue the following message to the ODT/OCS:

```
ALL FILES ON FAMILY WILL BE LOST!  
REBUILD FAMILY <family name>? (OK OR QUIT)
```

To proceed with the rebuild of the diskpack family, enter OK. To abort the rebuild, enter QUIT.

When a family is rebuilt, a new base pack timestamp is established and updated on all diskpacks of the family that are online. If a continuation diskpack of the family is offline, its timestamp is not updated. When the diskpack is placed online, because its timestamp will not match the timestamp of the base pack it will not be recognized as a member of the family. You must use the ALTER NEW PACK command to add the diskpack to the family.

The following options are available.

- NAME <family name>** This causes the family name in the label to be updated.
- RESTRICTED** This specifies that the family is restricted. If a family is restricted, the family name must be specified to access any of its files.
- RESOURCE** This specifies that the family is a resource family. If a family is a resource family, the family name is not required to access any of its files. If neither RESTRICTED nor RESOURCE is specified, the existing specification for the family is used.
- FILES <number of files>** This specifies the approximate number of files to be stored on each diskpack of the family. The <number of files> can range from 1 through 999,999. If <number of files> is not specified, the system reserves initial directory space for 1,100 files.
- This value causes the appropriate number of contiguous directory entries to be constructed on the base pack of the family. This minimizes head movement during directory lookup operations. If required, the number of entries in the directory will be expanded.

Renaming a Diskpack Family (RENAME)

Renaming a diskpack family changes various fields in the pack label without purging the pack, and without changing any of the tables.

When a family is renamed, new information is placed in the label of the base pack and in the labels of all diskpacks of the family that are online. If a continuation diskpack of the family is offline, its label is not updated. When the diskpack is placed online, because its label will not match the label of the base pack it will not be linked into any family list. You must use the ALTER NEW PACK command to add the diskpack to the family.

The following options are available.

- NAME <family name>** This causes the family name in the label to be updated.

ALTER

RESTRICTED

This specifies that the family is restricted. If a family is restricted, the family name must be specified to access any of its files.

RESOURCE

This specifies that the family is a resource family. If a family is a resource family, the family name is not required to access any of its files.

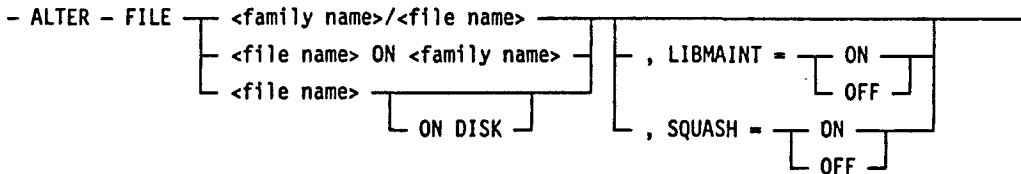
If neither RESTRICTED nor RESOURCE is specified, the existing specification associated with the family is used.

FILE Option

The FILE option changes the LIBMAINT status and SQUASH status of a file.

If the LIBMAINT status is OFF, the file cannot be changed (using the CHANGE (Change File ID) command), written to, or removed.

SQUASH refers to consolidation of areas on a designated diskpack. If the SQUASH status is OFF, the file is *not* moved as a result of the SQP (Squash Pack) command.



Explanation

<file name>

This is the name of the file. FILE <file> indicates a disk file.

<family name>

This is the diskpack family name, if applicable. You could use FILE <file name> ON DISK to indicate a disk file.

LIBMAINT

This switches the LIBMAINT status ON or OFF.

SQUASH

This switches the SQUASH status ON or OFF.

Example

```
ALTER FILE NEWMBB, LIBM = OFF
```

This switches the LIBMAINT status of the file NEWMBB off.

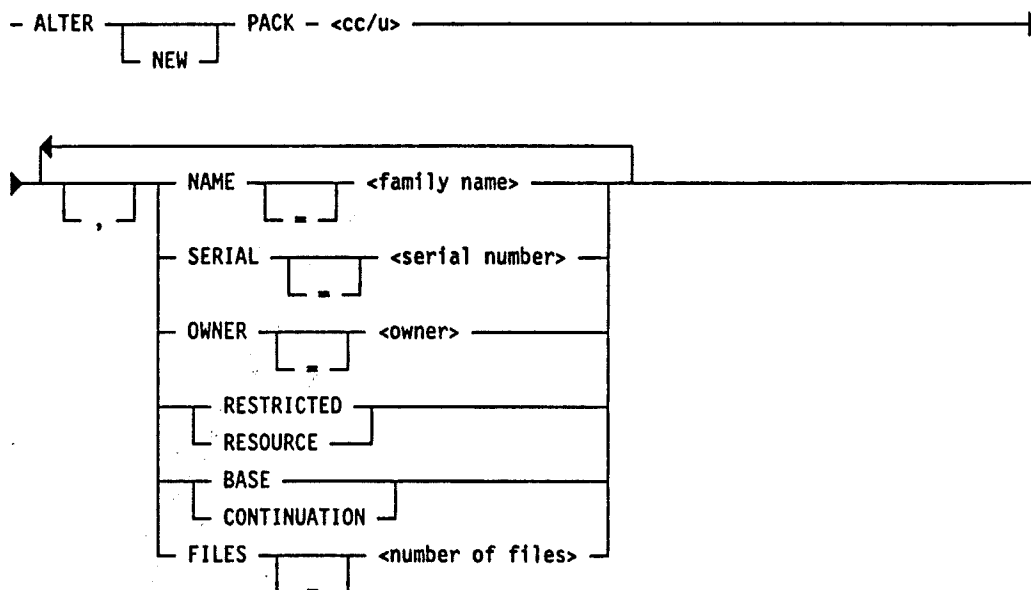
PACK Option

The PACK option changes or builds the label, Master Available table, or the Available table of a Version 1 diskpack. Also, the PACK option can be used to rename a Version 1 or a Version 2 diskpack.

The NEW PACK option builds the label, Master Available table, or the Available table of a Version 2 diskpack.

The diskpack device must be saved (using the SV (Save A Peripheral Unit) command) before you can perform this operation.

Note: Many of the ALTER PACK options are designed for Version 2 (or greater) diskpacks. For information on pack family versions and the process of converting Version 1 diskpacks to Version 2 diskpacks, refer to "Pack Subsystems" in the V Series Systems Operations Guide Volume 3.



Explanation

NEW

This specifies that a new label, Master Available table, and Available table are to be constructed for the diskpack. This action is valid only for creating Version 2 diskpacks.

If ALTER NEW PACK is entered from an ODT/OCS, the system will issue the following message to the ODT/OCS:

```

ALL FILES ON FAMILY WILL BE LOST!
ALTER NEW PACK <cc/u>? (OK OR QUIT)
  
```

To proceed with the creation of the diskpack, enter OK. To abort the command, enter QUIT.

ALTER

- <cc/u>** This specifies the channel number and unit number of the diskpack device to be altered.
- NAME <family name>** This causes the family name in the label of the diskpack to be updated.
- SERIAL <serial number>** This specifies a serial number to be placed in the label of the diskpack. The <serial number> must be a unique value in the range of 1 through 999999.
- OWNER <owner>** This specifies an owner identification code to be placed in the label of the diskpack. The <owner> can be up to 14 characters in length, and cannot contain embedded blanks. If an <owner> is not specified, the owner identification code of the diskpack is set to all blanks.
- RESTRICTED** This specifies that the diskpack is a member of a restricted family. If a family is restricted, the family name must be specified to access any of its files.
- RESOURCE** This specifies that the diskpack is a member of a resource family. If a family is a resource family, the family name is not required to access any of its files. If neither RESTRICTED nor RESOURCE is specified, the diskpack is assumed to be part of a resource family.
- BASE** This specifies that the diskpack is the base pack of a family.
- CONTINUATION** This specifies that the diskpack is a continuation pack of a family. If neither BASE nor CONTINUATION is specified, the diskpack is assumed to be a continuation pack.
- FILES <number of files>** This specifies the approximate number of files to be stored on each diskpack of the family. The <number of files> can range from 1 through 999999. If <number of files> is not specified, the system reserves initial directory space for 1,100 files.
- This value causes the appropriate number of contiguous directory entries to be constructed on the base pack of the family. This minimizes head movement during directory lookup operations. If required, the number of entries in the directory will be expanded.

Example

ALTER PACK 4/14, NAME = NEWPAK, SERIAL = 123456

This changes the name and serial number of a diskpack device with channel number 4, unit number 14.

PROC Option

The PROC option starts or stops running a specified processor.

- ALTER - PROC + <processor number>

Explanation

- | | |
|--------------------|---|
| + | This starts running the processor or processors specified by <processor number>. The processor you want to start running must be online prior to entering the ALTER PROC + command. |
| - | This stops running the processor or processors specified by <processor number>. |
| <processor number> | This indicates the number of the processor you want to start or stop running. For the ALTER PROC command, processor numbers start with one. |

Setting the Processor Online

To set a processor online, enter the following Maintenance Processor (MP) command *before* entering the ALTER PROC + command:

SET ONLINE P<n>

The <n> is the processor number. For the SET ONLINE command, processor numbers start with zero.

ALTER

Example

```
SET ONLINE P0
```

When this is entered from the Maintenance Processor, the specified processor is placed online. The following command is then entered.

```
ALTER PROC +1
```

This starts the specified processor.

Setting the Processor Offline

To set a processor offline, *after* you enter the ALTER PROC - command, enter the following Maintenance Processor (MP) command:

```
SET OFFLINE P<n>
```

The <n> is the processor number. For the SET OFFLINE command, processor numbers start with zero.

Examples

```
ALTER PROC - 2
```

This stops the specified processor. The following command is then entered from the Maintenance Processor.

```
SET OFFLINE P1
```

This places the specified processor offline.

For more information about the Maintenance Processor commands, see the *V Series Systems Operations Guide Volume 1: Installation*.

System Responses

The system responds to the ALTER PROC command by confirming that a processor is started or stopped.

If you try to stop the last processor, the following message appears:

```
**KBD IGNORED: LAST REMAINING PROCESSOR <n>
```

Example

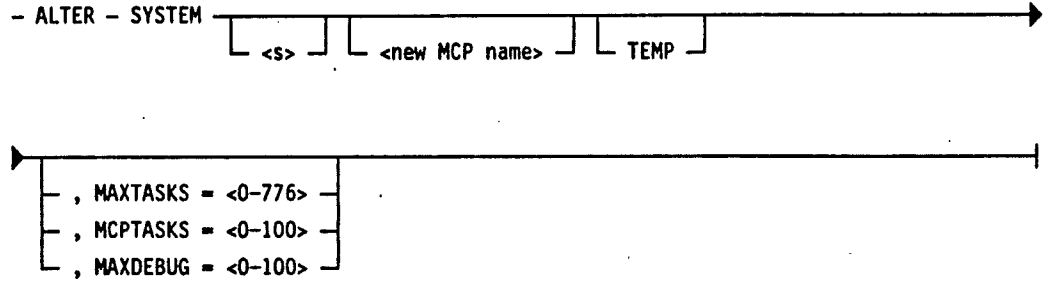
```
ALTER PROC +2
```

This starts processor 2 running if it is already online.

SYSTEM Option

The SYSTEM option changes the current MCP file or any of the special cold start constants. If a new MCP name is specified, you must first use the ALTER FILE option to set the LIBMAINT status of the new MCP file to OFF.

Note that multiple MCPs in a loosely coupled shared environment must be boot-compatible.



<s>

This is an optional system number and indicates in a multisystem shared configuration that only the specified system is to be affected.

If <s> is omitted, the MCP common to all systems is changed. Otherwise the MCP for system <s> is changed.

<new MCP name>

This specifies the new MCP file. The file must be on disk ID 01 and have LIBMAINT OFF (see the ALTER FILE option).

TEMP

This causes the new MCP file to be loaded with the next halt/load. The new MCP file will run until there is another halt/load; the original MCP file will then be reloaded automatically.

MAXTASKS = <0-776>

This alters the maximum number of tasks allowed in the mix and schedule. The number must be in the range 0 through 776.

MCPTASKS = <0-100>

This alters the maximum number of independent runners (IRs) allowed on the system. Independent runners are special tasks initiated by the operating system to perform services for user tasks or for the operating system itself. The number must be in the range 0 through 100.

MAXDEBUG = <0-100>

This alters the maximum number of debug sessions allowed. The number must be in the range 0 through 100.

ALTER

Examples

```
COPY MCPVS AS NEWMCP TO DISK (ID 01)
ALTER FILE NEWMCP, LIBM = OFF
ALTER SYSTEM NEWMCP, MAXTASKS = 200
```

This changes the current MCP file for all systems to the file NEWMCP, and establishes a MAXTASKS parameter of 200.

```
COPY MCPVS TO DISK (ID 01)
ALTER FILE MCPVS, LIBM = OFF
ALTER SYSTEM 2 MCPVS
```

This changes the MCP for System 2 to use MCPVS on disk. If four loosely coupled shared systems were using a common MCP before this was issued, the systems other than System 2 will continue to use the common MCP, while System 2 uses a special MCP.

TRAKBUFFER Option

The TRAKBUFFER option changes the size of the TRAK buffer. The TRAK buffer is used by the TRAK utility to store various operating system debugging information.

The ALTER TRAKBUFFER command is valid only when the TRAK system option is reset.

```
- ALTER - TRAKBUFFER [ ] <TRAK buffer size> _____ |
```

Explanation

<TRAK buffer size> This is the size to which the TRAK buffer is to be set. This value can range from 1 through 999, inclusive.

Information on the current TRAK settings can be displayed using the SHOW MLS command.

VCS Option

The VCS option changes VCS attributes. Currently, only the name of the VCS initialization file can be changed. The ALTER VCS INITFILE has no affect until VCS is initialized.

```
- ALTER - VCS - INITFILE [ ] <file name> [ ON <medium name> ]
```

Explanation

- = This is optional and does not affect the command.
- <file name> This is the name of the new VCS initialization file. The new VCS initialization file is used for any subsequent initialization of VCS.
- ON <medium name> This refers to the name of the storage device where the VCS initialization file is stored. The <medium name> can be one of the following choices:
- | | |
|------------------------|---|
| DISK | This looks for the VCS initialization file on 100-byte media only. If the file is not found, an error message is displayed on the ODT. |
| PACK | This looks for the VCS initialization file on all of the system's unrestricted diskpicks. If the file is not found, an error message is displayed on the ODT. |
| A diskpack family name | This looks for the VCS initialization file on the diskpicks of the entered family name. If the file is not found, an error message is displayed on the ODT. |

If <medium name> is not entered, a default medium of DISK is used.

See also

The SHOW (Display System Information) command on Page 2-343.

AT (Initiate BNA Job Transfer)

MCP Control Instruction

Access Level: 0

AT directs a system command to a remote host.

The AT command initiates a BNA job transfer dialogue (the transfer of job files) between the local host and a specified remote host.

Various commands can be included in this job transfer. Any number of jobs can be transferred to the same host in one job transfer process, and multiple jobs can execute simultaneously.

This command is valid only from an actual or pseudo card deck.

- AT - <host name> _____

Explanation

<host name> This indicates the remote host to which the job is transferred.

Details

Each record following "?AT" in the pseudo deck is checked to determine if it is a control record, recognizable by a question mark (?) in the first character position.

Each control record is further checked for an END command indicating the end of a job. Each data record or control record is transferred to the remote host until an ENDAT record terminates the dialogue. Each record deck starts with a ?AT <hostname> record and ends with a ?ENDAT record. (If the deck is from a pseudo reader, the ?ENDAT record is optional.)

Commands can be concatenated on one record; they must be separated by semicolons. There must not be commands following the hostname on the first record.

An ?END control record indicates the end of a job and is optional for the last job in the file. The MCP control instructions that can start an execution are

ADD	DEBUG	MERGE
CHECK	DIR	PERFORM
COMPARE	DUMP	PRINT
COMPILE	EXECUTE	UNLOAD
COPY	GENERATE	
COPYADD	LOAD	

The following keyboard input messages can also start an execution:

BF	KS	PP
CV	PA	PS
ID	PB	RF
KA	PC	
DP	PM	

These commands must be preceded by the word SPO. For example, ?AT HOSTA SPO CV5 is a valid SPO execution command.

One copy of the BNAJXR program enters the mix when a job is initiated with an AT command through the card reader or a pseudo card file. BNAJXR handles all commands entered that are preceded by AT, until an ENDAT is encountered. When ENDAT is encountered, BNAJXR terminates. Another AT command causes initiation of another copy of BNAJXR.

Commands preceded by the AT (Pass Message to Remote Host) keyboard input message are processed entirely by the BNAODT program. They do not cause initiation of BNAJXR.

The following are examples of job files:

Example 1

?AT YOURHOST	%NO COMMAND
PRM LIST1 INPUT CARD	%Beginning of Job 1
?DATA INPUT	%Control Record
	%
[DATA RECORDS]	%Data Records
	%
?END	%End of Job 1
?EX A ; DATA INPUT	%Beginning of Job 2
	%
[DATA RECORDS]	%Data Records
?END	%End of Job 2
?ENDAT	%ENDAT is optional

Example 2

?AT SGHOST	%No Command on first record
?COMPILE PGMA COBOL9 LIB	%Beginning of Job 1
?DATA CARD	%Control record
<data records>	%
?ENDAT	%

See also

- The AT (Pass Message to Remote Host) keyboard input message command on Page 2-38.
- The ENDAT (End Job Transfer) command on Page 2-157.

AT (Pass Message to Remote Host)

Keyboard Input Message

Access Level: 0

The local host recognizes the AT command as an ODT input message that is destined for a remote host. The local host does not check the message, but passes it to Host Services for transmission to the remote host. The remote host acts upon the message as though it were received from a local ODT subsystem and returns any responses to the originating host.

- AT - <host name> - <input message> _____|

Explanation

<host name> This identifies the remote host that will receive and act upon the message.

<input message> This is any valid command or inquiry for the remote host.

Commands entered with AT are processed by the BNA program BNAODT. Consequently, use of the AT command requires the presence of BNA on the system. If AT is used and BNA is not running, the MCP returns the following message:

NOT IN NETWORK MODE

Example

AT SYSTEM2 DQ 30

This causes a DQ command to be passed to a remote host named SYSTEM2.

See also

The AT (Initiate BNA Job Transfer) MCP Control Instruction command on Page 2-36.

AX (Pre-Respond to an ACCEPT)

MCP Control Instruction

Access Level: 1

The AX MCP control instruction places data into an ACCEPT queue for a program prior to its execution.

- AX - <data>

Explanation

<data>

This is the data saved for a program. The data can be up to 60 characters in length, and is subject to the following limits.

Alphanumeric Data: Data containing alphabetic characters, the digits 0 through 9, and spaces can be entered immediately following the AX. The data begins with the first non-blank character following the AX and must end with a period or ETX character. Detection of any character other than those previously described will cause a syntax error. The data can be enclosed in quotation marks, in which case the terminating period or ETX character is not required.

Non-Alphanumeric Data: Data that contains non-alphanumeric characters or has leading spaces must have the entire string of characters enclosed within quotation marks.

Note: AX can be intermixed with other job control commands, such as INSERT (Insert Data into Program). However, each AX instruction must be terminated with a semicolon (;). No characters other than spaces can separate the final quotation mark from the terminating semicolon.

Details

ACCEPT statements in a program cause the program to halt and wait for the data to be entered on the ODT. The ACCEPT function is available for BPL, COBOL ANSI-74, COBOL ANSI-68, FORTRAN, Pascal, RPG, and Work Flow Language (WFL) programs.

If the length of the data (including trailing spaces where quotes are not used) exceeds the size of the program's input buffer, the text is ignored.

If the text is shorter than the input buffer, the text is terminated with an ETX character in the program's ACCEPT area. The contents of the input buffer following the ETX are undefined.

Examples

```
EX COPY1; AX 'COPY AAAAjm AS ZZZZjm'; AX 'END'
```

This starts the program COPY1 and enters into the ACCEPT queue the data "COPY AAAAjm AS ZZZZjm" and then the data "END".

```
COMPILE PROG1 COBOL9; AX 'SET DEBUG'; INSERT 32 2 UN 66
```

This compiles PROG1 with the COBOL9 compiler and enters into the ACCEPT queue the data "SET DEBUG". It also inserts into the program at address 32 the data string 66, which is 2 digits of unsigned numeric (UN) data.

See also

- The AX (Respond to an ACCEPT) keyboard input message command on Page 2-41.
- The ACCEPT function in the various programming language manuals.

AX (Respond to an ACCEPT)

Keyboard Input Message

Access Level: 0

The AX keyboard input message enters data into a program in response to an ACCEPT message.

- <mix number> - AX - <data>

Explanation

<mix number>

This is the mix number of the program requiring the data.

<data>

This is the data passed to the program with the specified <mix number>. The data entered can be up to 60 characters in length, and is subject to the following limits.

Alphanumeric Data: Data containing alphabetic characters, the digits 0 through 9, and spaces can be entered immediately following the AX. The data begins with the first non-blank character following the AX and must end with a period or ETX character. Detection of any character other than those previously described will cause a syntax error. The data can be enclosed in quotation marks, in which case the terminating period or ETX character is not required.

Non-Alphanumeric Data: Data that contains non-alphanumeric characters or has leading spaces must have the entire string of characters enclosed within quotation marks.

Details

ACCEPT statements in a program cause the program to halt and wait for data to be entered on the ODT. The ACCEPT function is available for BPL, COBOL ANSI-74, COBOL ANSI-68, Fortran, Pascal, and Work Flow Language (WFL) programs.

If the length of the data (including trailing spaces where quotes are not used) exceeds the size of the program's input buffer, the text is ignored and an error message is displayed.

If the text is shorter than the input buffer, the text is terminated with an ETX character in the program's ACCEPT area. The contents of any characters in the input buffer following the ETX are undefined.

An AX command can be entered for a program before the display of the ACCEPT message. If this is done, the MCP saves the data until the program issues the ACCEPT. In this case, if the length of the data entered exceeds the size of the program's input buffer, the data is truncated and no error message is displayed.

Examples

2 AX YES

This responds to an ACCEPT message from the program associated with mix number 002 with the alphanumeric data YES.

15 AX ''032080''

responds to an ACCEPT message from the program associated with mix number 015 with the non-alphanumeric data "032080".

See also

- The AX (Pre-Respond To An ACCEPT) MCP control instruction command on Page 2-39.
- The ACCEPT function in the various programming language manuals.

BCL (Mark Beginning of Card Data File)

MCP Control Instruction

Access Level: 0

The BCL command indicates the beginning of a punched card data file.

Whenever the BCL command is encountered in the card reader, it causes an End-Of-File (EOF) condition on any previous file. The MCP marks the unit as containing the new file.

This command is valid only from an actual or a pseudo card deck.

```

┌ BCL ───┐ ┌ <file name> ───────────────────────────────────┐
└  DATAB ┘ └ CTLDCK ───┘
  
```

Explanation

BCL
DATAB

BCL and DATAB are synonymous.

The word BCL specifies that subsequent input is in BCL format. BCL format is a coded character set in which each character is represented by a six-bit code.

<file name>

This is the name of the file to be read.

Any information after the <file name> on the same record is ignored.

CTLDCK

is a reserved file identifier that allows valid control records to be included in the file. CTLDCK is the name of the physical card deck used by the LDCNTL utility to build a pseudo card deck.

Note: If the <file identifier> is CTLDCK, no EOF action is taken until the ?ENDCTL control instruction is encountered. This is primarily intended for use by the utility program LDCNTL for building pseudo card decks.

Example

```
?BCL CARDS
```

This specifies the beginning of a punched card data file called CARDS.

See also

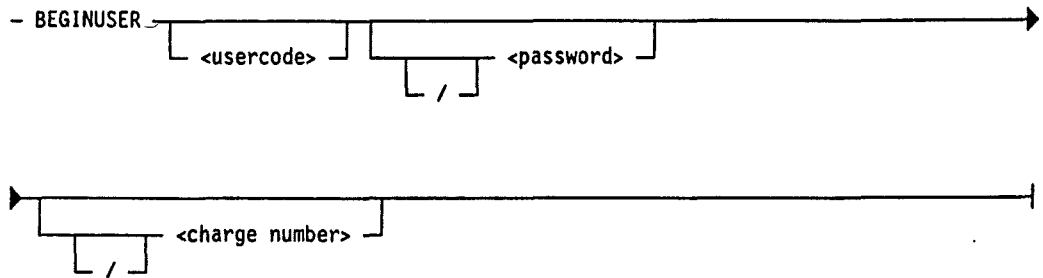
- The **BINARY (Read Binary Card Deck)** control instruction on Page 2-57.
- The **DATA/DATAB (Mark Beginning Of Card Data File)** control instruction on Page 2-105
- The **EBCDIC (Mark Beginning Of Card Data File)** control instruction on Page 2-153.
- The **END/ENDCTL (Mark End of Card File or Control Instructions)** control instruction on Page 2-156

BEGINUSER (Mark Beginning of Job Stream)

MCP Control Instruction

Access Level: 0

The BEGINUSER command is used for system security. This command delimits the beginning of a job stream and assigns a <usercode>/<password>/<charge number> access combination to all function requests in that job stream.



Explanation

<usercode>	This refers to a code that identifies a user to the system.
<password>	This refers a group of characters by which the system determines the legality of a usercode.
<charge number>	This refers to a number used to assign charges for system use.

Note: The <usercode>, <password>, and <charge number> are all optional. Slashes are required only if leading parameters are omitted (for example, BEGINUSER //55943).

Details

The BEGINUSER instruction can be entered whether or not the device from which it is entered is logged in.

The access combination specified in the BEGINUSER command is assigned to all functions initiated from the device until that combination is terminated by a matching ENDUSER (Mark End of Job Stream) command with the same access combination. Alternatively, the specified access combination can be overridden for any single function initiated from the device by preceding the function request with a USER (Assign Access Code) command.

BEGINUSER and ENDUSER commands can be nested. However, the access combination specified on the ENDUSER instruction must always match the access combination of the last BEGINUSER instruction.

BEGINUSER

Examples

BEGINUSER //92936

This marks the beginning of a job stream, assigning a charge number of 92936 to all function requests in that job stream.

BEGINUSER 6720/PASSWORD/67073

This marks the beginning of a job stream, assigning a usercode of 6720, password of PASSWORD, and charge number of 67073 to all function requests in that job stream.

See also

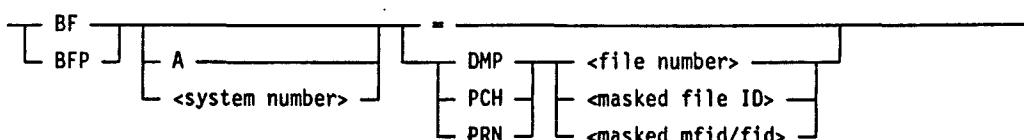
- The ENDUSER (Mark End of Job Stream) command on Page 2-158.
- The USER (Assign Access Code) command on Page 2-401.
- The *V Series System Software Security Installation and Operations Guide* for information about <usercode>, <password>, and <charge number>.

BF, BFP (List Backup Files on Disk or Pack)

Keyboard Input Message

Access Level: 0

This command lists backup files according to the systems that created them, their backup numbers, their file names, or their multifile or multiprogram identifier.



Explanation

BF BF lists backup files on disk; **BFP** lists backup files on diskpack.

A This lists backup files produced by all systems.

<system number> This lists backup files produced by a specified system.

If the character A or a system number is not used, the system will list only the files created on the system through which this request is made.

= These specify the type of backup files to be listed. The following defines the meaning of each option.

=	Lists all types of backup files
DMP	Lists dump files
PCH	Lists punch backup files
PRN	Lists printer backup files.

Note that you cannot use the BFP command to list dump files, as dump files are not written to diskpack. If you do, the message "****KBD IGNORED: INV SYNTAX" will be displayed.

<file number> A **<file number>** lists only the name of the file with that number.

If you use a **<masked file name>**, the MCP lists all the backup files with that name. For PRN and PCH files, the name is a multifile identifier/file identifier or file identifier; for DMP files, the name is a program name that can be of the form multiprogram identifier/program identifier.

Details

For printer and punch files, BF or BFP displays the total number of records in each file and the names of the programs that created them.

Examples

BF =

This lists all the backup files on disk created by the system through which this request is made.

BF PRN PRINT

This lists the printer backup files on disk named PRINT created by the system through which this command was issued.

BF DMP PROGA=

This lists all the memory dump files with names beginning with PROGA.

BF PRN ABC=

This lists all printer backup files on disk with names beginning with ABC.

See also

- The BFRF (Search, List, Remove Backup Files) command on Page 2-49.
- The RF (Remove Backup Files) command on Page 2-320.

BFRF (Search, List, Remove Backup Files)

Keyboard Input Message

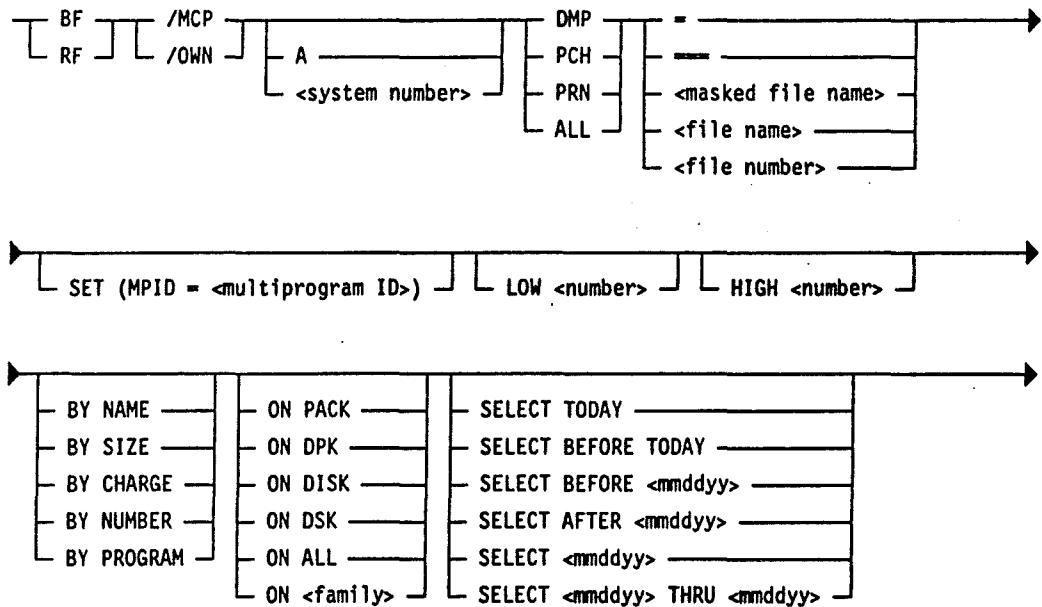
Access Level: 0

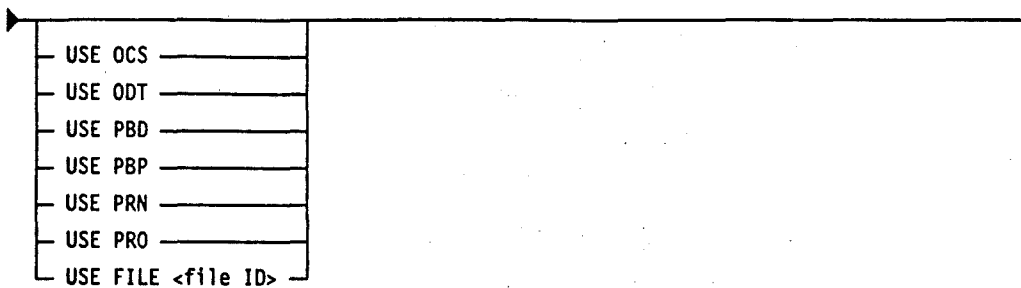
The BFRF command performs the same functions as the BF (List Backup Files) and RF (Remove Backup Files) commands do. However, instead of invoking the MCP, this command executes an intrinsic with additional file searching and listing features.

This program can list or list and remove backup files according to their types, numbers, and IDs. It can find the files according to name, date of creation, or backup file number. The files can be on disk or diskpack, families of diskpacks, an individual system, or the entire system.

Another BFRF feature organizes the lists of backup files. It sorts the lists alphabetically by name, sequentially by size, charge number, backup file number, or by the program that created the backup file.

Finally, another BFRF feature outputs those lists to a variety of media, including an ODT, a printer backup file on disk or diskpack, a line printer, or a disk file.





You can declare the keywords SET, LOW, HIGH, BY, ON, SELECT, and USE in any order following the file identifier.

Explanation

- BF** **BF** lists backup files.

- RF** **RF** lists and removes backup files.

- /MCP** **/MCP** initiates the intrinsic bound to the MCP.

- /OWN** **/OWN** initiates a freestanding version of BFRF, if one is on disk. If not, the MCP will display an error message that the program is not on disk.

- A** The element **A** initiates a search for files created on all systems.

- <system number>** The **<system number>** initiates a search for files created on the system with that number.

If the character **A** or a system number is not used, the search defaults to the system that initiates BFRF.

- ALL** These specify the type of backup files to be searched for. The following defines the meaning of each option.
DMP
PCH
PRN
- ALL** Lists all types of backup files
- DMP** Lists dump files
- PCH** Lists punch backup files
- PRN** Lists printer backup files

- <masked file name>** These indicate which files to list or remove.
=
===
- <masked file name>** If you use **<masked file name>**, BFRF lists or removes the files with matching names. A masked file name is a string of characters used to select one or more files. An equal sign (=) is used as a masking (wildcard) character. For example, **A=** selects all files beginning with **A**.
- <file name>**
- <file number>**

Equal signs can be used as masking characters to indicate that all files of a designated type should be removed. RF requires exactly three equal signs, while BF can use one or more equal signs.

If you use <file number>, BFRF lists or removes that file only. Note that when you list or remove files by backup number, you should use the MCP command BF/BFP or RF/RFP to avoid running a program just to list or remove a single file.

If you use <file name>, BFRF lists or removes backup files with that name. For PRN and PCH files, the name is a multifile identifier/file identifier or file identifier. For DMP files, the name is a program name that can be in the form multiprogram ID/program ID.

SET (MPID =
<multiprogram ID>)

This assigns different names to different executions of BFRF. For example, three BFRF operations in the mix simultaneously could be named MOE/BFRF, LARRY/BFRF, CURLY/BFRF. These names would appear on the Run log and make scheduling of other programs easier after a particular execution of BFRF.

LOW <number>
HIGH <number>

These are limits that can be used to search only those backup file numbers within a specified range. LOW defines the low end and HIGH defines the high end of that range. Both the LOW number and the HIGH number can be up to four digits long. A fifth digit is automatically taken from the system number.

LOW and HIGH are both optional. You can use one without the other. If you do not specify the LOW end of the range, it defaults to one. If you do not specify the HIGH end of the range, it defaults to 9999.

BY NAME
BY SIZE
BY CHARGE
BY NUMBER
BY PROGRAM

These options enable the user to specify the sort order of the backup file lists created by BFRF. The following describes the effect of each of these sort options.

BY NAME	Sort lists alphabetically by name
BY SIZE	Sort lists by file size (largest to smallest)
BY CHARGE	Sort lists by charge number
BY NUMBER	Sort lists by backup file number
BY PROGRAM	Sort lists by the name of the program that created the backup file

If you do not use these options, BFRF lists the files in the same order in which they appear in the directory.

ON PACK
ON DPK
ON DISK
ON DSK
ON ALL
ON <family>

These options narrow the search for backup files to a particular medium. The following describes the effect of each of these device options.

ON PACK Searches only diskpack
ON DPK

ON DISK Seaches only disk
ON DSK

ON ALL Searches all disk and unrestricted
 diskpack

ON
<family> Searches only a particular diskpack
 family

If you do not specify a device option, BFRF searches all disk and unrestricted diskpack.

SELECT TODAY
SELECT BEFORE TODAY
SELECT BEFORE
<mmddy>
SELECT AFTER
<mmddy>
SELECT <mmddy>
SELECT <mmddy>
THRU <mmddy>

These options enable the user to list or remove files based on the creation date of the file. The following describes the effect of each of these selection options.

SELECT Lists or removes files created on the
TODAY current date

SELECT Lists or removes files created before
BEFORE the current date
TODAY

SELECT Lists or removes files created before
BEFORE the specified date
<mmddy>

SELECT Lists or removes files created after
AFTER the specified date
<mmddy>

SELECT Lists or removes files created on the
<mmddy> specified date

SELECT Lists or removes files created between
<mmddy> the two specified dates
THRU
<mmddy>

USE OCS
USE ODT
USE PBD
USE PBP
USE PRN
USE PRO
USE FILE <file name>

These options enable the user to direct the output from BFRF to several media.

USE OCS This specifies that the output should
 be listed on the OCS.

USE ODT This specifies that the output should
 be listed on the ODT.

USE PBD This specifies that the output should
 be listed in a printer backup disk file.

- USE PBP This specifies that the output should be listed in a printer backup diskpack file.
- USE PRN This specifies that the output should be listed in a printer backup file using the system default device (PBD, PBP, PRN, or PBT).
- USE PRO This specifies that the output should be listed directly to the printer (no printer backup file is created).

If the USE option specifies FILE, the information is not formatted for display but is returned in a predefined format. The program accessing the file can then use that data as it is or format it as required.

The disk file created by the USE FILE option has the following attributes:

FAMILY DISK
 RECORDSIZE 44 BYTES
 BLOCKSIZE 20 RECORDS
 TITLE <file name>

The following program example illustrates the record format of the disk file created by BFRF.

```

01 BF-RECORD.
  03 BF-ID-FIELDS.
    05 BF-MULTI-FILE-ID          PICTURE X(06).
    05 BF-FILE-ID              PICTURE X(06).
    05 BF-CREATOR-MULTI-PROG-NAME PICTURE X(06).
    05 BF-CREATOR-PROG-NAME     PICTURE X(06).
    05 BF-FILE-TYPE            PICTURE 9(01) COMP.
    05 BF-BK-NUMBER            PICTURE 9(05) COMP.
  03 BF-STATUS-FIELDS-PRN
    05 BF-CREATION-DATE-YYDDD   PICTURE 9(05).
    05 BF-CHARGE-NUMBER         PICTURE 9(06).
    05 BF-EOF-PB-PC            PICTURE 9(06).
    05 BF-QT-LINE-PB           PICTURE 9(06).
    05 BF-QT-FLAG-PB           PICTURE 9(03).
  03 BF-STATUS-DMP-BRK
    REDEFINES BF-STATUS-FIELDS-PRN COMP.
    05 FILLER                   PICTURE 9(05).
    CREATION DATE
    05 FILLER                   PICTURE 9(06).
    CHARGE NUMBER
    05 BF-EOF-DMP-BRK           PICTURE 9(05).
    05 BF-DC-DATE-DMP-BRK      PICTURE 9(06).
  
```

05 BF-DC-TIME-DMP-BRK	PICTURE 9(04).
03 BF-ATTRIBUTES	COMP.
05 BF-MEDIA	PICTURE 9(01).
05 BF-NO-CHANNEL-12	PICTURE 9(01).
05 BF-UNLABELED	PICTURE 9(01).
05 BF-FORM	PICTURE 9(01).
05 BF-ATTRIBUTE-2	PICTURE 9(01).
05 FILLER	PICTURE 9(03).

Note that it is the user's responsibility to make the file name unique.

If the BFRF search does not find any files and you specify the USE FILE option, the system still creates a disk file even though it contains no records.

If <multiprogram ID> is not specified, the system places the program identifier in the variable BF-CREATOR-MULTI-PROG-NAME (for sorting purposes) and fills the variable BF-CREATOR-PROG-NAME with blank spaces.

The variable BF-FILE-TYPE defines the type of backup file as follows:

0	Printer
1	Punch
2	Dump

Some fields have no meaning when BF-BACKUP-TYPE is other than 1 (for example, BF-FORM and BF-NO-CHANNEL-12).

If the file has been stopped by the QT (Quit Program Operation) command, the BF-QT-LINE-PB and BF-QT-FLAG-PB fields will contain, respectively, the last record number printed and the three-hexadecimal QT flag.

The variable BF-ATTRIBUTE-2 is the medium on which the program was executing for dump files. The value of BF-ATTRIBUTE-2 is 0 for disk and 1 for diskpack. BF-MEDIA is the location of the backup file itself. The value of BF-MEDIA is 0 for disk and 1 for diskpack.

Executing the BFRF Intrinsic

You can execute the BFRF intrinsic with the EXECUTE (Execute Library Program) command.

- EX - BFRF

/SPO

Explanation

EX BFRF

This executes the BFRF intrinsic.

When the MCP schedules BFRF from a BF/RF /MCP or /OWN command, the system provides a passfile containing the syntax entered for the program. When BFRF is executed directly, the passfile does not exist and the program looks for a parameter file on disk named BFPARM to define the specifications to be used. BFPARM consists of 100-character records (blocked 1 on disk) and can be file equated.

If the parameter file does not exist, the program asks for input in the form of an AX (Respond To An ACCEPT) command. Enter one of the following to terminate this form of input:

<mix no> AX
<mix no> AX END

EX BFRF /SPO

This also executes the BFRF intrinsic, but provides no check for a parameter file and requests input in the form of an AX (Respond To An ACCEPT) command. Enter either "<mix no> AX" or "<mix no> AX END" to terminate this form of input.

Examples

RF /OWN PRN —

This initiates the freestanding version of BFRF to search on disk and unrestricted diskpacs and remove any printer backup files created by the system through which this command is entered. By default, the output from BFRF is directed to the ODT.

BF /MCP 1 ALL = ON DISK USE OCS

This initiates the bound intrinsic BFRF to search on disk for all types of backup files created by system number 1. The output is directed to the OCS.

BFRF

BF /MCP 2 ALL — BY NAME USE ODT

This initiates the MCP's bound intrinsic BFRF. It seeks backup files of all types created by system 2, sorts the list alphabetically by file name, and outputs that list to the ODT.

RF /OWN A DMP — BY CHARGE

This initiates a freestanding program BFRF. It seeks and removes dump backup files created by all the systems in a multi-system shared configuration. Then it sorts the list of the removed files sequentially by charge number. Because no particular medium is specified, it outputs that list to the ODT.

**BF /OWN A PRN FILE— LOW 1 HIGH 700
ON PACK SELECT TODAY BY NAME USE OCS**

This initiates a freestanding version of BFRF. It lists printer backup files on diskpack created by all the systems in a multi-system shared configuration. Of those files, it lists only those created today with backup numbers ranging between 1 and 700. Finally, it sorts the list alphabetically by file name and outputs the list to the OCS.

See also

- The BF, BFP (List Backup Files) command on Page 2-47.
- The EXECUTE (Execute Library Command) command on Page 2-162.
- The QT (Quit Program Operation) command on Page 2-308.
- The RF, RFP (Remove Backup Files) command on Page 2-320.

BINARY (Read Binary Card Deck)

MCP Control Instruction

Access Level: 0

The BINARY command reads a binary card deck.

The deck must be terminated by a binary end card to allow normal End-Of-File (EOF) operation.

Since the format of this card is difficult to keypunch, it is recommended that a binary punch file be opened and closed several times to produce a supply of these cards.

A binary card deck must come directly from the reader; it is not valid under LDCNTL, the system utility that creates pseudo decks from physical card decks.

- BINARY - <file name> _____

Explanation

<file name> This specifies the name of the card deck to be read.

Note: The <file name> of CTLDCK is reserved and should not be used.

Example

```
EXECUTE A; BINARY ACARD
```

This executes the library program called A, and then reads a binary card deck called ACARD.

See also

- The BCL (Mark Beginning of Card Data File) control instruction on Page 2-43.
- The DATA/DATAB (Mark Beginning of Card Data File) control instruction on Page 2-105.
- The EBCDIC (Mark Beginning of Card Data File) control instruction on Page 2-153.
- The END/ENDCTL (Mark End of Card File or Control Instructions) control instruction on Page 2-156.

BO (Blackout Access Code)

Keyboard Input Message

Access Level: 0

To keep access codes secret, you can use the BO command to prevent unauthorized persons from viewing a particular access code as it is entered during the log-in procedure.

- BO _____

Details

You can invoke the blackout feature by entering the BO command before logging in. This command "blacks out" a line on the input device and leaves the carriage positioned at the beginning of the line.

See also

The LI (Log In) command on Page 2-221.

BT (Begin TRAK)

Keyboard Input Message

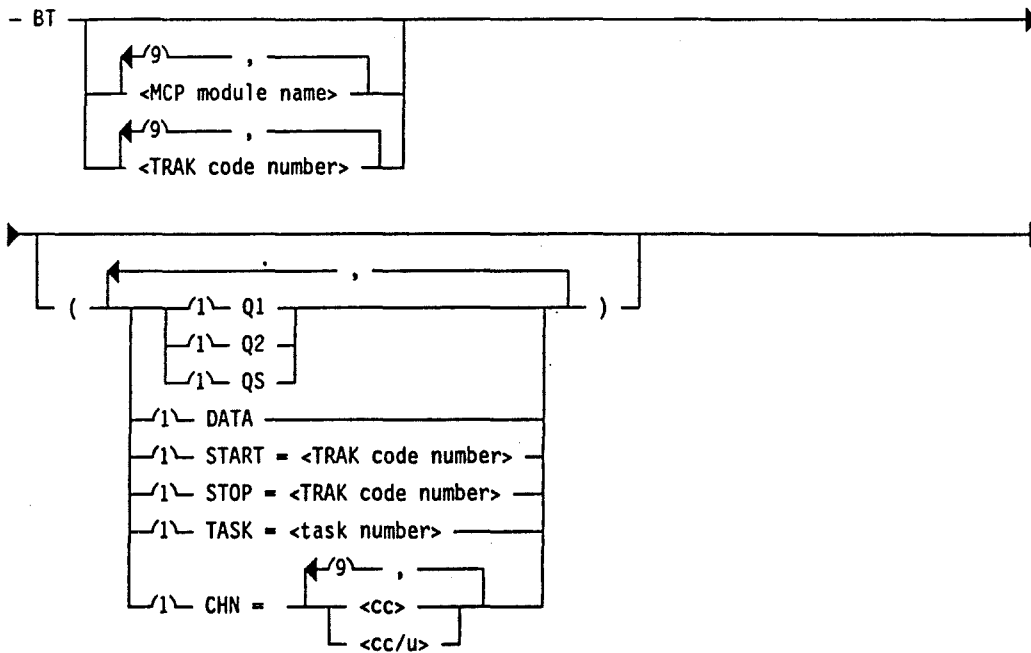
Access Level: 8

The BT command selects a set of TRAK calls and causes the TRAK facility to begin tracking those calls. This command is primarily used for debugging the operating system.

TRAK is a real-time debugging tool that records various MCP data structures or events. TRAK calls (sometimes called TRAK *invocations*) are embedded in the operating system at various points. Each TRAK call causes certain data to be stored in a portion of memory called the TRAK buffer. The TRAK buffer is included in the memory dump file and can be analyzed by DMPANL, the dump analysis utility.

When you select a set of TRAK calls using the BT command, that set of BT specifications remains in effect across halt/loads. The BT command can select TRAK calls based on the following criteria.

- TRAK code number (unique to each TRAK call)
- The name of an operating system module
- The kind of data that a TRAK call inserts in the TRAK buffer
- The task for which the operating system was executing when the TRAK call was encountered



Explanation

<TRAK code number> Selects TRAK calls based on the 4-digit code used to identify a TRAK call and its subsequent entry in the TRAK buffer. This code (and any text interpretation of it) are printed out by DMPANL in its analysis of the TRAK buffer. You can select a maximum of 10 different TRAK codes.

<MCP module name> Selects TRAK calls contained within a set of one or more operating system module. You can select a maximum of ten different modules. The following module names are valid:

Module Name	Description
CWT	Complex Wait module
DCOM	Data Communications module
DCP	Data Communications Processor module
DCU	Data Communications Processor module
DMS	Data Management System module
DPM	Diskpack Maintenance module
IOMOD	Input/Output module
ISC	Inter-System Connect module

Module Name	Description
JMR	Job Manager module
MSGMOD	Message module
NCP	VCS module
NWK	VCS module
PRT	Port module
RJE	Remote Job Entry module
SCA	Subports module
STATUS	STATUS module
STQ	Storage Queue (STOQ) module
VCS	VCS module

Q1 Q2 QS	These options select TRAK calls that insert copies of the I/O queue elements into the TRAK buffer. Q1 selects queue elements of I/O operations to be initiated (fired). Q2 selects queue elements of completed I/O operations. QS includes both types.
CHN = <cc> CHN = <cc/u>	Valid only when IOMOD is one of the selected modules, or with the Q1, Q2 or QS options. This option selects TRAK calls that insert I/O queue elements into the TRAK buffer and further selects only those occurrences of a TRAK call where the queue element is related to the specified channel number or channel number and unit number. Up to ten different channel number and channel number and unit number specifications can be made in a single BT command.
DATA	Selects TRAK calls that insert non-standard TRAK data into the TRAK buffer. (Non-standard includes any data other than the I/O queue elements or the stack frames and parameters).
START = <TRAK code number>	Starts tracking selected TRAK calls only after the entered TRAK code number is encountered.
STOP = <TRAK code number>	Suspends tracking selected TRAK calls after the entered TRAK code number is encountered.
TASK = <task number>	Selects only those occurrences of TRAK calls that relate to the entered task number.

Details

The TRAK option can be enabled either through the USE TRAK record of the system configuration file or through the SO (Set System Options) command.

The tracking initiated by the BT command is terminated by using the ET (End TRAK) command.

Examples

BT IOMOD,DCP (TASK = 042)

This begins tracking of the MCP modules IOMOD, and DCP; includes only TRAK calls relating to task 42.

BT 9310,1200,1234 (START. 8500)

This begins tracking of TRAK calls associated with TRAK codes 9300, 1200, and 1234 after TRAK code 8500 is encountered.

BT IOMOD (CHN = 26, 25/0, 25/1)

This begins tracking of all I/O operations initiated and completed on channel 26, and unit numbers 0 and 1 on channel 25.

See also

- The ALTER (Change System Parameters) command on Page 2-21.
- The ET (End TRAK) command on Page 2-161.
- The PM (Print Memory Dump) command on Page 2-284.
- The SO (Set System Options) command on Page 2-352.
- The USE TRAK and LIMIT TRAKBUFFER records in the *V Series Systems Operations Guide Volume 1: Installation*.
- The DMPANL utility in *V Series Systems Operations Guide Volume 3: System Utilities*.

BUFFER (Set MCS Result Queue Size)

MCP Control Instruction

Access Level: 0

The BUFFER command sets the number of input messages a Message Control System (MCS) result queue can store.

- BUFFER - <DCP file name> <integer> _____

Explanation

<DCP file name> This must match either the name defined in the "MCSNAME=" statement in the B 974 NDL or in the "MCS= " statement in the B 874 NDL. The name can be up to six characters long.

= This is optional and does not influence the command.

<integer> This indicates the number of messages the buffer holds. The number must be in the range of 1 through 99. The default is 5.

Use <integer> carefully. A low value can cause unnecessary delays in receiving a message from the DCP. Large values, on the other hand, use large amounts of memory.

Note: Note that the memory size (in digits) of an MCS result queue in a DCP file can be computed using the following equation:

$$\langle \text{maximum record size in digits} \rangle * (\langle \text{number of messages in the result queue} \rangle + 1) + 200.$$

Details

Use this command when

- You initiate an MCS program that uses the Data Communications Processor (DCP) file as specified by <DCP file name>.
- The DCP file as specified by <DCP file name> is closed and the DCP option is set.

Once the buffer size is set, that size remains in effect until it is changed by another BUFFER command or the DCP option is reset.

BUFFER

Example

BUFFER OURDCP 15

This allows the result queue in the DCP file called OURDCP to store up to 15 input messages.

See also

The V Series BNA Installation and Operations Guide for more information.

CA (List Port File Candidates)

Keyboard Input Message

Access Level: 0

The CA (CANDIDATE) command provides information about the current port file candidates. Port file candidates are ports and subports that are in the process of opening; they might be awaiting matching cooperators or be in some phase of the matching process.

```

┌ CA _____┐
└ CANDIDATE ┘ └ <port name> ┘
  
```

Explanation

CA
CANDIDATE

These are synonymous and, if used without the <port name> option, return a list of all candidates with summary information about each.

<port name>

This displays a list of only those candidates with this port name, and displays summary information about each.

Example

CA LIOPORT

This displays information about LIOPORT. A sample response is as follows:

```

CANDIDATES: LIOPORT SUBPORT= 0001 FROM (???) DBG450/DISK /0004
STATUS = OFFER OUTSTANDING      SECURITY = PRIVATE YOURUC = boU
MYHOST = HOSTQ                   YOURHOST = HOSTE
MYNAME = LIOINIT                 YOURNAME = LIOCOOP
  
```

See also

The V Series BNA Installation and Operations Guide.

CC (Enter Control Instructions)

Keyboard Input Message

Access Level: 0

The CC command supplies control instructions and text to the MCP through the ODT. The information following the letters CC is recognized in the same way as information following an invalid character in column 1 of a control record.

`[CC] <control information>`

Explanation

<control information> This is any valid MCP control instruction except DATA/DATAB (Mark Beginning of Card Data File), LABEL1 (Change Tape Label Format), or SPO (Simulate SPO through Card Reader). When entering control information, note the following:

- All control information including program parameter instructions must be entered on the same line as the CC command.
- The control information must be separated from the initial letters CC by at least one blank space.
- A hyphen (-) cannot be used to continue an instruction to the next line.

Details

The CC command is not required prior to entering any control information. It is provided to maintain compatibility with previous versions of the MCP. For example, if a system running MCP/VS 2.0 is communicating with a system running MCP/VS 1.0, system commands must be prefaced with the characters CC if they are entered through BNA from an ODT and if they start a job.

While the CC command allows you to enter MCP control instructions, the SPO (Simulate SPO through Card Reader) command provides capability for keyboard input messages to go to the MCP through a card reader as if they had been entered at the ODT.

Examples

CC REMOVE A=

This allows you to enter a REMOVE (Delete Files) command to delete the files with the file ID of A= from the disk directory.

CC COMPILE BOB WITH COBOL9 LIBRARY; PRIORITY 6

This allows you to enter a COMPILE (Compile Program) command to compile a COBOL program called BOB and then save it as a permanent file. This job is assigned a memory, processing, I/O, and schedule priority of 6.

CC EXECUTE TESTY; MEM 400000; MEMDUMP

This allows you to enter an EXECUTE (Execute Library Program) command to execute a program called TESTY, specifying that the program needs 400,000 digits of memory to execute. If the program terminates abnormally, a dump is requested.

CC CHANGE A=B=C= TO X=Y=Z=

This allows you to enter a CHANGE (Change File ID) command to change the name of files on disk with a file ID of A=B=C= to a file ID of X=Y=Z=.

See also

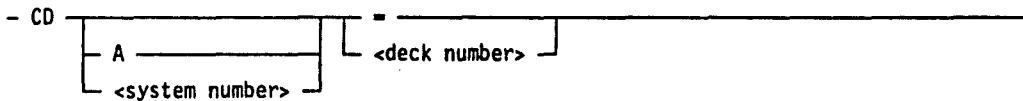
The SPO (Simulate SPO through Card Reader) command on Page 2-357.

CD, CDP (List Inactive Pseudo Decks)

Keyboard Input Message

Access Level: 1

The CD command lists on the ODT the inactive pseudo card decks on disk; CDP lists the inactive decks on diskpack. The MCP responds with a list of the file numbers and the contents of the first card image in the control decks.



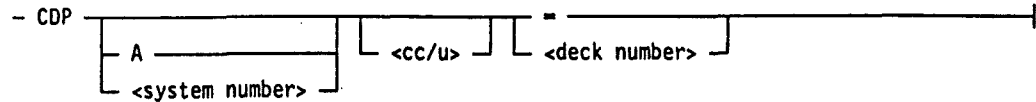
Explanation

- | | |
|-----------------|--|
| CD | This lists the inactive pseudo card decks on disk. |
| A | This indicates a system specification of all systems. |
| <system number> | This can be specified in a multi-system shared configuration to obtain a list of the inactive pseudo decks created on the indicated system.

If you omit either A or <system number>, the response includes only those decks created on the system through which the request is made. |
| = | This lists all decks of the indicated system. |
| <deck number> | This causes the MCP response to include only the specified deck.

Pseudo decks are designated as #snnnn. The <i>s</i> represents the system number and is not required unless the deck was created by a system other than the one from which you are entering the command. The <i>n</i> is a four-digit number with leading zeros assumed.

Do not include the pound sign (#) in the command syntax. |



Explanation

CDP	This lists the inactive pseudo card decks on diskpack.
<cc/u>	This specifies a channel number and unit number. The <cc/u> can be used only with CDP and causes files from a diskpack with the specified channel number and unit number to be displayed.

Examples

CD A =

This displays a list of all the inactive pseudo card decks on all the systems.

CD 1 3

This displays the inactive pseudo card deck number 3 on system number 1.

CDP 6/3 =

This displays a list of all the inactive pseudo card decks from the diskpack on channel number 6, unit number 3.

See also

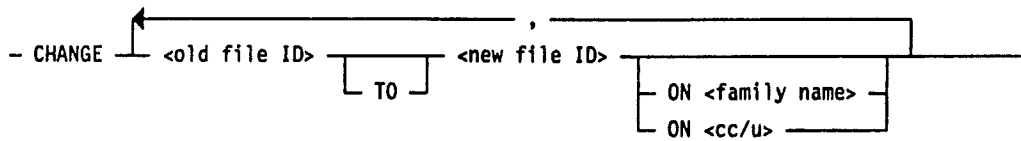
- The DA, DAP (Deactivate Pseudo Reader) command on Page 2-103.
- The PD (Print Directory) command on Page 2-278.
- The RD, RDP (Remove Pseudo Decks) command on Page 2-313.
- The RN (Activate Pseudo Reader) command on Page 2-324.

CHANGE (Change File ID)

MCP Control Instruction

Access Level: 5

The CHANGE command changes a disk or diskpack file ID. If the SECURITY option (DFSC) is set and the file has a SECURITYTYPE other than PUBLIC, the change can be made only under the access code of the creator of the file or by a user with LIBMAINT privileges.



Explanation

CHANGE
CH

These are synonymous.

<old file ID>

This indicates the name of the file that you want to change. The file name can be specified as a masked file name using the equal sign (=) as a masking character.

If the <old file ID> is not masked and more than one file exists with the specified name, a DUP FILE message will be displayed.

If you specify an <old file ID> using a masked file name and multiple files exist that match it, only the first matching file encountered will be changed. All subsequent matching files will get DUP FILE errors.

<new file ID>

This indicates the new name you want to apply to the file. The new file name can be specified as a masked file name using the equal sign (=) as a masking character.

An attempt will be made to change all matching files. A file will not be changed if an existing file has the same name. A DUP FILE error will be produced in this case.

If any specified files are IN USE or declared to have NO LIBMAINT, those files will not be changed.

ON <family name>

If the ON clause is omitted or the <family name> is DISK, the files to be changed are assumed to reside on disk (100-byte media). If the ON clause specifies a

<family name> of PACK, all system resource diskpacks will be searched for the requested files.

If the <family name> is a valid diskpack family, all diskpacks of the family with the specified name will be searched. If no file BASE areas are found which match the <old file ID>, a NO FILE error message will result.

ON <cc/u>

The <cc/u> (channel number and unit number identifier) must specify a diskpack. All files specified by <old file ID> will be located. If no files are found on the specified <cc/u> (BASE areas only) matching the <old file ID>, a NO FILE error will result.

A duplicate file check for the new file ID will be performed across the entire diskpack family, not just the specified <cc/u>.

Details

If any areas of the file you want to change are on a diskpack that is missing or offline, the following occurs:

- Only the areas that are present are changed.
- The file is flagged as CHANGE IN PROGRESS.
- All further access to the file is restricted.

When the missing diskpacks are made available to the system, you can issue another CHANGE command. The diskpack containing the first area of the file must still be present for all subsequent change attempts.

After all flagged areas have been changed, the file is no longer flagged as CHANGE IN PROGRESS, and access to it is no longer restricted. If for some reason the entire file is not available to complete the change, a REMOVE (Delete Files) command can be issued to remove any areas that have not yet been changed. Areas that have been changed will not be located by a REMOVE command even if present on the system. Eventually, a REMOVE command using the serial number option will have to be used to restore diskpack integrity.

Pseudo-Device File-Identifiers

If <new file ID> specifies a pseudo device, the file ID will be changed to ascending-numbered backup files of the appropriate type.

Using the following identifiers, the file ID of a file, or of a series of files, can be changed to a pseudo-device file identifier recognizable by the MCP:

CHANGE

File Identifier	Function
@	Creates a printer backup file.
*	Creates a punch backup file.
#	Creates a pseudo card deck file.
\$	Creates a memory dump file.

Each of the preceding files are assumed to be in the correct format to complete the change because a check is not made by the MCP.

Trailing zeros are not permitted on the pseudo-device file identifiers.

If a file is changed to a pseudo-reader device and APCR is set, an attempt will be made to run the resulting pseudo decks. If you enter CH <old file ID> # or CH <old file ID> #= and APCR is set, all pseudo card decks that have not been previously activated through an RN or RNP (Activate Pseudo Reader) command will be activated.

Note: If a channel number and unit number is specified, the channel number and unit number must refer to the base pack of a Version 2 pack family. An attempt to directly reference a continuation pack of a Version 2 pack family results in an error message. For information on pack family versions, refer to "Pack Subsystems" in the V Series Systems Operations Guide Volume 3.

Examples

CH FILEB TO *

This changes the file name FILEB to the next available punch backup file ID.

CH FILEC TO @00256

This changes the file name FILEC to printer backup file 00256.

CH FILE= TO @

This changes all files on disk beginning with FILE to the next available pseudo-device file identifier; therefore, the operator is notified of the CHANGE regardless of the setting of the MCP LIB option.

CHANGE PBD= TO @

This causes the file identifiers of all files beginning with PBD to be changed to printer backup file identifiers.

CH A TO B, C D, E TO F, G= H=

This changes file name as follows: A to B, C to D, E to F, file names beginning with G to file names beginning with H.

CHANGE ABDFIL TO XYZFIL ON 15/3

This changes the file name of a file on a diskpack with channel number 15, unit number 3.

See also

- The FILE (Equate File Name With File Identifier Or Medium) command on Page 2-167.
- The REMOVE (Delete Files) command on Page 2-315.

CHARGE (Insert Charge Number)

MCP Control Instruction

Access Level: 1

The CHARGE instruction inserts a user charge number into the MCP Run log (RLOG) records.

The CHARGE command can be used to assign a user-designated charge number to the execution of any program. If the CHRGE option is set, all programs being scheduled for operation must have a charge number to be accepted by the MCP.

┌ CHARGE ────┐ <integer> ───────────────────────────────────┐
└ CG ────┘

Explanation

CG
CHARGE
<integer>

These are synonymous.

This is the charge number to be associated with the program. The charge number must be a number not longer than six digits. If fewer than six digits are used, leading zeros are assumed. This number is carried in the RLOG file for subsequent time analysis or billing use.

Details

The ZIP facility allows user programs to send system commands, and the commands sent with the ZIP facility are acted on exactly as if they had been entered at the ODT. The ZIP facility is invoked by constructs in the various programming languages. When a program uses a ZIP statement to schedule another program for operation and fails to supply a charge number for the ZIPped program, the charge number of the program doing the ZIP will be used for the ZIPped program.

The execute phase of a compile-and-go operation is assigned the charge number associated with the compile. A compile-and-go operation occurs if the LIBRARY or SYNTAX option is not used with the COMPILE (Compile Program) command. An object code file will be generated and executed, and then removed from the system unless SAVE is specified.

CHARGE associates a program with a charge number but does not affect the security attributes of the user or the program.

Examples

EXECUTE A; CHARGE 670345

This executes a program called A and inserts a user charge number of 670345 for it in the RLOG.

PERFORM LIST A; CG 5

This invokes the DMPALL utility to print the contents of a file called A and inserts a user charge number of 000005 for it in the RLOG.

See also

- The COMPILE (Compile Program) command on Page 2-80.
- The USER (Assign Access Code) command on Page 2-401.
- The USE CHRГ record in the *V Series Systems Operations Guide Volume 1: Installation*.
- The various language manuals for information about the ZIP facility.

CHECK (Verify Library Tape)

MCP Control Instruction

Access Level: 5

The CHECK command verifies the validity of a library tape.

It is recommended that you use the COPY command instead of the CHECK command, especially the COMPARE, CHECK, and SUMMARY options. The CHECK command will become unavailable at a future release.

The CHECK command starts a program called LOADMP, if the files are to be checked on disk, or a program called PACKUP, if files are to be checked on diskpack. The user must supply the programs LOADMP and PACKUP; Unisys does not supply them. If the appropriate program is not present on disk, the CHECK command is ignored.

See Appendix B for CHECK syntax if you have user-coded LOADMP and/or PACKUP utilities.

See also

- The COPY (Invoke File Transfer Utility) command on Page 2-86.
- Appendix B.

CK (Test Peripheral Device)

Keyboard Input Message

Access Level: 7

The CK command tests the status of a peripheral unit by using the Test I/O operation. Any peripheral device except a reader-sorter (SOR, S4A, or S4B) or a non-status device (NST) can be checked with the CK command.

- CK 

Explanation

<cc/u>

This is the channel number and unit number of the device to be checked.

Details

The MCP issues a Test I/O to the device. The MCP displays four words (that is, four groups of four digits) of result descriptor (R/D) information. An R/D is a coded description of the results of an I/O operation.

Example

CK 6/3, 6/4

This checks the status of the peripheral devices on channel 6, unit numbers 3 and 4.

See also

The *V Series Programming Reference Manual* for information about reader-sorter interfaces.

CN (Display Tape Number)

Keyboard Input Message

Access Level: 7

The CN command displays the 5-digit physical tape number contained in a magnetic tape label.

- CN - <cc/u>

Explanation

<cc/u>

This is the channel number and unit number of the tape unit. The <cc/u> must reference a magnetic tape unit.

Details

You can use the AC (Purge and Place Tape Number in Label) or SN (Purge and Place Tape Number in Label) command to insert a physical tape number into a magnetic tape label.

Example

CN 2/1

This displays the physical tape number in the label of the magnetic tape on channel number 2, unit number 1.

See also

- The AC (Purge and Place Tape Number in Label) command 2-2.
- The SN (Purge and Place Tape Number in Label) command 2-350.

COMPARE (Compare Files on Different Media)

MCP Control Instruction

Access Level: 5

COMPARE invokes the file transfer utility SYSTEM/COPY to compare the contents of files on different media. See the COPY (Invoke File Transfer Utility) command for details about syntax and how to use this command.

See also

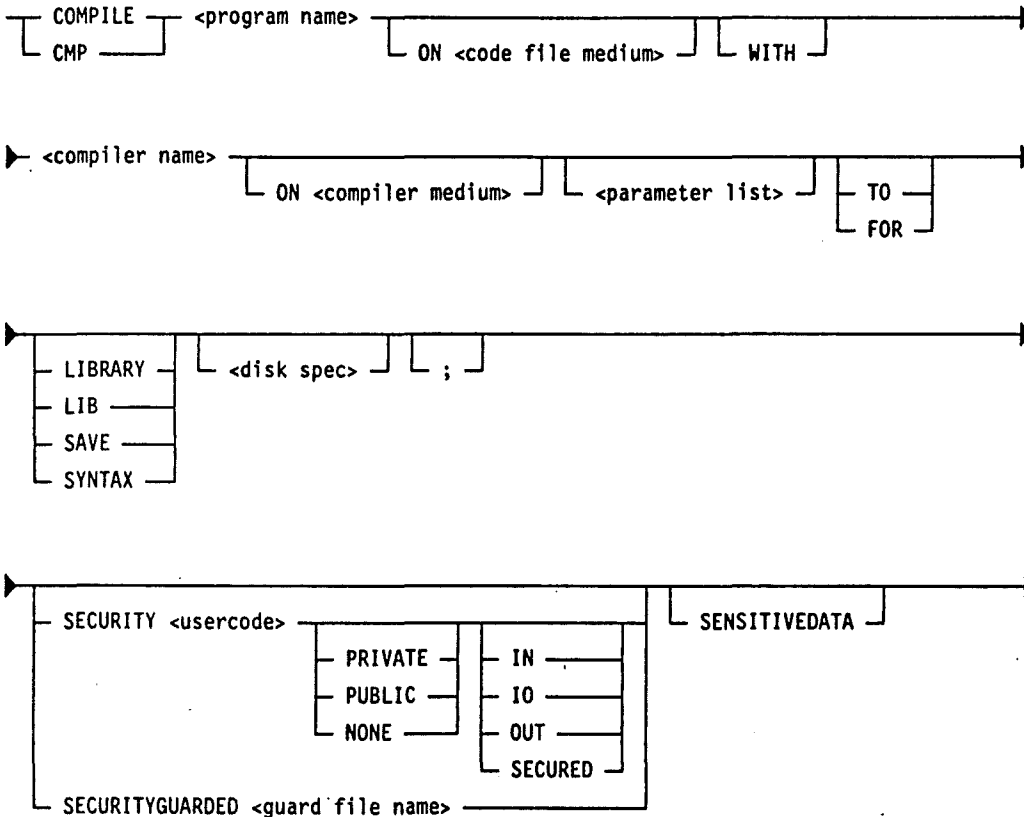
The COPY (Invoke File Transfer Utility) command on Page 2-86.

COMPILE (Compile Program)

MCP Control Instruction

Access Level: 1

COMPILE executes an assembler or compiler to generate an object program from source language images.



Explanation

CMP
COMPILE

These are synonymous.

<program name>

This is the name of the code file to be created.

<code file medium>

This specifies where the code file will reside after it is created. The available options are as follows; if no option is specified, the default is to create the code file on 100-byte disk.

DISK

This specifies creation of the code file on the default disk subsystem.

PACK	This specifies creation of the code file on a system resource diskpack.
A diskpack family name	This specifies creation of the code file on the given diskpack family name.

Note: The <code file medium> can specify a diskpack only if the compiler used is capable of writing code files to diskpack. Some older compilers (such as COBOLV and BPL) lack this feature.

WITH This is optional and does not influence the command.

<compiler name> This is the name of the program that creates the code file.

<compiler medium> This informs the MCP where to look for the compiler code file.

The CONTROL CODEPACK record that can be included at cold start in the system configuration file affects the sequence of attempts to find the compiler for execution. If ON <compiler media> is not included, the MCP will first look for the compiler on 100-byte disk. If not found, the MCP will look for the compiler on the diskpack specified in the CONTROL CODEPACK record, if specified. Otherwise, all nonrestricted diskpacks will be searched.

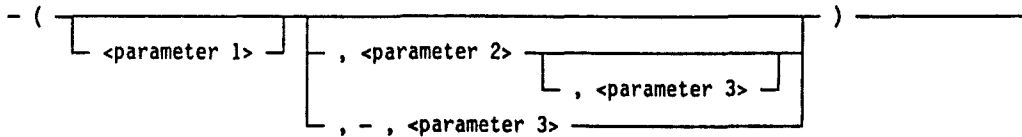
The allowable values for <compiler medium> are as follows:

DISK	This causes the MCP to look for the compiler code file only on disk.
PACK	This causes the MCP to look for the compiler code file among all available non-restricted diskpacks regardless of the presence of a CONTROL CODEPACK declaration at cold start.
A diskpack family name	This causes only the specified diskpack family to be searched.

<parameter list> This contains up to three parameters that can be inserted into a program when the program is started.

The <parameter list> option appears in the syntax of the COMPILE (Compile Program), DEBUG (Debug Session), and EXECUTE (Execute Library Program) commands. These parameters allow you to enter Boolean, string, and integer values, which can be used by the program when the program is being compiled or executed.

The syntax for <parameter list> is shown in the following figure.



The rules for <parameter list> resemble those for parameters in the Work Flow Language (WFL) RUN command.

Integer parameters

- Contain unsigned numeric data up to eight digits in length (8 UN)
- Have values ranging from 0 through F
- Are right-justified and zero-filled in the high-order digits if less than the maximum length
- Will cause a syntax error if greater than the maximum length

String parameters

- Contain alphanumeric data up to six characters in length (6 UA)
- Must be within quotation marks.

To pass a quoted string as a parameter, a double quote must appear in the parameter string (for example, COMPILE MYPROG COBOL9 (, 'TRY' 'XX')).

- Can be composed of any EBCDIC characters
- Are left-justified and blank-filled in the low-order bytes if less than the maximum length
- Will be truncated in the low-order bytes if greater than the maximum length

<parameter 1>

This can be an integer up to eight digits in length. It sets the switches at program address 0. For example, CMP MYPROG COBOL9 (10100000).

<parameter 2>
<parameter 3>

This can be 8-digit integers or 6-character strings. The value of <parameter 2> is inserted into the code file at base-relative address 8; The value of <parameter 3> at base-relative address 20. For example, COMPILE MYPROG WITH COBOL9 (,12345678) or COMPILE MYPROG WITH COBOL9 (101, 'TOTEST').

When either <parameter 2> or <parameter 3> are specified, you must have either a value or at least a comma in the preceding parameters. For example, `COMPILE MYPROG WITH COBOL9 (,,'TAPE')`.

A null entry is allowed. For example, `COMPILE MYPROG WITH COBOL9 (,0,'ABC''''')` does not change the switches and inserts 00000000 at address 8 and the string "ABC" at address 20.

Note: Slash parameters were available on previous releases of the MCP. In later releases, slash parameters will be deimplemented. The <parameter list> option replaces them.

**TO
FOR** These are optional and do not influence the command.

**LIBRARY
LIB** This specifies that the new code file is to be made a permanent file on the specified disk or diskpack after an error-free compile. The code file is not active until explicitly executed.

SAVE This specifies that at the completion of an error-free compile the new code file is to be made a permanent file on the specified disk or diskpack. In addition, it is automatically executed.

SYNTAX This specifies that this is a compile for syntax-checking only. The code file is neither saved nor executed.

If you do not specify LIB, SAVE, or SYNTAX, the program will be compiled and executed. (This is referred to as compile-and-go.) If error free, the object file will be removed at EOJ.

<disk spec> This is a 1- or 2-digit integer specifying the logical ID number on which the new code file is to reside (values 01-39) or the disk subsystem on which it is to reside (values 90-98). Note that <disk spec> can be included only if the code file is to be created on a 100-byte medium.

SECURITY This indicates that security attributes will be applied to this file.

PRIVATE This means that the file can be accessed only by the creator of the file in the manner specified by the SECURITYUSE attribute. PRIVATE is the default SECURITYTYPE attribute.

COMPILE

NONE	This specifies that no security attributes are to be applied to this file.
PUBLIC	This means that the file can be accessed by all users in the manner specified by the SECURITYUSE attribute.
IO	This means that the SECURITYUSE attribute will allow READ/WRITE access to the file. IO is the default value assigned to SECURITYUSE.
IN	This specifies that only READ access is allowed to the file.
OUT	This specifies that only WRITE access is allowed to the file.
SECURED	This means that the file is a program and can only be executed. Data cannot be read from or written to this file.
SECURITY GUARDED <guard file name>	This specifies that another file identified by <guard file name> must be accessed before a user can be given permission to use the file. The SECURITYUSE attribute for a guarded file is stored in the GUARD file.
SENSITIVEDATA	This specifies that all records of the file are to be overwritten with random data before the file is returned to the available disk pool. The SENSITIVEDATA attribute will be ignored if the file is declared as PUBLIC.

Getting Programs in the Correct Format

You can use the COMPILE (Compile Program) command to create a program code file on diskpack in the correct 180-byte code file format.

The PCOPY utility, which is described in the *V Series Systems Operations Guide Volume 3: System Utilities*, can be used to copy a 100-byte disk format program code file to diskpack in the correct 180-byte code file format.

Examples

```
COMPILE OBJPGM ON SYSPAK WITH COBOL9 ON SYSPAK TO LIBRARY
```

This compiles OBJPGM, which resides on a diskpack called SYSPAK, with the COBOL9 compiler, which is also on SYSPAK. If the compile is error-free, the new code file, OBJPGM, will become a permanent file on SYSPAK.

CMP TEST RPG SYNTAX

This compiles TEST with the RPG compiler only to check syntax. The code file is neither saved nor executed.

CMP OBJ001 COBOL9 LIB ; SECURITY PRIVATE SENSITIVEDATA

This compiles OBJ001 with the COBOL9 compiler. If it is an error-free compile, the new code file, OBJ001, will become a permanent file. This file can be accessed only by the creator of the file, and when the file is removed, all areas of the file will be overwritten with random data before they are returned to the available disk pool.

See also

- The EXECUTE (Execute Library Program) command on Page 2-162.
- The CONTROL CODEPACK records in the *V Series Systems Operations Guide Volume 1: Installation*.
- The PCOPY utility in the *V Series Systems Operations Guide Volume 3: System Utilities*.

COPY (Invoke File Transfer Utility)

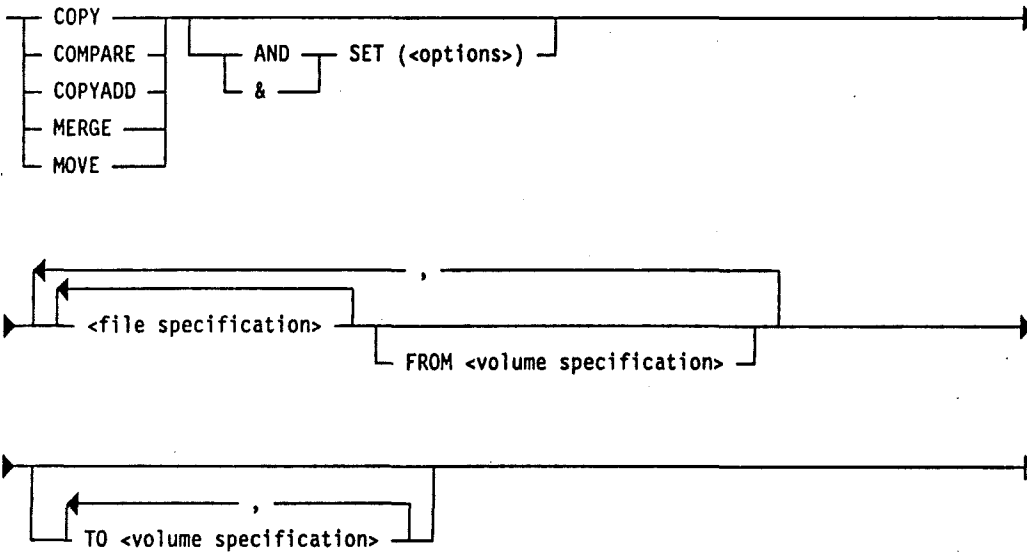
- COMPARE (Compare Files on Different Media)
- COPYADD (Transfer Files)
- MERGE (Transfer Files from Multiple Sources, Resolving Duplicates)
- MOVE (Move Files to New Media)

MCP Control Instructions

Access Level: 5

These five commands invoke the SYSTEM/COPY utility, which transfers files and performs library maintenance functions between disk, diskpack, and tape.

See the *V Series Systems Operations Guide Volume 3: System Utilities* for more information about the SYSTEM/COPY utility.



Explanation

COPY

This copies the indicated files. If the files already exist at the destination, they will be replaced with the newly copied files.

Note that you cannot selectively copy files from multiple sources to multiple destinations.

COPYADD

This copies only files that do not already exist at the destination. This is not a valid function if tapes are being used as destination media.

If AS is used to rename files as they are copied, the new names are used in the check for duplicates.

COMPARE

This is both a keyword and an option. As a keyword it is used primarily when a large number of files are being copied to tape.

Both the option and the keyword compare newly copied files to their original source files and issue error messages in case of discrepancies.

The keyword can be used to compare files on one reel as more files are being written to another reel. For example, you can compare the files that were just written to reel number 3 while writing more files to reel number 4.

A comparison of files on tape can start as soon as SYSTEM/COPY has finished writing files to that reel. If the COMPARE option is used, the comparison will not start until all the files have been copied to all the reels.

COMPARE as a keyword requires source and destination tapes to have identical directories (that is, to have the same files in the same order). Its use is based on the assumption that the two tapes were just copied.

MERGE

This copies files from any number of sources to up to 10 destinations, resolving any duplicates. If the same file exists on more than one source, it is only copied from the last source specified in the command string on which it exists.

If the same file exists more than once on a given source tape, the SPECIFY option must be used to ensure that duplicates do not occur on any tape destination.

If AS is used to rename files as they are merged, the new names are used in the check for duplicates.

MOVE

This copies files and then removes them from their sources. Files will not be removed from tape.

**AND
&**

These are synonymous.

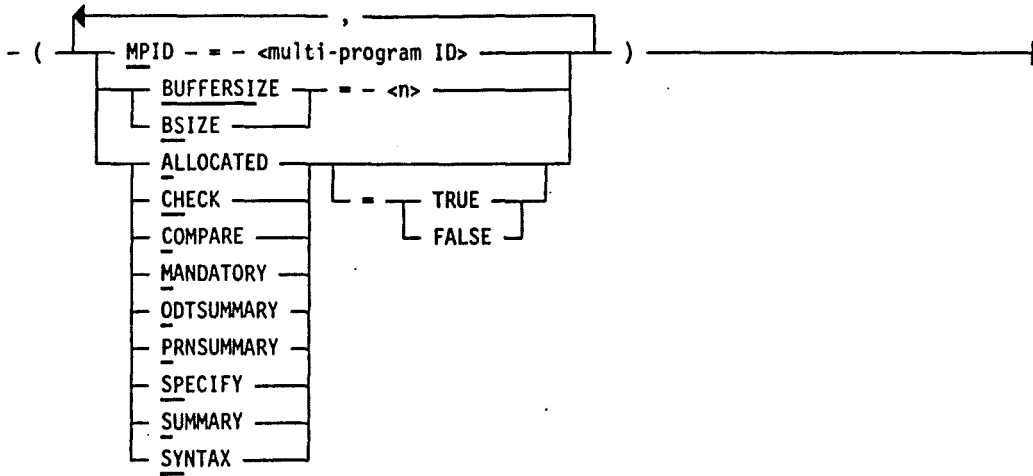
SET

This causes the options listed to be set.

<options>

Options can be used to complement the keywords. *Each option or string of options must be enclosed in parentheses.* More than one option can be used at a time, as long as options are separated by a comma. Underlining indicates the minimum abbreviation for

each option. The syntax for these options is shown in the following figure.



All options except MPID and BUFFERSIZE are Boolean-valued options. Therefore, SET (COMPARE) is equivalent to SET (COMPARE=TRUE).

MPID = <multi-program ID>

This assigns a name to an execution of SYSTEM/COPY. For example, three COPY functions running simultaneously can appear on the ODT as LARRY/COPY, MOE/COPY and CURLY/COPY. Multiprogram identifiers can be up to six alphanumeric characters in length.

BUFFERSIZE = <n>
BSIZE = <n>

This allows the user to specify the size of the internal buffers used by SYSTEM/COPY. When a file is copied to tape, the BUFFERSIZE option sets the blocksize that is written to tape. The value of *n* specifies the buffer size in digits, and can range from 3600 through 99000 in multiples of 1800. The default value is 18000.

ALLOCATED

This copies all source files onto predesignated sectors of disk or diskpack. All source files must already be present on all destinations. The source and destination files must have the same blocksize, blocks per area, and number of areas declared. However, the source and destination End of File can be different.

Use of this option results in the destination file being opened input/output, and the source file then being copied onto the sectors already occupied or "allocated" by the destination file. Any additional sectors needed will be acquired, and any unused sectors will be returned.

-
- This option does not allow a tape as a destination. The source and destination cannot be identical.
- CHECK** This checks newly copied files for parity errors. If the file is on tape, COPY also checks for block count errors. If the file is on disk or diskpack, COPY also checks for end-of-file errors.
- COMPARE** This is both a keyword and an option. It compares data between source and destination files and sends a message to the ODT if there are discrepancies. It is highly recommended that you use this option with the keyword MOVE because files will not be removed from their sources if there are discrepancies between the original and the new files.
- If an operation contains both a COMPARE option and an AS specification, the COPY function compares the original file to the new, renamed file.
- MANDATORY** This causes SYSTEM/COPY to require operator intervention if files are missing. If this option is not set, notification of missing files is still given, but the execution will continue.
- ODTSUMMARY** This displays a summary of the operation on the ODT. The summary details the execution and the names of the files involved.
- PRNSUMMARY** This outputs a summary of the operation to a print file. The summary details the execution, error messages, operator responses, and the serial numbers of any tapes involved in the operation.
- SPECIFY** This allows the user to resolve multiple occurrences of the same file on a given source tape. This option is valid only with the keyword MERGE. For each file in a tape directory, a check is made for any duplicates against the remaining files in that directory.
- If a duplicate is found, the following message will be displayed, and the user is given the opportunity to specify which file is desired:

```
SYSTEM/COPY=<mix no> TWO OCCURRENCES OF FILE <file name>  
REQUESTED FROM <tape name> (TAPE)  
AX 'E' TO SPECIFY EARLIER OCCURRENCE  
AX 'L' TO SPECIFY LATTER OCCURRENCE  
** SYSTEM/COPY=MX ACCEPT
```

After a choice is made, the check will then continue looking through the rest of the directory. If another occurrence is found, the message is again displayed, and the choice of EARLIER or LATTER is given. EARLIER now refers to the previously chosen file. This process will continue for each additional duplicate occurrence of the file found.

SUMMARY

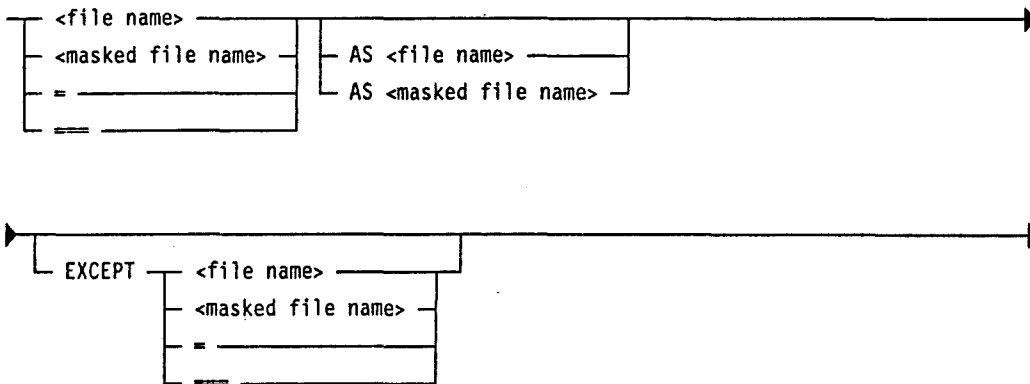
This displays a summary of the operation on the ODT and also generates a print file that details the execution, error messages, and operator responses.

SYNTAX

This causes SYSTEM/COPY to parse your commands without executing them. If there are any errors in the input, error messages will be displayed. If there are no errors, SYSTEM/COPY goes to EOJ.

<file specification>

This indicates the files involved in an operation, and also changes the names of files as they are being copied. If more than one file is involved in an operation, their names must be separated by commas. The following figure shows the syntax for the file specification clause.



For disk and diskpack files, the file names can be unique names (MYFILE), masked names (TEST=), or total group names (= and ===). When you use total group names to select group files from disk or diskpack media, there is a difference between = and ===. A single equal sign (=) excludes all special system files (control deck, backup, pass, and dump files). Three equal signs (===) includes them.

When you use the AS option, the following file specification combinations are not recommended:

- <file name> AS <masked file name>
- <masked file name> AS <file name>

The first combination, <file name> AS <masked file name>, would copy only one file. For example, ABFILE AS XY= would produce only XYFILE.

The second combination, <masked file name> AS <file name>, would copy all files matching the masked file name as the specified destination file as many times as the number of files that match the masked file name. For example, A= AS XYZ would copy all files with file names starting with A as XYZ. The file XYZ would contain only the last file found to match the masked source file name A=.

Caution

No checking is done to prevent duplicate files unless MERGE is used. COPY = AS A= FROM DISK TO PACK will replace the first character of each destination file name with an A. In this case, both source files X123 and Y123 would be renamed A123, and therefore one will overwrite the other.

Tape destination files can have multilevel file names through the use of the AS option.

The maximum size for multilevel file names is 10 levels of 17 characters per level. Slashes (/ /) are reserved for level separation. Quotation marks (' ') and parentheses are used for enclosing IDs with special characters. Equal signs (=) are used to mask file names and are only valid at the lowest ID level.

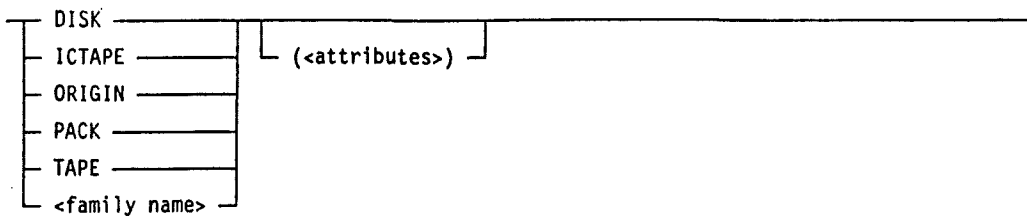
Tape files with multilevel IDs must be renamed using AS when they are reloaded from tape to disk or diskpack.

EXCEPT

This indicates that the files specified by <file name>, <masked file name>, or masking characters should not be copied.

<volume specification>

This indicates where to copy files FROM and where to copy files TO. Files from one or more sources can be copied to up to 10 destinations. Copies of all source files are sent to all destinations. The syntax for the volume specification clause is shown in the following figure.

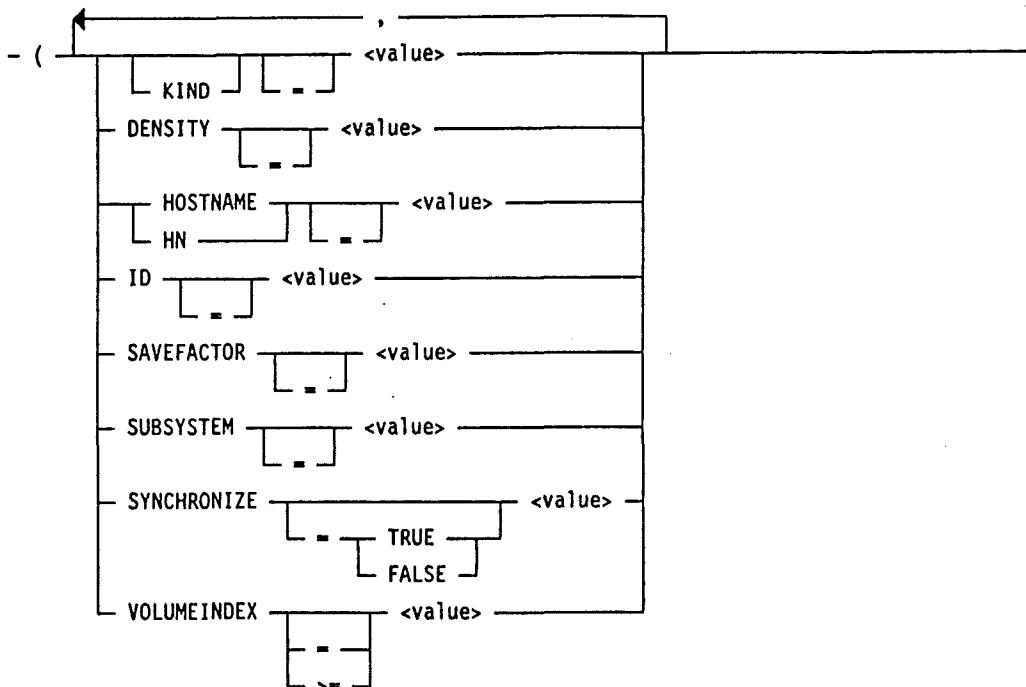


- DISK** This is a generic family name for system resource disk.
- ICTAPE** This is shorthand for a general source tape whose family name is SYSTEM. If the name for an ICTAPE is not specified when copying to an ICTAPE, the system assigns it the name SYSTEM by default. If the name of an ICTAPE is not specified when copying from an ICTAPE, the system attempts to locate a tape named SYSTEM.
- ORIGIN** This specifies the return of files from a tape to their origin. In other words, files copied from disk will return to disk, and those files copied from diskpack will return to diskpack. This is valid only for destination volume specifications.
- PACK** This is a generic family name for system resource diskpack.
- TAPE** This is shorthand for a general resource tape whose family name is SYSTEM. If a tape name is not specified when copying to a tape, the system assigns it the name SYSTEM by default. If a name is not specified when copying from a tape, the system attempts to locate a tape named SYSTEM.

Tape types are distinguished automatically on input.

Note: *Two tape formats exist, identified by the volume specifications TAPE and ICTAPE. The TAPE format requires less tape than ICTAPE and is read and written faster. Only ICTAPE, however, is compatible with older releases of SYSTEM/COPY (those prior to the 6.7.1 releases of MCPiX and MCPVI).*

- <family name>** This is the name of a diskpack family.
- <attributes>** These further specify source and destination media. The syntax for specifying attributes is shown in the following figure.



The information listed in Table 2-3 shows the supported attributes, their possible values, and whether the attribute is valid for a source or a destination volume specification.

Table 2-3. Valid Attributes for Source and Destination Volume Specifications

Attribute	Value	Source	Destination
KIND	DISK, PACK, TAPE, ICTAPE, OLDTAPE	Valid	Valid
	TAPE9, TAPEPE, TAPEGCR, TAPEMTC	Invalid	Valid
DENSITY	BPI800, BPI1600, BPI6250, BPI38000	Invalid	Valid
HOSTNAME, HN	1- to 17-character identifier	Valid	Valid
ID	1 through 89	Invalid	Valid
SAVEFACTOR	1 through 999	Invalid	Valid
SUBSYSTEM	0 through 8	Invalid	Valid
SYNCHRONIZE	TRUE, FALSE	Invalid	Valid
VOLUMEINDEX	1 through 999	Valid	Invalid

KIND

This specifies hardware types and is required for all user-named volumes. The volume names DISK, PACK, and TAPE have implied hardware types and the KIND attribute is not required to use them. Their values are as follows:

DISK	100-byte media
PACK	180-byte media
TAPE/ICTAPE	9 track NRZ, PE, GCR, or MTC
TAPE9	9 track NRZ only
TAPEPE	PE only
TAPEGCR	GCR only
TAPEMTC	MTC only
OLDTAPE	A LOADMP or PACKUP tape

SYSTEM/COPY does not support 7-track tape.

The KIND = syntax can be omitted. For example, (KIND=PACK) is equivalent to (PACK).

A specified value will override any implied value in the case of DISK and PACK. In other words, DISK (KIND=PACK) will be interpreted to mean a diskpack whose family name is DISK.

SYSTEM/COPY automatically recognizes and reads LOADMP, PACKUP, TAPE, and ICTAPE formats, so it is not necessary to indicate what kind of tape SYSTEM/COPY will be copying from. A volume kind of (TAPE) is all that needs to be specified when copying from tape. While SYSTEM/COPY automatically recognizes and reads those tape formats, it writes only in the TAPE and ICTAPE formats.

TAPE9, TAPEPE, TAPEGCR, and TAPEMTC result in a TAPE format tape of the corresponding density. To create an ICTAPE with a specific density, you must use the DENSITY attribute described below, as well as KIND = ICTAPE. For example the following command results in a tape of ICTAPE format and GCR density.

COPY FILEA TO MYTAPE (ICTAPE, DENSITY = BPI6250)

DENSITY	This allows you to specify density when KIND = TAPE or ICTAPE is indicated. The allowed values are as follows: BPI800 800 bits per inch - 9 track NRZ BPI1600 1600 bits per inch - PE BPI6250 6250 bits per inch - GCR BPI38000 38000 bits per inch - MTC
HOSTNAME HN	These are identical. They specify the BNA host where this volume resides. They are valid for both source and destination volumes; the HOSTNAME can be up to 17 characters long. When you use BNA for the COPY command, the blocksize of the file must be a value less than 49,999 bytes.
ID	This specifies the number of the logical disk unit. This attribute is restricted to disk destinations and can range from 1 through 89. All file areas will be assigned to the specified ID, with missing ID and no user disk conditions being resolved by the MCP.
SAVEFACTOR	This specifies the number of days beyond the creation date that a destination tape can be automatically purged by the MCP. It ranges from 1 through 999. The default value is 999.
SUBSYSTEM	This specifies the disk subsystem to be used for this volume. This attribute is restricted to disk destinations and has the range of 0 through 8 (0 being the default subsystems). All file areas will be assigned to identifiers defined within the specified subsystem, with undefined subsystem and no user disk situations being administered by the MCP. If this option is not used, disk native files will be returned to their original subsystems if possible.
SYNCHRONIZE	This determines whether data is transferred between the host and tape drive in buffered or unbuffered mode. <ul style="list-style-type: none">• When SYNCHRONIZE is FALSE, transmission is buffered (asynchronous). This maximizes tape drive performance.• When SYNCHRONIZE is TRUE, transmission is unbuffered. This attribute applies only to magnetic cartridge tape devices (KIND = TAPEMTC) and tape drives that use the Small Computer Systems Interface (SCSI).

When SYNCHRONIZE is not entered in a volume specification, the default value is FALSE. To set SYNCHRONIZE to TRUE, enter SYNCHRONIZE by itself or SYNCHRONIZE = TRUE.

VOLUMEINDEX

This applies to source tape volumes and is used to isolate a single reel of a volume set. Because each reel contains a directory, a file required from the third reel of a volume set can be loaded directly from that reel instead of having to process the first two reels sequentially. This attribute can also be used to overlap transfer of a complete volume set by initiating separate SYSTEM/COPY executions, each loading an exclusive VOLUMEINDEX. This attribute has the range of 1 through 999.

VOLUMEINDEX >= <value> can be used to load files from reel <value> through the end of that volume set. For example, to load all the remaining files from a volume set starting at reel number 2, use the syntax VOLUMEINDEX >= 2.

Examples

COPY FILE1 AS FILE2

This copies a file named FILE1 from system disk to system disk, renaming the file FILE2. Because both the source and destination volumes were not named, the system uses the system disk by default.

COPY AND SET (ODTSUMMARY, MPID=JOHN) MYFILE TO DEPACK (PACK)

This copies a file named MYFILE from disk to a diskpack named DEPACK, and sends a SUMMARY of this operation to the ODT. The MPID option gives this execution of SYSTEM/COPY the unique name JOHN/COPY.

MERGE AND SET (SUMMARY, MPID=ANZA) = FROM TAPE1 (TAPE), = FROM TAPE2 (TAPE) TO TAPE3 (TAPE)

This places a copy of all the files from TAPE1 and TAPE2 on TAPE3. If TAPE1 and TAPE2 have duplicate files, the duplication is resolved by taking the file from TAPE2, the latter tape. An ODT and printer summary are produced and the program will be in the mix as ANZA/COPY.

COPY X= TO TAPEA (ICTAPE)

This copies all files whose names begin with X from system disk to a tape, names the tape TAPEA, and writes the tape in ICTAPE format.

COPY = FROM TAPEA (TAPE)

This copies all the files on the tape named TAPEA to system disk. The COPY command automatically recognizes tape formats, making it unnecessary to specify the type of tape.

COPY AND SET (COMPARE) FILE1, FILE2 FROM FINBAR (PACK)

This copies two files named FILE1 and FILE2 from a diskpack named FINBAR to system disk. The system uses the system disk by default because no destination medium is defined. After all the copying is complete, the system compares the new files to their original files, and sends an error message if there are any discrepancies between the new and original files.

COPY AND SET (SUMMARY) = FROM SYSTEM (TAPE), = FROM UPDATE (TAPE) TO DISK

This copies all files from a tape named SYSTEM and a tape named UPDATE to system disk. In addition, the system sends a summary of the operation to the ODT and another, more detailed summary to a print file.

If any of the tape files already exist on disk, they will be replaced with the new files. If both tapes contain a file with the same name, only the file from the last tape named as a source will be copied.

MOVE AND SET (COMPARE) A=, B= FROM DISK, C= FROM PACK, D= FROM KTEL (PACK) TO MORGUE (PACK)

This transfers all the files whose names begin with A and all the files whose names begin with B from system disk to a diskpack named MORGUE; also transfers all files whose names begin with C from each unrestricted diskpack to a diskpack named MORGUE, and transfers all files whose names begin with D from diskpack KTEL to a diskpack named MORGUE.

After all files have been transferred, the compare option causes each newly copied file to be compared to its source file for accuracy.

COPY A= FROM MORGUE (PACK), B= FROM DISK TO MYTAPE (ICTAPE)

This transfers all files whose names begin with A from a diskpack named MORGUE to MYTAPE, and transfers all files whose names begin with B from the system disk to MYTAPE. This command also instructs the system to write MYTAPE in ICTAPE format.

COPY = FROM MYTAPE (TAPE) TO ORIGIN

This transfers all the files on a tape named MYTAPE back to the particular media from which they came. (The function invoked by ORIGIN works only when files are being transferred FROM tape.)

COPY

COPY = EXCEPT D3=, T3= FROM UTIL(PACK), = EXCEPT RLOG= FROM BACKUP(PACK)
TO W52Y88(TAPE)

This transfers all files whose names do not begin with D3 or T3 from a pack named UTIL and all files whose names do not begin with RLOG from a pack named BACKUP to a tape named W52Y88.

See also:

The discussion of SYSTEM/COPY in the *V Series Systems Operations Guide Volume 3: System Utilities*.

COPYADD (Transfer Files)

MCP Control Instruction

Access Level: 5

COPYADD invokes the SYSTEM/COPY utility to transfer files from one location to another. Only files whose names are not already present on the destination medium are transferred.

COPYADD cannot specify tape as an output medium.

See COPY (Invoke File Transfer Utility) for details about syntax and how to use the command.

See also:

The COPY (Invoke File Transfer Utility) command on Page 2-86.

CV, CVP (Convert Punch Backup File to Pseudo Card Deck)

Keyboard Input Message

Access Level: 5

The CV command initiates the LDCNTL utility to convert a punch backup disk or diskpack file into a pseudo card input deck. Punch backup files are always converted to pseudo decks on the same medium, whether disk or diskpack.

You can specify either an MCP or user-coded version of LDCNTL. The following figure shows the syntax for an MCP version of LDCNTL.

```

┌ CV ────────────────────┐ ┌───────────────────┐ ┌───────────────────┐
└ CVP ───────────────────┘ └ /MCP ───────────┘ └ SAVE ───────────┘
<file number>
  
```

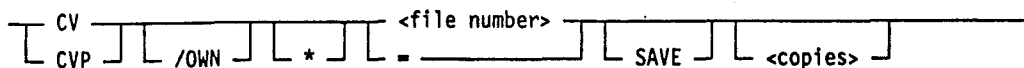
Explanation

CV CVP	CV indicates that the punch backup file to be converted is on disk; CVP indicates diskpack.
/MCP	This specifies the bound MCP intrinsic LDCNTL. If this option is omitted, the user-coded version is initiated if present on disk; otherwise the MCP intrinsic is initiated.
<file number>	This is the number of the punch backup file to be converted. A punch backup file in the mix is designated as *snnnn, where s is a system number and nnnn is a 4-digit number.
SAVE	This indicates that the punch backup file is not to be removed after the conversion.

User-Coded LDCNTL

A user-coded version of LDCNTL can include parameter entries for CV and CVP that resemble the <copies>, =, and asterisk (*) parameters of the PRINT command. The MCP LDCNTL program ignores these parameters. For details, see the PRINT (Print Files) command.

The *V Series MCP/VS Programming Reference Manual* contains instructions for creating a user-coded version of LDCNTL.



Explanation

/OWN	Using the /OWN option prevents the scheduling of the bound MCP intrinsic when a user-provided LDCNTL is required but is not on disk. If this option is omitted, the user-coded version is initiated if present on disk, otherwise the MCP intrinsic is initiated.
*	This informs LDCNTL that further specifications, such as skipping lines or restarting, will be provided through an AX response after the LDCNTL program begins operation. After locating the specified file, LDCNTL displays ENTER SPECS and issues an ACCEPT. The response to the ACCEPT must be in one of the formats shown in the AX syntax diagrams described under the PRINT (Print Files) command.
=	This indicates that all punch backup files on the specified media will be converted.
<copies>	This indicates the number of copies of the file you want. One copy is the default.

Details

The LDCNTL utility is a bound intrinsic or a user-coded program that creates pseudo decks from physical card decks. Pseudo card decks are created with the LD (Create Pseudo Card Deck) or EXECUTE LDCNTL commands. Use the CV or CVP command only if you want to convert a punch backup file into a pseudo card deck.

Examples

```
CV /MCP *10023 SAVE
```

This converts punch backup disk file *10023 into a pseudo card input deck and causes the file not to be removed after the conversion.

```
CVP /OWN =; CG 67073
```

This starts a user-coded version of the LDCNTL utility to convert all punch backup diskpack files into pseudo card decks. A user charge number of 67073 for this program is inserted into the MCP Run log records.

See also:

- The EXECUTE (Execute Library Program) command on Page 2-162.
- The LD (Create Pseudo Card Deck) command on Page 2-217.
- The PC, PCP (Create Punch Backup File) command on Page 2-275.
- The PRINT (Invoke PBDPRN) command on Page 2-291.
- The SD (Set Deck Limit) command on Page 2-342.
- The LDCNTL utility in the *V Series Systems Operations Guide Volume 3: System Utilities* and the *V Series MCP/VS Programming Reference Manual*.

DA, DAP (Deactivate Pseudo Reader)

Keyboard Input Message

Access Level: 5

The DA command deactivates a pseudo card reader on disk or diskpack. The following figure shows the syntax used to deactivate a pseudo card reader on disk.

```
- DA [ <system number> ] <deck number>
```

Explanation

DA This is used to deactivate pseudo card readers on disk.

<system number> If this is omitted, the number of the originating system is used.

<deck number> This must refer to a previously activated pseudo card reader.

Pseudo decks are designated as #snnnn. The *s* represents the system number and is not required unless the deck was created by a system other than the one from which you are entering the command. The *nnnn* is a 4-digit number with leading zeros assumed.

Do not include the pound sign (#) in the command syntax.

You can determine activated pseudo card readers with an OL PCR command.

Pseudo card readers are activated either with the RN (Activate Pseudo Reader) command or automatically by the MCP if the APCR option has been set to ON.

The following figure shows the syntax used to deactivate a pseudo card reader on diskpack.

```
- DAP [ <cc/u> ] [ <system number> ] <deck number>
```

Explanation

DAP

This is used to deactivate pseudo card readers on diskpack.

<cc/u>

This indicates the channel number and unit number of the diskpack on which to deactivate pseudo card readers. If <cc/u> is not specified, the first deck encountered with the specified system number, if indicated, and deck number is deactivated.

Examples

DA 153

This deactivates pseudo card reader number 153 on disk.

DA 3 1

This deactivates pseudo card reader on system number 3 with deck number 1.

DAP 6/0 1 32

This deactivates pseudo card reader on the diskpack on channel number 6, unit number 0, from system number 1, deck number 32.

See also:

- The CD, CDP (List Inactive Pseudo Decks) commands on Page 2-68
- The OL (Display Peripheral Status) command on Page 2-248.
- The RD, RDP (Remove Pseudo Decks) command on Page 2-313.
- The RN (Activate Pseudo Reader) command on Page 2-324.
- The USE APCR record in the *V Series Systems Operations Guide Volume 1: Installation*.

DATA/DATAB (Mark Beginning of Card Data File)

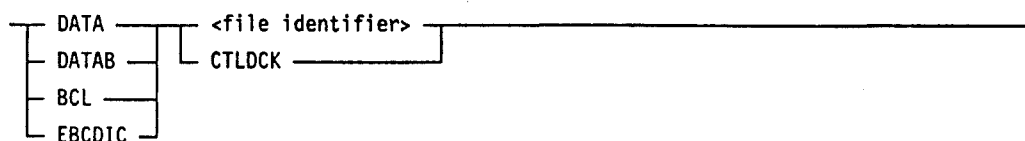
MCP Control Instruction

Access Level: 0

The DATA/DATAB command indicates the beginning of a punched card data file.

Whenever the DATA/DATAB command is encountered in the card reader, it causes an End-Of-File (EOF) condition on any previous file, and the unit is marked by the MCP as containing the new file.

This command is valid only from an actual or pseudo card deck.



Explanation

DATA
EBCDIC

These signify that subsequent input is in EBCDIC format. EBCDIC stands for Extended Binary Coded Decimal Interchange Code, a coded character set in which each character is represented by an 8-bit code.

DATAB
BCL

These indicate that the file is in BCL format. The BCL language is a coded character set in which each character is represented by a 6-bit code.

<file identifier>

This indicates the name of the file to follow.

Any information following the file identifier on the same punched card is ignored.

CTLDCK

This is a reserved file identifier that allows valid Control records to be included in the file. CTLDCK is the name of the physical card deck used by the LDCNTL utility to build a pseudo card deck.

Note: If <file identifier> is CTLDCK, no EOF action is taken until the ENDCTL command is encountered. This is primarily intended for use by the utility program LDCNTL for building pseudo card decks.

Examples

DATAB INPUT3

This specifies the beginning of a punched card data file called INPUT3. The file is in BCL format.

COMPILE PROG COBOL9 LIBRARY; DATA CARD

This compiles the COBOL program PROG using the COBOL9 compiler and then saves the new code file as a permanent file on disk. Then this statement specifies the beginning of a punched card data file called CARD. CARD is in EBCDIC format.

See also:

- The BCL (Mark Beginning of Card Data File) control instruction on Page 2-43.
- The BINARY (Read Binary Card Deck) control instruction on Page 2-57.
- The EBCDIC (Mark Beginning of Card Data File) control instruction on Page 2-153.
- The END/ENDCTL (Mark End of Card File or Control Instructions) control instruction on Page 2-156.

DBCA (Clear SSP Address Contention)

Keyboard Input Message

Access Level: 8

The DBCA command directs the MCP to clear the system contention flag for the specified addresses in a Shared System Processor (SSP) on the system.

DBCA allows you to clear all SSPs and all address contentions for the specified system; to clear one SSP and all addresses for the specified system; or to clear one SSP and one address.

This command is not intended for use during normal operations; it has been implemented primarily for system debugging.

```

- DBCA - <system number> = _____
                        |
                        | <SSP name> = _____
                        |               |
                        |               | <ID number> <address>
  
```

Explanation

- <system number> This designates the system (0 through 3) whose contention must be reset.
- <system number> = This clears all SSPs and all address contentions for the specified system.
- <SSP name> This is the mnemonic name given to an SSP on the DLP SSP record of the system configuration file.
- <SSP name> = This causes all address contentions for the specified system on the specified SSP to be cleared.
- <ID number> This is the 3-digit unit designation used to specify the ID number of the diskpack or disk device. The number can range from 1 through 255.
- <address> This designates the specific address (range from 0 through 16700000) to be cleared; the address must be decimal.

Details

An SSP is a type of Data Link Processor (DLP) that controls concurrent file access in a multi-system shared environment. The SSP stores addresses that have been locked in order to prevent simultaneous updating of shared files.

You can use the DBDR (Produce SSP Content Report) command or the SHOW SSP option of the SHOW (Display System Information) command to display on the ODT the current contents of all SSPs on the system.

Examples

DBCA 1 SSPABC =

This clears all address contentions in SSP SSPABC for system number 1.

DBCA 2 =

This clears all address contentions in all SSPs on system number 2.

See also:

- The DBDR (Produce SSP Content Report) command on Page 2-110.
- The SHOW (Display System Information) command on Page 2-343.

DBCL (Cancel DLP)

Keyboard Input Message

Access Level: 8

This command directs the MCP to send an unconditional cancel request to a specified Data Link Processor (DLP).

- DBCL - <channel number> _____ |

Explanation

<channel number> This represents the channel number of the DLP to be canceled.

Example

DBCL 6

This sends a cancel request to the DLP associated with channel number 6.

DBDR (Produce SSP Content Report)

Keyboard Input Message

Access Level: 8

The DBDR command provides a report on the ODT of the current contents of all the Shared System Processors (SSPs) on the system. This command is valid only if the system is equipped with an SSP.

DBDR is not intended for use during normal processing; it has been implemented primarily for system debugging.

- DBDR

Details

The SHOW (Display System Options) command has a parameter that displays an SSP or all SSPs on the system.

If DBDR is used, the system records the report information in the ODT log. However, if SHOW SSP is used, the system responds to the command but does not log it.

See also:

- The DBCA (Clear SSP Address Contention) command on Page 2-107.
- The SHOW (Display System Information) command on Page 2-343.

DBIC (Initialize Controller Parameters)

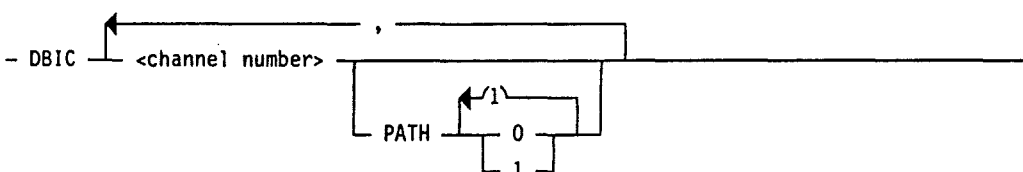
Keyboard Input Message

Access Level: 8

DBIC enables, disables, or displays the status of a path from a B 9387 controller to a dual exchange. It is valid only for sequential Data Link Processors (DLPs).

This command is used normally after analyzing the Maintenance log (MLOG) for the presence or absence of retry result descriptors for exchange paths.

The DBIC status is remembered across halt/loads and after controller host loads.



Explanation

<channel number>

This is a valid channel number. If you do not specify a PATH option, the status of the specified channel will be displayed.

PATH

0

1

This enables one or both paths. One path can be disabled by specifying the other path. For example, DBIC <channel> PATH 0 enables path 0 and disables path 1. Both paths are enabled by specifying both paths. For example, DBIC <channel> PATH 0 1 enables both paths.

Examples

DBIC 24, 7

This displays the status of channels 24 and 7.

DBIC 24 PATH 0, 7 PATH 1

This enables PATH 0 on channel 24 and PATH 1 on channel 7.

DBIC 24 PATH 0 1

This enables PATH 0 and PATH 1 on channel 24.

See also:

The *V Series System Software Logging Operations Reference Manual* for information about the Maintenance log.

DBMM (Set/Reset Maintenance Mode)

Keyboard Input Message

Access Level: 8

DBMM sets or resets the maintenance mode option available on 206, 207, 659, and 677 type diskpacks.

```
- DBMM 

|   |
|---|
| + |
| - |

 <cc/u>
```

Explanation

- | | |
|--------|--|
| + | This sets maintenance mode, making the unit unavailable to the system. |
| - | This resets maintenance mode and can return the unit to system availability, depending on the SAVED or RESERVED status of the unit. |
| <cc/u> | This is the channel number and unit number of the device. If you enter DBMM <cc/u>, the current maintenance mode status for the specified device is displayed. |

Details

Putting a drive in maintenance mode makes it unavailable to the system and allows offline testing of the drive.

Resetting maintenance mode makes the unit available for system operation. Maintenance mode can be reset only by issuing a DBMM - <cc/u> command or by powering off the drive locally.

The current maintenance mode setting is retained in the diskpack controller.

The DBMM status is remembered across halt/loads and is not affected by the UR (Inhibit or Disinhibit Unit) command, which inhibits units.

Note: This command is not valid for 680, MD4, MD8, or M9710 (SCSI) type diskpacks or MCP disk.

Examples

DBMM + 6/3

This sets maintenance mode on the diskpack on channel 6 unit number 3.

DBMM 4/2

This displays the maintenance mode status of the diskpack on channel 4 unit number 2.

DBSR (Produce Block Lockout Table Report)

Keyboard Input Message

Access Level: 8

The DBSR command requests the MCP to provide a report of the current contents of the MCP Block Lockout Table (BLT).

The BLT is a table that keeps track of which file blocks are locked and/or contended for by which programs. The BLT resides in memory and is updated by the MCP every time a program locks, unlocks, or contends for a file block.

If the system is part of a multi-system shared configuration, the status of the SSP lock for each BLT entry is also given.

DBSR is not intended for use during normal processing; it has been implemented for system and program debugging.

- DBSR -

Details

The SHOW (Display System Options) command has a parameter that displays an entry from a BLT or an entire BLT.

If DBSR is used, the system records the requested BLT information in the ODT log. However, if SHOW BLT is used, the system responds to the command but does not log it.

See also:

- The SHOW (Display System Informations) command on Page 2-343.

DBUA (Unlock Address)

Keyboard Input Message

Access Level: 8

DBUA initiates an unlock operation on the designated address. The DBUA command is valid only if the designated system is equipped with a Shared System Processor (SSP).

DBUA allows you to unlock all addresses and all SSPs for the specified system; to unlock all addresses on one SSP for the specified system; or to unlock one address on one SSP.

This command is not intended for use during normal operations; it has been implemented primarily for system debugging. It is useful for unlocking addresses held by a processor that has become unavailable for some reason.

```

- DBUA - <system number> = _____
                    |
                    | <SSP name> = _____
                    |
                    | <ID number> <address>
  
```

Explanation

<system number>	This designates the system (0 through 3) for which you want to perform the unlock operation.
<system number> =	This unlocks all SSPs and all addresses for the specified system.
<SSP name>	This is the mnemonic name given to an SSP on the DLP SSP record in the system configuration file.
<SSP name> =	This causes all addresses for the specified system on the specified SSP to be unlocked.
<ID number>	This is the 3-digit unit designation used to specify the ID number of the diskpack or disk device. The number can range from 1 through 255.
<address>	This designates the specific address (range from 0 through 16700000) to be unlocked; the address must be decimal.

Details

An SSP is a type of Data Link Processor (DLP) that controls concurrent file access in a multi-system shared configuration. The SSP stores addresses that have been locked in order to prevent simultaneous updating of shared files.

Examples

DBUA 0 -

This unlocks all addresses and all SSPs for system 0.

DBUA 2 FRED 12 50000

This unlocks address 50000 on ID number 12 in SSP FRED on system 2.

See also:

- The DBCA (Clear SSP Address Contention) command on Page 2-107.
- The DLP SSP record in the *V Series Systems Operations Guide Volume 1: Installation*.

DBUL (Clear Locked Files on Shared Systems)

Keyboard Input Message

Access Level: 8

In a shared system environment a system can fail and leave files locked on shared disk and diskpack media. The DBUL command directs the MCP to initiate a search for locked files on disk and shared diskpacks and unlock them.

The DBUL command can be used if the following are true:

- The designated system allows concurrent file access.
- The designated system is declared as connected for shared file access to the system from which this command is entered.
- The current state of the designated system is "dead". The SHOW SHARED keyboard command can be used to display the current state of a system.
- The system is equipped with a Shared System Processor (SSP).

```

- DBUL 

```

Explanation

<system number> This is the number (0 through 3) of the "down" system.

Examples

DBUL 0

This clears locked files on system 0.

DBUL 1, 2

This clears locked files on systems 1 and 2.

See also:

- The DBCA (Clear SSP Address Contention) command on Page 2-107.
- The DBSR (Produce Block Lockout Table Report) command on Page 2-114.
- The DBUA (Unlock Disk Address) command on Page 2-115.
- The SHOW (Display System Information) command on Page 2-343.
- The CONNECT configuration record in *V Series Systems Operations Guide Volume 1: Installation*.
- The discussion of shared systems in *V Series Systems Operations Guide Volume 3: System Utilities*.

DC (Display Compilation Data)

Keyboard Input Message

Access Level: 1

The DC command displays the following information on the ODT about a specified program residing on disk or diskpack:

- Program name
- Date compiled
- Compiler type (first character of the compiler name used to create this program)
- Memory requirement
- Internal program name or compile time

DC can also be used for an Independently Compiled Module (ICM). ICMs are modules that can be bound together with other modules to create a single program. There are different types of ICMs. The types restrict the language or languages to which that ICM can be bound.

The display will provide the ICM name, version level, internal ID (can contain compiler name and release number depending on ICM type), compile date, and time of compilation.

```
- DC _____
  |_____| |_____|
  |<program name>| |ON <family name>|
  |_____| |_____|

- DC - /MCP <bound program name> _____
```

Explanation

DC	If this is entered without any options, the system displays the MCP name, release level, patch level, and date and time compiled of the currently running MCP.
<program name>	This designates the name of the program whose compile information you want to display. A program name can have up to six characters. The first character must be a letter of the alphabet. If you are using the /MCP option, specify a bound program name.
ON <family name>	This indicates that the specified <program name> is located on a family labeled with the name indicated by <family name>.

If you are using the DC command to display a bound program, you cannot specify ON <family name>. Bound programs are always on disk.

/MCP

This indicates that the specified program is bound to the MCP.

Details

The memory requirement specified does not include any additional space the program might require after it begins execution (for example, disk file header space).

Example

DC COBTY2

This causes the ODT to display the program name, data compiled, compiler type, memory requirement and the date and time of compilation.

Note: Some compilers display the internal program name rather than the time of compile.

Displaying ICM Compilation Data

- DC - <ICM name> _____

Explanation

<ICM name>

This is the name of the ICM for which the compilation data is requested.

The ICM cannot be a FORTIV ICM, XFORTN ICM, or a BPL type 1 ICM.

Example

DC COBICM

This causes the ODT to display the ICM name, version level, compiler name, internal ID, and the date and time of compilation.

See also:

- The DISPLAY/WHATS (Display Information) command on Page 2-131.
- The FN (List File Names) command on Page 2-177.
- The WB (Display Bound Programs) command on Page 2-405.

DD (Delete DLP)

Keyboard Input Message

Access Level: 8

This command removes a DLP from the system environment. This allows you to change the firmware ID, hardware type, or other information about the DLP, and then re-declare the DLP.

The primary channel of an exchange is the first DLP declared on the EXCHANGE record; the primary channel has the same number as that of the exchange number. If the primary channel has no attached units, it can be deleted.

All other channels on an exchange are alternate channels. If the channel to be deleted is an alternate channel of an exchange, you must use the XA (Add/Remove a Channel) command first, followed by the DD command to delete the channel.

- DD - <channel number>

Explanation

<channel number>

This indicates the channel number of the DLP you are deleting. If the channel number is not present, the system displays the following message:

DLP <channel number> NOT PRESENT

If you attempt to delete a primary channel that has units attached, the system displays the following message:

UNIT xx/yy EXISTS ON THIS CHANNEL - CANNOT DELETE

Use the DL (Delete Peripheral Devices) command to delete units from the primary channel. You can then delete the primary channel.

See also:

- The DL (Delete Peripheral Devices) command on Page 2-135.
- The XA (Add/Remove a Channel from an Exchange) command on Page 2-432.

DEBUG (Attach Program to DEBUG Session)

MCP Control Instruction

Access Level: 1

The DEBUG command starts a program and simultaneously attaches the ODT to a DEBUG session for that program if the DBUG option is set. This command is entered only through an ODT.

```
- DEBUG - <program name> _____
                | ON <medium name> | | <parameter list> |
```

Explanation

<program name> This is the name of the program that will be attached to an interactive debug session. The file must contain executable object code and must be stored on the disk or diskpack entered as **<medium name>** if specified. Otherwise, it will be stored on disk.

<medium name> This refers to the name of the storage device where the desired program is stored. The **<medium name>** can be one of the following choices:

DISK This looks for the desired program on 100-byte media only. If the program is not found, an error message is displayed on the ODT.

PACK This looks for the desired program on all of the system's unrestricted diskpacks. If the program is not found, an error message is displayed on the ODT. The program cannot be found unless it is stored in 180-byte code file format.

To ensure that the program is stored in the correct format, copy it onto the diskpack using the PCOPY utility or compile the program using the **<code file media ID>** option of the COMPILE (Compile Program) command.

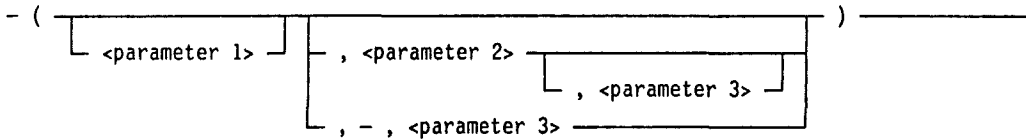
A diskpack family name This looks for the desired program on the diskpacks of the entered family name. If the program is not found, an error message is displayed on the ODT.

If <medium name> is not entered, the operating system searches all 100-byte disks for the desired program. If the program is not stored on disk, the operating system continues searching for the program on all unrestricted diskpacks. If the program is not found on disk or on unrestricted diskpack, an error message is displayed on the ODT.

<parameter list>

These are up to three parameters that can be inserted into a program when the program is started.

The <parameter list> option appears in the syntax of the COMPILE (Compile Program), DEBUG, and EXECUTE (Execute Library Program) commands. These parameters allow you to enter Boolean, string, and integer values, which can be used by the program when the program is being compiled or executed. The syntax for <parameter list> is shown in the following figure.



The rules for <parameter list> resemble those for parameters in the Work Flow Language (WFL) RUN command.

Integer parameters

- Contain unsigned numeric data up to eight digits in length (8 UN)
- Have values ranging from 0 through F
- Are right-justified and zero-filled in the high-order digits if less than the maximum length
- Will cause a syntax error if greater than the maximum length

String parameters

- Contain alphanumeric data up to six characters in length (6 UA).
- Must be within quotation marks. To pass a quoted string as a parameter, a double quote must appear in the parameter string (for example, DEBUG MYPROG ('TRY' 'XX')).
- Can be composed of any EBCDIC characters.
- Are left-justified and blank-filled in the low-order bytes if less than the maximum length.

- Will be truncated in the low-order bytes if greater than the maximum length.
- <parameter 1> This can be an integer up to eight digits in length. It sets the switches at program address 0. For example, DEBUG MYPROG (10100000).
- <parameter 2>
<parameter 3> This can be 8-digit integers or 6-character strings. The value of <parameter 2> is inserted into the code file at base-relative address 8; The value of <parameter 3> at base-relative address 20. For example, DEBUG MYPROG (,12345678) or DEBUG MYPROG (101,'TOTEST').
- When either <parameter 2> or <parameter 3> are specified, you must have either a value or at least a comma in the preceding parameters. For example, DEBUG MYPROG (,,'TAPE').
- A null entry is allowed. For example, DEBUG MYPROG (,0,'ABC''''') does not change the switches and inserts 00000000 at address 8 and the string 'ABC'''' at address 20.

Note: Slash parameters were available on previous releases of the MCP. Later releases will not support slash parameters. The <parameter list> option replaces them.

Details

The interactive DEBUG module is used to debug user programs or the operating system itself. DEBUG can

- Perform DEBUG sessions concurrently
- Establish program breakpoints
- Output DEBUG tracings to the ODT, a printer, tape, or to a printer backup file on disk
- Debug all or part of a task
- Allow modification of memory within the task being debugged

The GT (Initiate Trace) and TRACE (Trace A Program) commands start tracing facilities for user programs only. These tracing facilities do not have all the capabilities of the DEBUG module.

Related Commands

The ID command (Invoke DEBUG Session) starts a DEBUG session for a program that is already running and attaches the ODT to that DEBUG session.

The ED (Enter DEBUG Session) and QD (Query DEBUG Sessions) commands, like the ID command, manipulate DEBUG sessions. The ED command stops the ODT from displaying system information and returns it to a previously started DEBUG

session. The QD command displays the number of active debug sessions on the system with their corresponding session numbers.

Effect on the Run Log

The DEBUG command sets the 2-bit in the supplementary execution code field (RL-SXE, which is digit 71) of the job's Job Schedule record (type 3/0), the Short Job Schedule (type 4/0), and Beginning-of-Job record (record type 6/0) in the system Run log.

If one program starts another program using the ZIP construct, the DEBUG command can be included in the ZIP syntax. In this way an entire system of programs can be correctly flagged in the Run log, as well as the one originally executed.

Examples

```
DEBUG PAYRL1
```

This looks for a program named PAYRL1 on disk, CODEPACK, or unrestricted diskpack. It then starts the program and attaches it to an interactive debugging session.

```
DEBUG UPDATE ON PACK
```

This looks for a program named UPDATE on all unrestricted diskpacks, starts the program, and attaches it to an interactive debugging session.

```
DEBUG STAT01 ON ACCTS ; FILE OUTPUT = LINEPR/ST1985
```

This looks for a program named STAT01 on the diskpack family named ACCTS, starts the program, and directs the file named OUTPUT to a specific printer.

See also:

- The COMPILE (Compile Program) command on Page 2-80.
- The ED (Enter DEBUG Session) command on Page 2-155.
- The GT (Initiate Trace) command on Page 2-183.
- The ID (Invoke DEBUG Session) command on Page 2-191.
- The QD (Query DEBUG Sessions) command on Page 2-307.
- The SO (Set System Options) command on Page 2-352.
- The TRACE (Initiate Trace) command on Page 2-381.
- The USE DBUG record in the *V Series Systems Operations Guide Volume 1: Installation*.
- The DEBUG utility in the *V Series Systems Operations Guide Volume 3: System Utilities*.

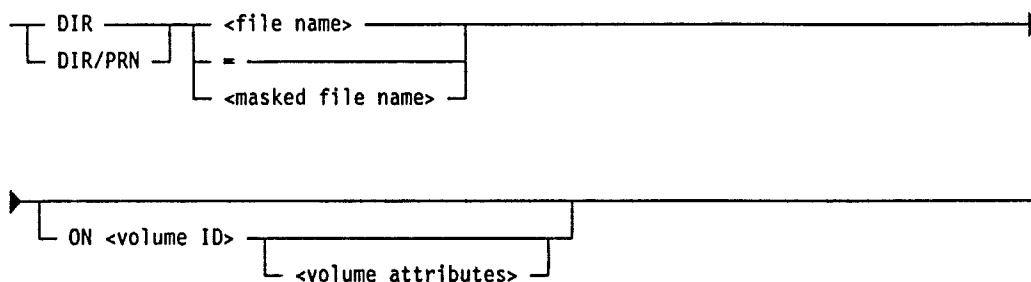
DIR (List File Directory)

MCP Control Instruction

Access Level: 5

The DIR command invokes the SYSTEM/COPY intrinsic, which searches file directories of tape, diskpack, or disk. The command is used to search a directory for one file or one group of files. It can also list all files in the directory. DIR searches tape directories produced by SYSTEM/COPY or user-supplied LOADMP and PACKUP utilities. File names are displayed on an ODT. A printed version of the file list can be created.

DIR can be used on SYSTEM/COPY tapes to determine which reel of a multi-reel volume set contains a particular file. Each reel contains a directory that lists only the files stored on the current and subsequent reels. Analyze the directory of each reel, beginning with the last, until the file is listed. The file is contained in the first reel whose directory shows the file.



Explanation

DIR
DIR/PRN

DIR displays the file list on the ODT. DIR/PRN displays the file list on the ODT, and also places the list in a printer backup file for later printing.

If DIR/PRN is performed on a SYSTEM/COPY tape, the printer backup file contains the list of file names, the tape name, the serial number, the creation date, the tape format, the reel number, first file number, and the library level of the tape.

If DIR/PRN is performed on a LOADMP/PACKUP tape, the printer backup file contains only the file list, tape name, the serial number, the creation date, and the tape format.

<file name>
=
< masked file name>

This specifies the files in the directory. The equal sign (=) can be used as a masking character to search for a group of files. If one equal sign is entered as the <file name>, all of the files in the directory are listed.

<volume ID>	This determines which directory will be searched. The volume ID can be one of the following choices:
<tape name>	This searches the directory of a specific magnetic tape.
<family name>	This searches the directory of a specific diskpack family.
DISK	This searches the directory for all 100-byte media.
PACK	This searches directories of all unrestricted diskpacks on the system.
TAPE	This searches the directory of a tape with the default name of SYSTEM.

If no **<volume ID>** is entered (that is, if the ON phrase is not used), DISK will be assumed. More details about **<volume ID>** and **<volume attributes>** are explained in the volume specification syntax of the COPY command earlier in this section.

<volume attributes> These apply to named diskpacks or tapes. The **<volume attributes>** further define the device entered as **<volume ID>**. If an individual tape name is entered as **<volume ID>**, the **<volume attribute>** must be TAPE. An individual diskpack name must be followed by a **<volume attribute>** of PACK. The **<volume attributes>** of a SYSTEM/COPY tape can optionally include a VOLUMEINDEX attribute.

See the description and syntax under the COPY (Invoke File Transfer Utility) command for a complete explanation of these attributes.

Details

The directory on a continuation reel of a multi-reel set will probably not be found at the beginning of the reel; instead, it is located past the end of the file that is continued from the preceding reel. Depending on the length of the continued file, the directory can be some distance into the reel. If a single file requires more than one full reel, it is possible that one or more continuation reels will not have directories.

Examples

```
DIR = ON TAPE1 (TAPE)
```

This displays all the files stored on a tape named TAPE1.

DIR TEST= ON PACK

This searches all unrestricted diskpacks and displays all files whose names begin with the characters TEST.

DIR PR= ON PMDUMP (TAPE, VOLUMEINDEX = 5)

This searches the fifth reel of a SYSTEM/COPY tape named PMDUMP and displays all files whose names begin with the characters PR.

DIR/PRN = ON PUBLIC (PACK)

This searches a diskpack named PUBLIC, displays all files stored on that diskpack, and places the file list into a printer backup file for later printing.

See also:

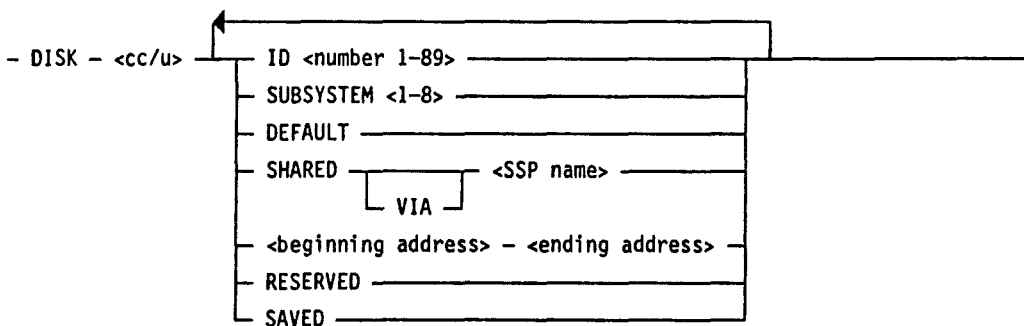
- The COPY (Invoke File Transfer Utility) command on Page 2-86.
- The PD (Print Directory) command on Page 2-278.

DISK (Add a Disk)

MCP Control Instruction

Access Level: 8

The DISK command adds a disk to the MCP environment. The syntax for the DISK command is identical to that for the DISK record of the system configuration file.



Explanation

- | | |
|------------------|--|
| <cc/u> | This is the channel number and unit number of the disk spindle you want to add to the system. |
| ID <number 1-89> | This indicates the disk ID number and can be any number from 1 through 89. |
| SUBSYSTEM <1-8> | Disk devices can be grouped into logical subsystems. Up to 8 logical subsystems can be declared. This element is required.

Indicate the number of the logical subsystem to which this spindle belongs. |
| DEFAULT | This is optional and declares this spindle as a default disk.

A default disk is a subsystem that can be used without specific assignment. If a program opens an output file on disk without requesting a specific subsystem or disk ID, the output file is assigned automatically to any disk in any subsystem declared as a default disk.

If one spindle on a logical subsystem is specified as DEFAULT, all spindles on that subsystem must be declared DEFAULT as well.

More than one logical subsystem can be declared default. |

If no subsystem has been declared default, logical subsystem 01 becomes the default subsystem and the residence of the MCP code file. This is not recommended for performance reasons.

SHARED <SSP name>
SHARED VIA <SSP
name>

Use this option if you are declaring a system in a multi-system shared configuration and want to share disks among the different systems.

Indicate the name of the Shared System Processor (SSP) that controls the concurrent file access between systems. Names are assigned to SSPs with the DLP SSP record.

If the files on this spindle are to be used by programs running on the other systems in the shared system, the spindle must be declared SHARED in each system's configuration files. It must also be declared with the same ID number and as part of the same logical subsystem.

If one DISK drive on a subsystem is declared shared, all DISK spindles on that subsystem must be declared shared.

<beginning address>
<ending address>

These are optional and can be used to specify the beginning and ending address of the disk spindle, indicating to the MCP the amount of room the spindle has for data. If the entire spindle is to be used, the MCP will automatically determine the proper capacity.

The maximum address ranges possible for various types of disk and LAK (look-alike) diskpack spindles are given in Table 2-4.

Table 2-4. DISK/LAK Diskpack Maximum Address Ranges

Spindle Type	Beginning	Ending
5N disk (4 DSs)	0	221183
235 diskpack as LAK	0	969144
206I diskpack as LAK	0	565729
206S diskpack as LAK	0	476189
207I diskpack as LAK	0	998854
207S diskpack as LAK	0	998854
659I diskpack as LAK	0	999999
659S diskpack as LAK	0	999999

Table 2-4. DISK/LAK Diskpack Maximum Address Ranges (cont.)

Spindle Type	Beginning	Ending
677I diskpack as LAK	0	999999
677S diskpack as LAK	0	999999
MD4	0	682259
QWIK disk	0	999999
M9710	0	3746952

RESERVED

This reserves the unit after a cold start or halt/load. The drive remains reserved, unavailable to the system until made ready with the UR (Inhibit or Uninhibit Unit) command. The MCP disk cannot be reserved.

SAVED

This puts the drive in a saved state after a cold start or halt/load. The MCP disk cannot be saved. See the SV (Save a Peripheral Unit) for more information.

Examples

DISK 6/3 ID 2 SUBSYSTEM 3 RESERVED

This adds to the system environment the disk spindle on channel 6, unit number 3. This spindle has a disk ID number of 2 and belongs to logical subsystem 3. After a cold start or halt/load this disk will be made unavailable to the system until unreserved by a UR (Inhibit or Uninhibit Unit) command.

DISK 6/4 ID 5 SUBSYSTEM 5 DEFAULT

This adds to the system environment the disk spindle on channel 6, unit number 4. This spindle has a disk ID number of 5 and belongs to logical subsystem 5. This disk will be a default disk, and subsystem 5 will be a default subsystem.

See also:

- The SV (Save a Peripheral Unit) command on Page 2-368.
- The UR (Inhibit or Uninhibit Unit) command on Page 2-399.
- The DISK and DLP SSP records in the *V Series Systems Operations Guide Volume 1: Installation*.

DISPLAY/WHATS (Display Information)

Keyboard Input Command

Access Level: 5

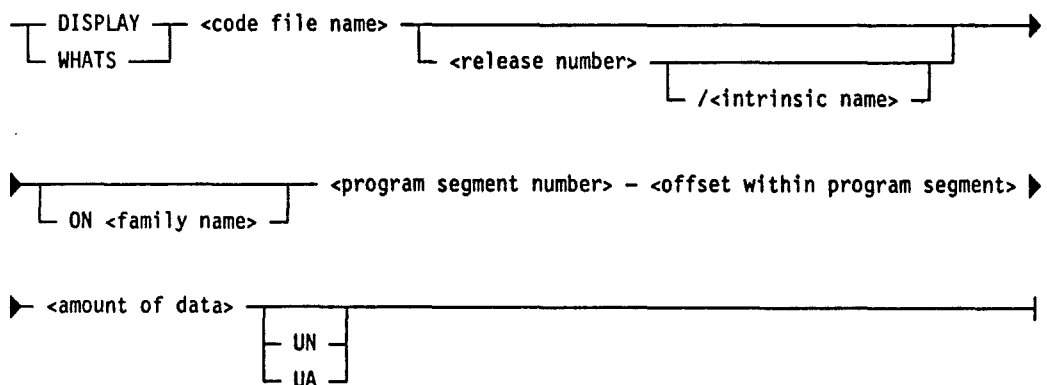
The DISPLAY/WHATS command displays code file or disk/diskpack file information on the ODT. DISPLAY can obtain and display information from the following sources:

- A program code file, the MCP code file, or an MCP-bound intrinsic code file
- A disk or diskpack file
- An absolute disk sector

The DISPLAY command is most often used in conjunction with the PATCH (Patch File) command. Before patching a given location within a code file or file, display the information stored at that location. This lets you verify that the sector numbers, offsets, and other values to be used in the PATCH command are calculated correctly.

The following describes each different use of the DISPLAY command.

Displaying a Program Code File, MCP Code File or MCP Intrinsic Code File



Explanation

DISPLAY
DISP
WHATS

These are synonymous.

<code file name>

This specifies the name of a program code file to be displayed. To display the MCP code file or an MCP intrinsic code file, enter the name of the code file containing the currently running operating system.

DISPLAY/WHATS

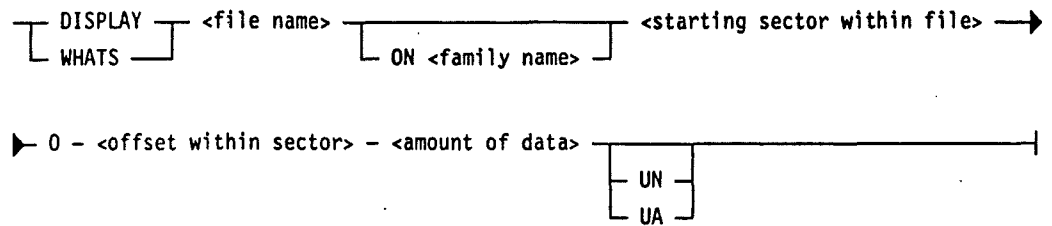
<release number>	This refers to the release level of the currently running MCP. Do not enter <release number> unless the code file to be displayed is either the MCP code file or that of an intrinsic program bound to the MCP. For <release number> to be valid, the name entered as <code file name> must be the name of the currently running MCP code file.
/ <intrinsic name>	This is the name of an intrinsic program bound into the MCP. For <intrinsic name> to be valid, the name entered as <code file name> must be that of the currently running MCP code file.
<family name>	This is the name of the family where the file or code file is stored. If the ON clause is not entered, DISK is assumed.
<program segment number>	This indicates the number of the actual segment within the program where the desired information is located. A valid entry can be any number from 0 through 999. If the MCP code file is being displayed, this is the environment number (in decimal) of the code area of the desired MCP module or overlay.
<offset within program segment>	This specifies the offset within the program segment where the desired information is located. This offset is expressed in digits.
<amount of data>	This specifies the amount of information to be displayed. The maximum <amount of data> is 30 digits or bytes.
UN UA	These represent the format of the information to be displayed. UA is alphanumeric; the default is UN (numeric).

Example

```
DISPLAY MONITR XO 20 10 UN
```

This displays information within a program code file named MONITR that is stored on disk. The operating system will display the 10 digits of numeric information that are stored 20 digits from the beginning of the first segment of the program.

Displaying a Disk or Diskpack File



Explanation

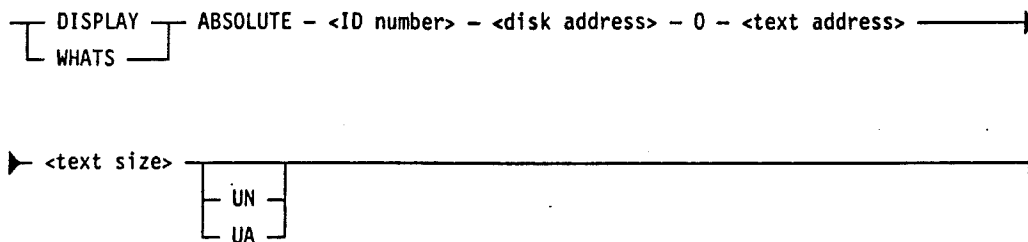
<file name>	This specifies the name of a disk file or diskpack file to be displayed.
<family name>	This is the name of the diskpack family where the file is stored. If the ON clause is not entered, DISK is assumed.
<starting sector within file>	This is the relative sector within a disk or diskpack file where the desired information is located. The first sector within the file is numbered zero. Disk sectors are also referred to as segments.
0	This separates the starting sector number from the offset number.
<offset within sector>	This is the number of digits from the beginning of the disk or diskpack sector where the desired information is located. The first digit within the sector is numbered zero. Disk sectors are also referred to as segments.
<amount of data>	This specifies the amount of information to be displayed. The maximum <amount of data> is 30 digits or bytes.
UN UA	These represent the format of the information to be displayed. UA is alphanumeric; the default is UN (numeric).

Example

```
DISP ACTR01 ON ACCTNG 1 0 1 10 UN
```

This displays information within a file named ACTR01 stored on a diskpack named ACCTNG. The operating system will display the 10 digits of numeric information that are stored 1 digit from the beginning of the second sector of the file. (The first sector of the file is numbered 0, as is the first digit within that sector.)

Displaying an Absolute Disk Sector



Explanation

<ID number>	This specifies the logical ID number of the disk, and can be any number from 1 through 255.
<disk address>	This indicates the number (in decimal) of the absolute disk sector that contains the information to be displayed. This is also called a physical disk address. This address can be 1 to 12 digits long. Disk sectors are also referred to as segments.
0	This separates the absolute disk sector address from the text address number.
<text address>	This indicates the displacement in digits from the beginning of the disk sector being displayed. The first digit within the sector is numbered zero. There are 200 digits in a disk sector. Disk sectors are also referred to as segments.
<text size>	This is the number of digits to be displayed. The maximum <text size> is 30 digits.
UN UA	These represent the format of the information to be displayed. UA is alphanumeric; the default is UN (numeric).

Example

WHATS ABSOLUTE 32 140010 0 20 10 UA

This displays information starting at sector address 140010 on the disk drive that has an ID number of 32. The operating system will display the 10 digits of alphanumeric information that are stored 21 digits into the sector address 140010. (The first digit of the sector is numbered 0.)

See also:

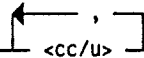
- The PATCH (Patch File) command on Page 2-264.
- The SHOW (Display System Information) command on Page 2-343.

DL (Delete Peripheral Devices)

Keyboard Input Message

Access Level: 8

The DL command deletes peripheral devices from the list of peripherals maintained by the system.

- DL  `<cc/u>`

Explanation

`<cc/u>`

This is the channel number and unit number of the device to be deleted. More than one device can be deleted by entering several channel number and unit number pairs, separated by commas.

Details

You cannot enter DL while the desired peripheral device is assigned to an executing program.

The UNIT (Add Peripheral Devices) command can be used to add a unit back in after it is deleted.

Several records in the system configuration file associate peripheral devices with their respective channel and unit numbers. Applicable records include the DISK record, PACK record, and the various UNIT records.

Examples

DL 6/1

This deletes the peripheral device designated as unit number 1 on channel number 6.

DL 4/4, 12/1

This deletes the device designated as channel 4, unit 4 and the device designated as channel 12, unit 1.

See also:

- The DD (Delete DLP) command on Page 2-120.
- The DLP (Add DLP) command on Page 2-137.
- The DISK (Add a Disk) command on Page 2-128.

- The PACK (Add a Pack) command on Page 2-262.
- The SV (Save a Peripheral Unit) command on Page 2-368.
- The UNIT (Add Peripheral Devices) command on Page 2-385.
- The UR (Inhibit Or Uninhibit Unit) command on Page 2-399.
- The DISK, PACK, and UNIT records in the *V Series Systems Operations Guide Volume 1: Installation*.

DLP (Add DLPs)

MCP Control Instruction

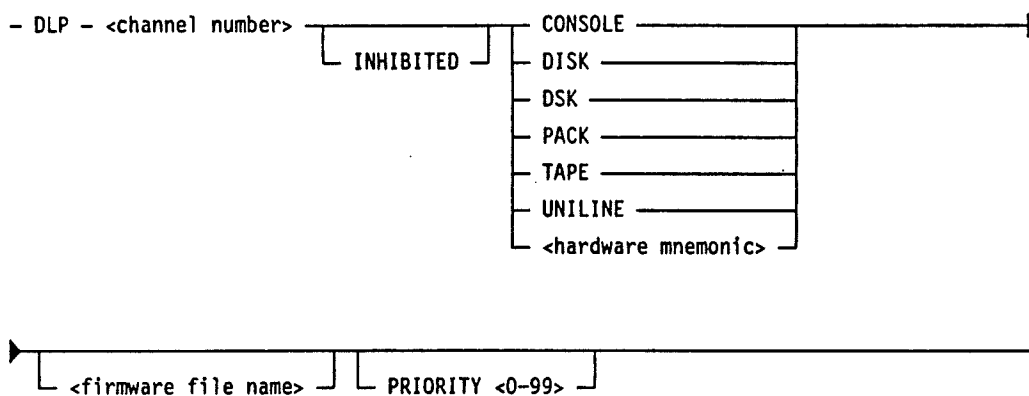
Access Level: 8

The DLP command adds Data Link Processors (DLPs) to the system environment while the operating system is running. The syntax of the DLP command differs when peripherals are shared between more than one system.

The syntax for the DLP command corresponds to the syntax for the DLP and DLP SSP records used in the system configuration file.

Adding a DLP to a Non-SSP System

The following syntax diagram shows the syntax used for non-SSP systems.



Explanation

<channel number>	This is the number of the physical channel where the DLP resides.
INHIBITED	This enters the DLP into the system environment in an inhibited state, that is, unavailable for use. (You can use the XC (Inhibit/Uninhibit Channel) command to uninhibit the channel.)
CONSOLE	This sets up the channel to serve a Universal Console. If you use this option, do not declare a <firmware file name>.
DISK DSK	This declares that the DLP supports disk and diskpack units (100-byte media and 180-byte media).
PACK	This indicates that the channel supports diskpack devices.

TAPE This indicates that the channel supports tape devices. GCR, MPE, MTP and MT9 tape devices can be declared using either the keyword **TAPE** or the hardware mnemonic of the device. Tape devices introduced in the 2.2 and succeeding releases of the operating system are declared using *only* the keyword **TAPE**. This includes MTC tape devices.

UNILINE This declares a UNILINE DLP.

<hardware mnemonic> This represents the type of DLP being added if it is other than disk, diskpack, SSP, universal console, or UNILINE DLP. The following lists these mnemonics and the corresponding devices.

Mnemonic	Hardware Device
CRD	80-column Card Reader
DCP	Data Communications Processor
GCR	GCR Magnetic Tape Drive
ISC	Inter-System Connect
MPE	Phase-Encoded (PE) Magnetic Tape Drive
MT9	Non-return to Zero Magnetic Tape Drive
NCP	Network Communications Port
NST	Non-Status Device
OCS	Operator Control Station
ODT	Operator Display Terminal
PCH	80-column Card Punch
PRN	Buffered Printer
RJE	Remote Job Entry Device
S4A	9137 Type Reader Sorter
S4B	9138 Type Reader Sorter
TC5	TC 500 Device
TWX	Teletypewriter through a UNILINE DLP
TPR	Train Printer
VDD	B 20 or ET Series Workstation or a B 9352 or TD 800 Series Terminal

<firmware file name> This is the name used as the default firmware file name for the DLP being added.

PRIORITY <0-99>

This assigns a relative priority to the channel. The value can be any number from 0 through 99, with 99 being the highest priority.

It is recommended that you assign each channel a priority so that when I/O operations occur simultaneously, the system knows which one to process first. In such a case, the channel with highest priority gets processed first.

If one or more channels have the same priority, the channel declared first is serviced first.

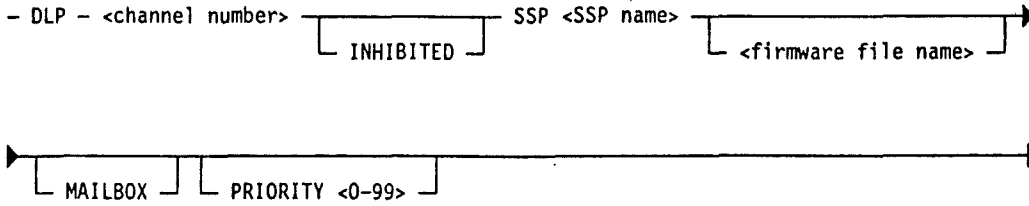
If channels are not assigned priorities, they are serviced in the order in which they are declared. That is, the channel declared first is serviced first.

The following table lists the recommended priority settings for the various devices.

Hardware Type	Priority Range
DCP, ISC, DC-DLP, NCP	96-97
Disk and diskpack	90-95
Magnetic tape	80-89
Console DLP	70
UNILINE DLP	60-69
Printer devices	40-49
Card devices	20-29

Adding a DLP to an SSP System

The following syntax diagram shows the syntax used for SSP systems.



Explanation

- <channel number> This indicates the channel the SSP is connected to. It is recommended that each system in a multi-system shared configuration use the same channel number for the same SSP.
- INHIBITED See the description of this option under "Adding a DLP to a Non-SSP System" earlier in this command description.
- SSP <SSP name> This declares that the channel serves an SSP DLP in a shared base. The <SSP name> identifies the SSP. It is recommended that all systems in a multi-system shared configuration use the same name for the same SSP. This name will be used in DISK and/or PACK UNIT records that are declared as SHARED.
- <firmware file name> This documents the name of the firmware file used by the SSP.
Note that loading firmware to an SSP clears all its lockout entries. Load firmware to an SSP only when all the systems in the configuration are not running.
- MAILBOX This declares that the SSP being added contains the intersystem communication mailbox used to access shared peripheral devices. There can be only one MAILBOX SSP per shared system.
- PRIORITY <0-99> See the description of this option under "Adding a DLP to a Non-SSP System" earlier in this command description.

Examples

DLP 01 DSK HSTLQG PRIORITY 92

This defines an access path to a physical disk or diskpack subsystem on channel 1. The firmware file HSTLQG will be loaded to the controller if an LH (Load Firmware into a Channel Controller or DCP) command for channel 1 is entered or if the controller is not ready. The priority of this channel is set to 92.

DLP 06 MPE PRIORITY 83

This defines a phase-encoded (PE-type) magnetic tape DLP on channel 6, with a priority of 83.

DLP 12 UNILINE USP2LH

This defines a UNILINE DLP on channel 12. The firmware file for the UNILINE DLP is USP2LH.

DLP 77 SSP DISKA SSP302 MAILBOX

This defines an SSP attached to channel 77. The SSP on this channel controls all shared peripheral devices that include the SSP named DISKA on their system configuration records. The firmware for this SSP is stored in the file SSP302. This SSP contains the intersystem communications mailbox.

See also:

- The DD (Delete DLP) command on Page 2-120.
- The LH (Load Firmware to Channel Controller/DCP) command on Page 2-218.
- The PACK (Add Pack) command on Page 2-262.
- The UNIT (Add Peripheral Devices) command on Page 2-385.
- The XC (Inhibit/Uninhibit or Display Status of a Channel) command on Page 2-434.
- The DLP and DLP SSP records in the *V Series Systems Operations Guide Volume 1: Installation*.

DM (Dump and Continue)

Keyboard Input Message

Access Level: 1

The DM command produces a memory dump of an executing task and continues execution of the task after the memory dump is complete.

Memory Dumps of a User Task

The operating system dumps the memory associated with a user task into a file on disk named \$snnnn, where *s* indicates the system executing the task and *nnnn* is a sequential number assigned by the operating system.

Once the memory is dumped, you can print and analyze it by using the PM (Print Memory Dump) command.

- <mix number> - DM _____|

Explanation

<mix number> This is the mix number of the task to be dumped.

Memory Dumps of the Operating System

The memory dump file of the operating system is always named \$s0001, where *s* is the number of the system. This file contains the contents of the system's memory at the time the command is entered.

Once the file is dumped, it can be printed and analyzed by using the PM (Print Memory Dump) command.

- 0 - DM _____|

DM is the only system command that allows a mix number of 0.

Details

The DUMP option, if set, automatically initiates an MCP dump if the operating system fails. The dump can be directed to tape, disk, or diskpack. The function of the 0 DM command depends on the DUMP option, as follows:

- If the USE DUMP record was not present when the system was cold started, 0 DM is not allowed. An error message is returned if the command is entered.
- If the medium specified on the USE DUMP record is TAPE, 0 DM is not

allowed. You must use the DMPMEM utility to dump the MCP to tape. An error message is returned if the command is entered.

- If the medium specified on the USE DUMP record is DISK or PACK, 0 DM is valid and the operating system creates a memory dump file on the indicated media.

Example

43 DM

This produces a memory dump of the task associated with mix number 43.

See also:

- The DP (Dump and Discontinue) command on Page 2-144.
- The PM (Print Memory Dump) command on Page 2-284.
- The USE DUMP record in the *V Series Systems Operations Guide Volume 1: Installation*.
- The DMPANL utility in the *V Series Systems Operations Guide Volume 3: System Utilities*.

DP (Dump and Discontinue)

Keyboard Input Message

Access Level: 1

DP specifies that a memory dump of the indicated object program is to be produced and the program discontinued. The DP command is essentially a combination of the DM (Dump and Continue) and DS (Discontinue Program) commands.

```
- <mix number> - DP _____  
                   |_____|  
                   <integer>
```

Explanation

- | | |
|--------------|--|
| <mix number> | This is the mix number of the program to be dumped and discontinued. The mix number cannot be 0 or 1, which indicates the MCP, and you cannot specify a locked program or one that is already terminating. |
| <integer> | This is entered into the MCP Run log EOJ record for analysis and can be any value from 00 through 99. Undigits are not permitted. If an <integer> is not specified, a default value of 00 is used. |

Details

Once the file is dumped, it can be printed and analyzed by using the PM (Print Memory Dump) command.

Examples

7 DP

This dumps and terminates the program associated with mix number 007.

14 DP 30

This dumps and terminates the program associated with mix number 014 and enters 30 into the Run log EOJ record.

See also:

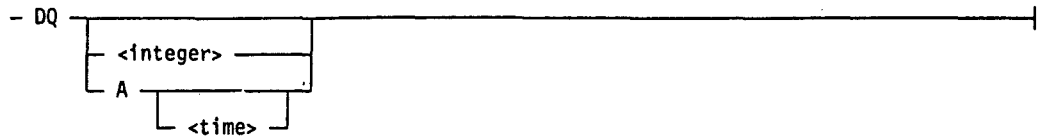
- The DM (Dump and Continue) command on Page 2-142.
- The DS (Discontinue Program) command on Page 2-148.
- The PM (Print Memory Dump) command on Page 2-284.

DQ (Display System Messages)

Keyboard Input Message

Access Level: 0

The DQ command displays system output messages from the ODT log.



Explanation

<code><integer></code>	This specifies the number of previous lines back from the most current line in the queue you want to have displayed. The <code><integer></code> can have a value from 1 to 9999, with the default value being the number of lines on the ODT screen.
A	This specifies that all messages in the ODT log should be displayed.
<code><time></code>	This specifies that all messages added to the ODT log since the specified time (hour and minute) should be displayed. The <code><time></code> is specified as <code>hhmm</code> where <code>hh</code> is the hour of the day (using 24-hour format), and <code>mm</code> is the minute within the hour.

Examples

DQ 40

This displays system messages starting 40 lines back from the most recent entry in the ODT log.

DQ A 1630

This displays all messages added to the ODT log since 4:30 pm.

See also:

- The AD (Specify ODT/OCS Screen Parameters) command on Page 2-4.
- The AJ (Display Active Jobs) command on Page 2-14.
- The ES (End Screen Display) command on Page 2-160.
- The LNS (Transfer and Print ODT Log) command on Page 2-223.

- The OL (Display Peripheral Status) command on Page 2-248.
- The TLS (Transfer ODT Log) command on Page 2-375.
- The WJ (Display Waiting Jobs) command on Page 2-411.
- The WS (Display Jobs in Schedule) command on Page 2-423.
- The WY (Display Job Status) command on Page 2-427.

DR (Date Reset)

Keyboard Input Message

Access Level: 8

DR changes the value of the current system date used by the MCP. This command is not valid if jobs are running in the mix.

- DR - <date>

Explanation

<date> This is the date to which the system date is to be set. The <date> must be in the form mm/dd/yy, and separating slashes are required. Month, day, and year values must be within valid ranges.

Example

DR 12/09/89

This changes the value of the current system date used by the MCP to December 9, 1989.

See also:

- The TR (Time Reset) command on Page 2-380.
- The WD (Display Date) command on Page 2-408.
- The WT (Display Time) command on Page 2-424.
- The DATE and TIME records in the *V Series Systems Operations Guide Volume 1: Installation*.

DS (Discontinue Program)

Keyboard Input Message

Access Level: 1

The DS command immediately terminates a program.

```
- <mix number> - DS _____  
                   |_____|  
                   <integer>
```

Explanation

<mix number>

This is the mix number of the program you want to terminate. The mix number cannot be 0 or 1, nor can it reference a locked program or one in a terminating status.

<integer>

This specifies a value to be placed in the Run log EOJ record for time analysis purposes. The <integer> can be any value from 00 through 99; undigits (A to F representing hexadecimal 10 to 15) are not permitted.

If the <integer> is not specified, a default value of 00 is used.

Details

You can use the AFTER (Hold Program) command to hold initiation of a program until the completion of another program. If you discontinue a program associated with another through the AFTER command, the following can occur:

- If the AFTR option is set, all jobs scheduled to be executed after the program being discontinued, plus all successors to these jobs, are automatically removed from the schedule.
- If the AFTR option is reset, the scheduled jobs are not disturbed.
- You can manually remove an after-linkage between programs with the RA (Remove After-Linkage) command.
- You can manually remove a job from the schedule with the RS (Remove Job from Schedule) command.

Caution

If DS is used to terminate a stopped job, and there is insufficient memory available to reload the job (in order to discontinue it), the system might be forced to stop other jobs and swap them out to disk to create enough space to reload the job. Also, the job might be marked "to be resumed" if insufficient memory is available.

Examples

101 DS

This terminates the program associated with mix number 101.

35 DS 12

This terminates the program associated with mix number 035 and specifies that the number 12 is to be placed in the RLOG EOJ record.

See also:

- The AFTER (Hold Program) command on Page 2-12.
- The DP (Dump and Discontinue) command on Page 2-144.
- The QT (Quit Program Operation) command on Page 2-308.
- The RA (Remove After-Linkage) command on Page 2-312.
- The RS (Remove Job from Schedule) command on Page 2-333.
- The ST (Suspend Program Processing) command on Page 2-364.
- The STOP (Suspend Program Processing) command on Page 2-367.
- The USE AFTR record in *V Series Systems Operations Guide Volume 1: Installation*.

DUMP (Copy Files to Library Tape)

MCP Control Instruction

Access Level: 5

The DUMP command copies disk or diskpack files to a library tape without removing the files.

It is recommended that you use the COPY command instead of the DUMP command. The DUMP command will become unavailable at a future release.

The DUMP command starts a program called LOADMP, if the files are to be copied from disk, or a program called PACKUP, if files are to be copied from diskpack. The user must supply LOADMP and PACKUP; Unisys does not supply them. If the appropriate program is not present on disk, the DUMP command is ignored.

See Appendix B for DUMP syntax if you have user-coded LOADMP and/or PACKUP utilities.

See also:

- The COPY (Invoke File Transfer Utility) command on Page 2-86.
- Appendix B

DX (Display DLPs Attached to Exchange)

Keyboard Input Message

Access Level: 8

This command lists the DLPs that are attached to an exchange.

- DX - <exchange number>

Explanation

<exchange number> This indicates the exchange for which you want to list the DLPs.

The system displays the same information about each DLP that is produced by the SHOW (Display System Information) command. For example:

Channel	Priority	State
004	90	Available
014	90	Available

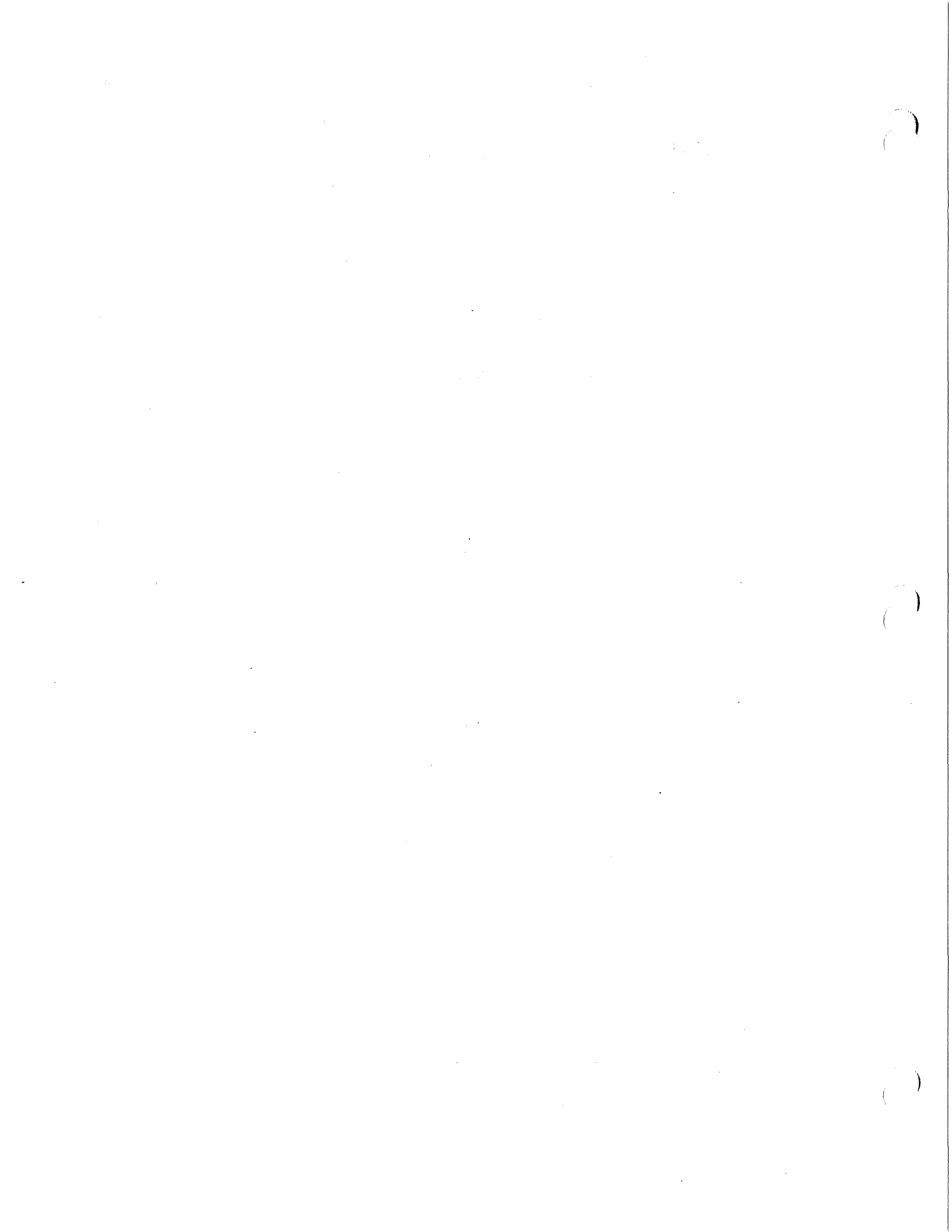
In this display, the first channel is the primary channel (that is, the exchange number). The following channels listed are alternate channels, and the State is one of the following: Available, Busy, Inhibited.

If you enter an invalid exchange number, the system responds:

****KBD IGNORED: INV EXCH**

See also:

- The OL (Display Peripheral Status) command on Page 2-248.
- The SHOW (Display System Information) command on Page 2-343.



EBCDIC (Mark Beginning of Card Data File)

MCP Control Instruction

Access Level: 0

The EBCDIC command indicates the beginning of a punched card data file.

Whenever the EBCDIC command is encountered in the card reader, it causes an End-Of-File (EOF) condition on any previous file, and the unit is marked by the MCP as containing the new file.

EBCDIC is valid only from an actual or pseudo card deck.

```

┌ EBCDIC ───┐ ┌ <file name> ───────────────────────────────────┐
└ DATA ───┘ └ CTLDCK ───┘
  
```

Explanation

EBCDIC
DATA

These are synonymous. EBCDIC or DATA signifies that subsequent input is in EBCDIC format. EBCDIC stands for Extended Binary Coded Decimal Interchange Code, a coded character set in which each character is represented by an 8-bit code.

<file identifier>

This is the name of the file that follows follow.
Any information following the <file identifier> on the same punched card is ignored.

CTLDCK

This is a reserved file identifier that allows valid control records to be included in the file. CTLDCK is the name of the physical card deck used by the LDCNTL utility to create a pseudo card deck.

Note: *If the <file identifier> is CTLDCK, no EOF action is taken until the ?ENDCTL control instruction is encountered. CTLDCK is primarily intended for use by the utility program LDCNTL for building pseudo card decks.*

Examples

```
COMPILE PROG COBOL9; EBCDIC CARD
```

This compiles the COBOL program PROG using the COBOL9 compiler. Then this statement specifies the beginning of a punched card data file called CARD. CARD is in EBCDIC format.

DATA INPUT

This specifies the beginning of a punched card data file called INPUT. The file is in EBCDIC format.

See also:

- The BCL (Mark Beginning of Card Data File) control instruction on Page 2-43.
- The BINARY (Read Binary Card Deck) control instruction on Page 2-57.
- The DATA/DATAB (Mark Beginning of Card Data File) control instruction on Page 2-105.
- The END/ENDCTL (Mark End of Card File or Control Instructions) control instruction on Page 2-156.

ED (Enter DEBUG Session)

Keyboard Input Message

Access Level: 8

The ED command allows you to go back to a specified DEBUG session from the ODT.

- ED - <session number> _____

Explanation

<session number> This is a unique 4-digit number, that is prominently displayed within each DEBUG session.

Example

ED 1387

This enables you to resume DEBUG session number 1387.

See also:

- The DEBUG (Attach Program To DEBUG Session) command on Page 2-121.
- The ID (Invoke DEBUG Session) command on Page 2-191.
- The QD (Query DEBUG Sessions) command on Page 2-307.
- The DEBUG utility in the *V Series Systems Operations Guide Volume 3: System Utilities*.

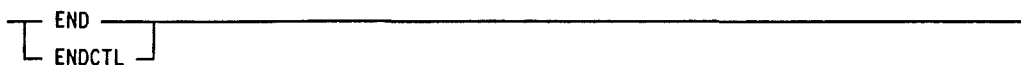
END/ENDCTL (Mark End of Card File or Control Instructions)

MCP Control Instruction

Access Level: 0

The END command indicates the end of a card data file or sequence of control instructions.

This command is valid only from an actual or pseudo card deck.



Explanation

END
ENDCTL

These indicate the End-Of-File (EOF) for a punched card file or the end of a series of MCP control instructions. The END card is always the last card of a file, a series of files, or a series of control instructions and must not be followed by any further data or control instructions for that file or program.

ENDCTL is the only command that provides EOF action on any card file having the reserved <file identifier> CTLDCK. Note that CTLDCK cannot be a BINARY file.

Details

When an END card is sensed by the LDCNTL utility, the pseudo deck (disk file) is closed, and the deck is given a unique numeric designation for operator interface. The card reader remains attached to the program LDCNTL until the special end card ?ENDCTL is reached, causing LDCNTL to terminate.

See also:

- The BCL (Mark Beginning of Card Data File) control instruction on Page 2-43.
- The BINARY (Read Binary Card Deck) control instruction on Page 2-57.
- The DATA/DATAB (Mark Beginning of Card Data File) control instruction on Page 2-105.
- The EBCDIC (Mark Beginning of Card Data File) control instruction on Page 2-153.

ENDAT (End Job Transfer)

MCP Control Instruction

Access Level: 0

The ENDAT command terminates the job transfer previously initiated by an AT (Initiate BNA Job Transfer) MCP Control Instruction.

The ENDAT command is valid only from an actual or pseudo card deck.

- ENDAT -

See also:

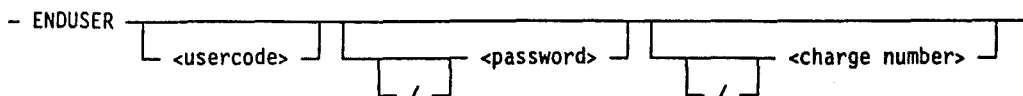
The AT (Initiate BNA Job Transfer) control instruction on Page 2-38 for a complete description of ENDAT.

ENDUSER (Mark End of Job Stream)

MCP Control Instruction

Access Level: 0

The ENDUSER command delimits the end of a job stream that has been assigned a <usercode>/<password>/<charge number> access combination by a BEGINUSER command.



Explanation

- <usercode> This refers to a code that identifies a user to the system.
- <password> This refers a group of characters by which the system determines the legality of a usercode.
- <charge number> This refers to a number used to assign charges for system use.

Note: The <usercode>, <password>, and <charge number> are all optional. Slashes are required only if leading parameters are omitted (for example, ENDUSER //30442).

Details

The <usercode>/<password>/<charge number> access combination specified by the ENDUSER command must match that of the previous BEGINUSER (Mark Beginning of Job Stream) command. If these access combinations match, then the access combination from the BEGINUSER or LI (Log In) command prior to the last BEGINUSER command becomes the active combination for the device.

BEGINUSER and ENDUSER commands can be nested. However, the access combination specified on the ENDUSER instruction must always match the access combination of the last BEGINUSER instruction.

Example

The following example shows a series of nested BEGINUSER and ENDUSER statements.

```
BEGINUSER  
USR1/PASS1/99999
```

Marks the beginning of the job stream and assigns any function requests for that job stream to usercode USR1, password PASS1, and charge number 99999.

EX PROGA	Executes a library program called PROGA.
EX PROGB	Executes a library program called PROGB.
BEGINUSER //88888	Marks the beginning of the job stream and assigns any function requests for that job stream to charge number 88888.
EX PROGD	Executes a library program called PROGD.
?SPO LNR	Sends a LNR (Transfer and Print Log) command to the MCP through a card reader to close the current system log file, directs logging information to the Run Log and starts the log analysis program.
ENDUSER //88888	Marks the end of the job stream assigned to charge number 88888.
ENDUSER USR1/PASS1/99999	Marks the end of the job stream assigned to usercode USR1, password PASS1, and charge number 99999.

See also:

- The BEGINUSER (Mark Beginning of Job Stream) MCP control instruction on Page 2-45.
- The USER (Assign Access Code) command on Page 2-401.
- The *V Series System Software Security Installation and Operations Guide* for information about <usercode>, <password>, and <charge number>.

ES (End Screen Display)

Keyboard Input Message

Access Level: 0

When the MCP response to a system command requesting information contains more data than can be displayed on the ODT or OCS at one time, the MCP displays one screen page and the "MORE DATA" message. The ES command causes the MCP to purge the information not being displayed and to return to automatic display mode.

- ES -

Details

This command is valid only from an ODT or OCS.

If you enter an ES command during normal processing, the MCP updates the automatic display of system status.

See also:

- The AD (Specify ODT/OCS Screen Parameters) command on Page 2-4.

ET (End TRAK)

Keyboard Input Message

Access Level: 8

The ET command terminates a TRAK facility tracking session.

TRAK is a real-time debugging tool that records various MCP data structures or events. TRAK entries are stored in a TRAK buffer in memory and can be analyzed by DMPANL, the dump analysis utility.

Tracking is initiated by the BT (Begin TRAK) command.

- ET -

Details

After using the ET command, the TRAK buffer can be printed and analyzed by first creating a dump file through a *0 DM* command followed by a *PM 1 TRAK* command. See DM (Dump and Continue) and PM (Print Memory Dump) in this section, for more information.

The TRAK option can be reset with the RO (Reset System Options) command.

See also:

- The BT (Begin TRAK) command on Page 2-59.
- The DM (Dump and Continue) command on Page 2-142.
- The PM (Print Memory Dump) command on Page 2-284.
- The RO (Reset System Options) command on Page 2-327.
- The SO (Set System Options) command on Page 2-352.
- The DMPANL utility in the *V Series Systems Operations Guide Volume 3: System Utilities*.

EXECUTE (Execute Library Program)

MCP Control Instruction

Access Level: 1

The EXECUTE command is used to execute a library program on disk or diskpack.

```

- EXECUTE [ /MCP ] [ /OWN ] <program name> [ ON <medium ID> ] [ <parameter list> ]

```

Explanation

EXECUTE	These are synonymous.		
EX			
/MCP	This specifies that the bound MCP intrinsic of the name specified is the program to be executed. A user-coded intrinsic of the same name is not executed. If you specify this option, the ON <medium ID> option (if included) must specify DISK or not be used.		
/OWN	This specifies that only the user-coded intrinsic with the correct name can be used. If the /MCP or /OWN option is omitted, the user-coded intrinsic will be used if present on disk. If not, then the MCP intrinsic will be executed.		
<program name>	This is the name of the program to be executed. It must be a permanent file on disk or on the named diskpack (or diskpack family if more than one diskpack is present with the appropriate name).		
<medium ID>	This specifies to the MCP the location of the program. The <medium ID> can be one of the following: <table> <tbody> <tr> <td>DISK</td> <td>This directs the MCP to look for the program to be executed only on 100-byte media. If the program is not found, an appropriate message will be displayed by the MCP.</td> </tr> </tbody> </table>	DISK	This directs the MCP to look for the program to be executed only on 100-byte media. If the program is not found, an appropriate message will be displayed by the MCP.
DISK	This directs the MCP to look for the program to be executed only on 100-byte media. If the program is not found, an appropriate message will be displayed by the MCP.		

- PACK** This directs the MCP to search all of the unrestricted diskpacks on the system for the program to be executed. If the program is not found, an appropriate message will be displayed. The program must be in 180-byte code file format. A 180-byte format code file is created when a code file is copied from disk to diskpack using the PCOPY utility, or when the generating compiler is directed to create such a code file. See **COMPILE** (Compile Program), in this section.
- A diskpack family name** This directs the MCP to search for the program to be executed on that diskpack family name only. If the program is not found, an error message will be displayed.

If no <medium ID> is specified, the MCP looks first on disk, then on unrestricted diskpack.

If you don't provide a <medium ID>, the MCP first searches all 100-byte disks for the program. If not found in the disk directory, a search of all unrestricted diskpacks is performed. If the program is not found in the disk directory or in any of the unrestricted diskpacks, an appropriate message is displayed.

You can reduce the extensive search that might result from not using the ON <medium ID> clause by including a CONTROL CODEPACK record in the system configuration file. Including the CONTROL CODEPACK record forces the MCP to look for the program on the specified diskpack only, if the program was not found on disk.

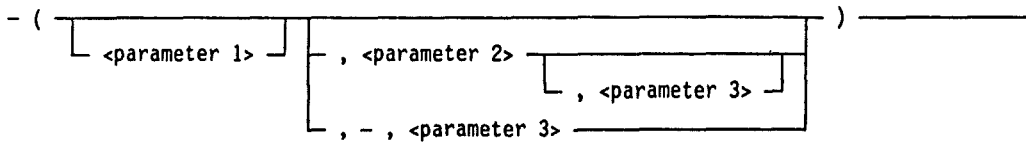
<parameter list>

These are up to three parameters that can be inserted into a program when the program is started.

The <parameter list> option appears in the syntax of the **COMPILE** (Compile Program), **DEBUG** (Attach Program to Session), and **EXECUTE** (Execute Library Program) commands. These parameters allow you to enter Boolean, string, and integer values, which can be used by the program when the program is being compiled or executed.

The syntax for <parameter list> is shown in the following figure.

EXECUTE



The rules for <parameter list> resemble those for parameters in the Work Flow Language (WFL) RUN command.

Integer parameters

- Contain unsigned numeric data up to eight digits in length (8 UN)
- Have values ranging from 0 through F
- Are right-justified and zero-filled in the high-order digits if less than the maximum length
- Will cause a syntax error if greater than the maximum length

String parameters

- Contain alphanumeric data up to six characters in length (6 UA)
- Must be within quotation marks. To pass a quoted string as a parameter, a double quote must appear in the parameter string (for example, EXECUTE MYPROG ON PACK (,'TRY' 'XX')).
- Can be composed of any EBCDIC characters
- Are left-justified and blank-filled in the low-order bytes if less than the maximum length
- Will be truncated in the low-order bytes if greater than the maximum length

<parameter 1>

This can be an integer up to eight digits in length. It sets the switches at program address 0. For example, EX MYPROG (10100000).

<parameter 2>
<parameter 3>

This can be 8-digit integers or 6-character strings. The value of <parameter 2> is inserted into the code file at base-relative address 8. The value of <parameter 3> at base-relative address 20. For example, EXECUTE /OWN MYPROG (,12345678) or EXECUTE /OWN MYPROG (101,'TOTEST').

When either <parameter 2> or <parameter 3> are specified, you must have either a value or at least a comma in the preceding parameters. For example, EXECUTE MYPROG /OWN (,,'TAPE').

A null entry is allowed. For example, EXECUTE MYPROG

/OWN (,0,'ABC'''' '') does not change the switches and inserts 00000000 at address 8 and the string 'ABC'''' at address 20.

Note: Slash parameters were available on previous releases of the MCP. However, slash parameters will be deimplemented in a future release of the MCP. The <parameter list> option replaces them.

Examples

EX BIOG ON PACK; PR 5

This executes the library program BIOG, which resides on PACK, and then assigns the program a setting of 5 for the memory priority, processing and I/O priority, and schedule priority.

EXECUTE /MCP AUDIT (,'TEST10')

This executes the library program AUDIT, an MCP intrinsic that resides on disk, and inserts "TEST10" at base-relative address 8. This command does not set any switches.

EX MDCOPV (0100000)

This executes the MDCOPV utility, which provides maintenance for flexible disks. Starting at memory location 1, this command sets switch number 1 to 1, and switch numbers 2 through 6 to 0.

See also:

- The COMPILE (Compile Program) command on Page 2-80.
- The DEBUG (Attach Program to DEBUG Session) command on Page 2-121.
- The CONTROL CODEPACK record in the *V Series Systems Operations Guide Volume 1: Installation*.
- The PCOPY utility in the *V Series Systems Operations Guide Volume 3: System Utilities*.

FI (List Files in Use by a Program)

Keyboard Input Message

Access Level: 0

The FI command produces a list of all files in use by a program.

- <mix number> - FI _____

Explanation

<mix number> This is the mix number of the program whose files you want to list.

Example

38 FI

This lists all files in use by the program associated with mix number 038. A sample response to this input is as follows:

```
FILES IN USE BY @20298/PBDPRN = 038
File      Hdw   cc/uu   I/O    I/O Count
@20298    DSK   15/01   I       123
SHORT/DASC10 PRN   13/00   0      1254
```

As many response lines as necessary will be produced.

See also:

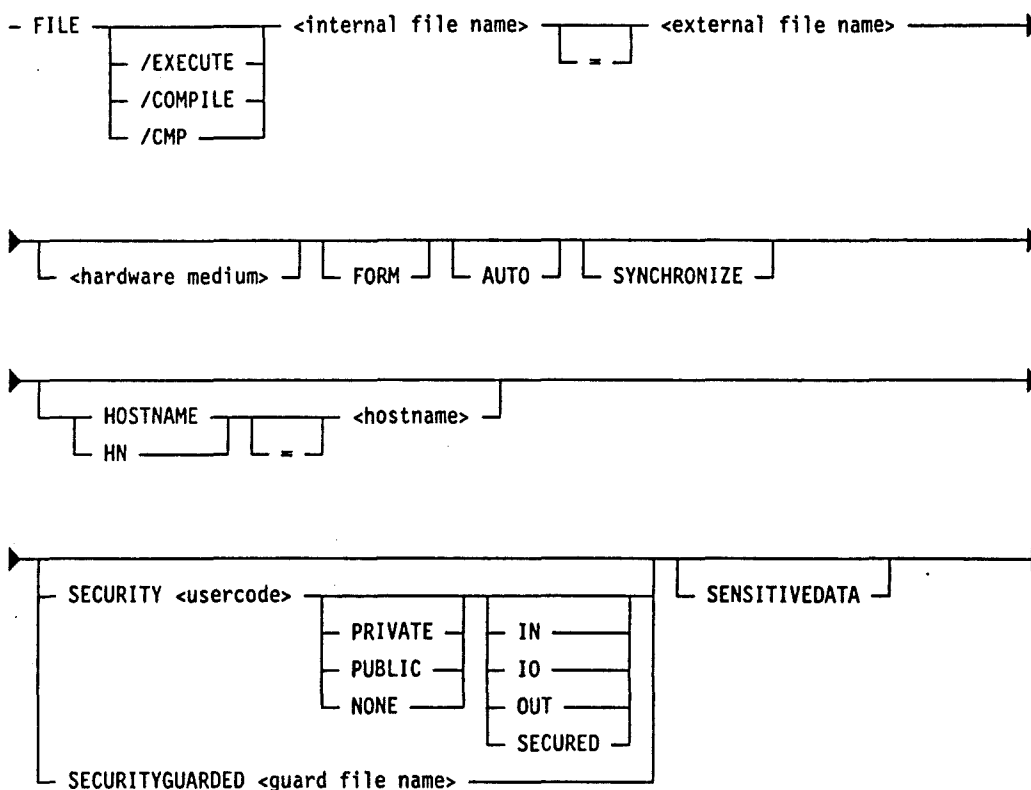
The BF, BFP (List Backup Files on Disk or Pack) command on Page 2-47.

FILE (Equate File-Name with File-Identifier or Medium)

MCP Control Instruction

Access Level: 3

The FILE command, an MCP label equation feature, allows a change at execution of the file identifier and, in some cases, of the hardware medium of a specified file.



The FILE command equates an <internal file name> as used in a given source program with a specified <external file name>, and optionally with a <hardware medium>. The new file name or <hardware medium> will be used at execution time. A PORT file cannot be specified in a FILE statement.

Explanation

/EXECUTE
/COMPILE
/CMP

This can be used to distinguish between the COMPILE and subsequent EXECUTE phases of a compile-and-go operation. If this option is omitted on a compile-and-go operation, the FILE command is

associated with the compiler. This option can be used only on compile-and-go operations.

/COMPILE and **/CMP** are synonymous.

<internal file name> This is the file name, as it is known to the program, of the file to be label-equated. If the **<internal file name>** consists of more than six characters, only the first six characters can appear in the **FILE** command. If the **<internal file name>** contains special characters (for example, **PRNT-F**), the **<internal file name>** must be enclosed in parentheses, for example, **(PRNT-F)**.

<external file name> This is the file name to be used instead of the **<internal file name>** specified in the source program. For media permitting a **<multifile ID>**, a **<multifile ID>/ <file ID>** entry can be used.

<hardware medium> This designates an alternate **<hardware medium>**, which can be any of those listed in the following table.

Mnemonic	Hardware Device
CRD	80-column Card Reader
DCP	Data Communications Processor
DISK	Head-per-track disk (100-byte media)
DPK	Pack (180-byte media)
DSK	Head-per-track disk (100-byte media)
GCR	GCR Magnetic Tape Drive
MPE	Phase-Encoded (PE) Magnetic Tape Drive
MTP	Magnetic Tape (any)
MT9	Non-return to Zero Magnetic Tape Drive
PACK	Pack (180-byte media)
PBD	Printer backup disk
PBK	Printer backup disk, pack, or tape
PBP	Printer backup pack
PBT	Printer backup tape
PCD	Card punch backup disk
PCH	80-column Card Punch
PCO	Card punch only
PCP	Card punch backup pack

Mnemonic	Hardware Device
PRN	Buffered Printer
PRO	Line Printer only
QWIK	QWIK disk

Table 2-5 shows the allowable file equations across hardware types:

Table 2-5. Allowable File Equations Across Hardware Types

TO → FROM↓	DSK	DPK	MTP	PRN	CRD	PCH	QWK
DSK		YES	NO	NO	YES	NO	
DPK	YES		YES	YES	YES	YES	YES
MTP	YES	YES		YES	YES	YES	NO
PRN	YES	YES	YES		NO	YES	NO
CRD	YES	YES	YES	NO		NO	YES
PCH	YES	YES	YES	YES	NO		NO
QWK	YES	NO	NO	YES	NO		

The hardware usage must be compatible; for example, output tape cannot become CRD.

DISK output cannot be file-equated to anything except diskpack. DISK input can be file-equated to CRD or diskpack.

DCOM files cannot be file-equated to any other hardware devices, including other DCOM devices.

FORM

When following any of the printer or punch options, this indicates that special forms are required.

AUTO

This specifies that the APBD (Autoprint PBD files) option has been set, and that the file is to be printed as soon as it has been closed. If the USE PBD option is set, and the APBD option is set, AUTO can be used on the FILE record to automatically schedule the printing of the file.

SYNCHRONIZE

This indicates that data is transferred between host and tape drive in an unbuffered mode. The default mode is asynchronous (buffered), which maximizes tape drive performance.

FILE

HOSTNAME HN	These are synonymous, and cause the indicated file to be opened on the processor for which this is the hostname. This requires that both the originating processor and the remote processor be included in a BNA network.
SECURITY	This specifies that security attributes are going to be applied to this file.
PRIVATE	This specifies that the file can be accessed only by the creator of the file in the manner specified by the SECURITYUSE attribute. (For a description of the SECURITYUSE attribute, see the <i>V Series MCP/VS Programming Reference Manual</i> .) PRIVATE is the default SECURITYTYPE attribute.
NONE	This specifies that no security attributes are to be applied to this file.
PUBLIC	This specifies that the file can be accessed by all users in the manner specified by the SECURITYUSE attribute.
IO	This specifies that the SECURITYUSE attribute will allow READ/WRITE access to the file.
IN	This specifies that only READ access is allowed to the file.
OUT	This specifies that only WRITE access is allowed to the file.
SECURED	This specifies that the file is a program and can be executed only if it passes the MCP requirements for a program. It cannot be opened as a file by a user program.
SENSITIVEDATA	This specifies that all areas of the file are to be overwritten with random data before the areas are returned to the available disk pool. SENSITIVEDATA will be ignored if the file is declared as PUBLIC.
SECURITY GUARDED	This specifies that another file (guard file name) must be accessed before a user can be given permission to use the file. The SECURITYUSE attribute for a guarded file is stored in the GUARD file.

Examples

```
EXECUTE A; FILE CARDFL = CARDX
```

This executes library program A; causes the MCP to equate the internal file name CARDFL with the external file name CARDX.

```
EXECUTE B; FILE PRINTR = SLPRNT PBD FORM
```

This equates the file PRINTR with a printer backup disk file called SLPRNT, requiring special forms.

```
COMPILE C COBOL9; FILE CARD X DSK; FILE/EX TAP TP1/TP
```

This contains two file-equate statements. The second statement indicates the EXECUTE phase of a compile-and-go operation, and equates the file name TAP with the file name TP1/TP.

Automatic Printer Assignment

Through the use of the FILE statement or by setting the FORMs flag for the print file, the MCP allows automatic routing of printer files to a specific printer. The automatic routing feature should be applied when the normal state of the printer is saved. A printer can be kept in a saved state through use of the SV (Save a Peripheral Unit). Print files can be routed to the saved printer using the FM (Direct File to Device—Special Forms Response) command, or by using the FORM option of the FILE command to maintain the SAVED status after use.

Automatic routing applies to train printers only and is set up by the UNIT record for the printer. The UNIT record syntax for a printer is shown in the following figure.

```
- UNIT - <cc/u> - <unit ID>   _   <hardware mnemonic> _____
```

Explanation

<cc/u>

This is the channel and unit number for the printer.

<unit ID>

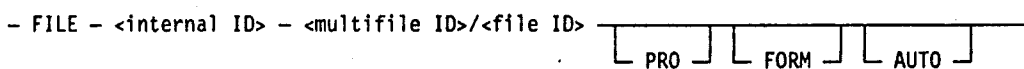
This is the default train ID (translate table) to be used for a train printer.

The <unit ID> field is used for printer identification when implementing automatic routing. The <unit ID> is an identifier (one to six digits) used to make a specific printer unique. When a print file that has FORMs specified is to be printed, the MCP looks at the multifile ID and assigns the print file to the printer whose first four characters of <unit ID> are the same as the first four characters of the print file's

multifile ID. For example, a print file with a multifile ID of SHORXZ would be assigned to a printer with a <unit ID> of SHORT.

- <hardware mnemonic> This is the hardware type. The following are valid.
- PRN This specifies that the unit is a line printer.
 - TPR This specifies that the unit is a train printer.

The syntax for the FILE command for automatic printer assignment is shown in the following figure.



Explanation

- <internal ID> This is the name of the print file to be printed. The internal ID of the print file for PBDPRN is PRINT.
- <multifile ID> This is the name of the file whose first four characters are the same as the first four characters of the <unit ID> for the printer.
- <file ID> This is the external ID of the print file.
- PRO This directs the file to the printer only, not to backup, and can be used when file equating with a PRINT or PBD command. PRO must be specified before FORM.
- FORM This assigns special forms to a print file that did not have forms assigned programmatically. FORM must be specified for automatic printer assignment. It will leave the printer in a SAVED state.
- AUTO This specifies that the APBD (Autoprint PBD files) option has been set, and that the file is to be printed as soon as it has been closed. If the USE PBD option is set, and the APBD option is set, AUTO can be used on the FILE record to automatically schedule the printing of the file.

Use the free-standing LOADER program to create the train printer translate tables to provide a name that corresponds to <unit ID> on the UNIT record. You could also copy an existing file and rename it.

The LOADER program must be run for each form that is to be maintained on a saved printer.

Note: If all printers on a system have the same train (96-character), LOADER has to be run once. If the printers are going to have the same form loaded, only the single translate table needs to be maintained. If the printers are going to have different forms, DMPALL or SYSTEM/COPY could be used to duplicate the translate table and change the name.

If multiple printers have the same <unit ID>, the MCP will assign the print file to the printer that is declared first.

Firmware is loaded onto an image printer by using the LDHOST utility.

Multiple printers can use bound train tables simultaneously.

The automatic routing can be used in an environment where forms are changed frequently. To do so, perform the following:

1. Declare the printer without a train ID and load FORMs.
2. Leave the printer not-ready.
3. Transmit the following:

RY <cc/u> <file ID>; SV <cc/u>

Manually readying the printer will save and lose FORM ID at end of run.

You could also use the FM (Direct File to Device—Special Forms Response) command and indicate a new file ID.

If the printer is a drum printer, the execution of the LOADER program is unnecessary. Only a <unit ID> needs to be put on the UNIT record.

Note: Because FORM and MFID are required, the printer must be saved only at halt/load but not made ready with an RY (Ready Peripheral Device) command. A form request will be direct to a readied form printer with a matching UNIT ID. However, other NO FORM prints will use it too.

File equation is not necessary if FORM and MFID are specified in the program.

File equation for a program should normally specify PBD/PBP/PBK. When the file is created in this manner, a normal PRINT command (without a file-equating command) can be used.

FILE Statement Examples

The following printer configurations are used in the examples:

UNIT 07/0 COMPLE PRN SAVED	standard drum printer
UNIT 32/0 LONGFM TPR SAVED	96 character train
UNIT 62/0 SHORTY TPR SAVED	96 character train

Example 1

```
COMPILE LGAD01 COBOL9 LIBRARY
FILE LINE COMPLE/LGAD01 FORM
DATA CARD
END
```

This example will change the file ID of this print file from LINE to COMPLE/LGAD01 and set the FORMs option for the file. When the file is printed, it will automatically go to printer 7/0. If AUTO had been specified and the APBD option were on, the file would have printed automatically after it was created.

Example 2

```
EXECUTE LGAD01
FILE LOGRPT LONGFM/LGAD01 FORM AUTO
END
```

This example assigns a name of LONGFM/LGAD01 to the print file LOGRPT and sets the FORMs option for the file. AUTO specifies that the file be printed automatically on printer 32/0 after it is created.

Note: If the APBD option is not set, the file will not be printed automatically. A PRINT command will have to be issued to print the file.

See also:

- The FM (Direct File To Device—Special Forms Response) command on Page 2-175.
- The RY (Ready Peripheral Device) command on Page 2-338.
- The SECURITYUSE attribute in the *V Series MCP/VS Programming Reference Manual*.
- The UNIT and USE APBD records in the *V Series Systems Operations Guide Volume 1: Installation*.

FM (Direct File to Device—Special Forms Response)

Keyboard Input Message

Access Level: 7

The FM command lets you reply to the following MCP message:

```
<hardware mnemonic> REQD <file name> <job name> = <mix number>
```

This message indicates that the job identified by <mix number> requires a particular output device.

The FM command is identical to the OU (Direct File to Device) command except that the device specified by the <cc/u> can be in a saved status, and the device is left in a saved status when the output process is completed.

```
- <mix number> - FM _____
                    |_____| |_____| |_____|
                    <cc/u> <file identifier> [ DK ]
                                                [ PK ]
```

Explanation

<mix number>	This must specify the mix number of a program that requires an output printer or punch device.
<cc/u>	This must designate the unit to be used for the subject file and must be a valid peripheral type for the file in question.
<file identifier>	This can be used to specify which translate file is to be loaded into the control, when the <cc/u> refers to a train printer.
DK PK	This specifies that the print file should be directed to disk backup (DK) or diskpack backup (PK), respectively.

Examples

```
2 FM 8/0
```

This directs the output of a program with a mix number of 2 to the output device with channel number 8 and unit number 0.

```
12 FM DK
```

This directs the output of a program with a mix number of 12 to a disk printer backup or file.

10 FM 16/0 A123

This directs the output of a program with a mix number of 10 to a train printer with channel number 16 and unit number 0. Also, the system is directed to use a printer translation file named A123.

See also:

The OU (Direct File to Device) command on Page 2-257.

FN (List File Names)

Keyboard Input Message

Access Level: 3

The FN command produces a list of the internal file names and the corresponding external file names of a program. This enables the operator or programmer to determine the appropriate file name to use in a file-equation statement.

```
- FN [ /MCP ] <program name> [ ON <family name> ]
     [ <multifile ID>/<file ID> ]
```

Explanation

/MCP	This indicates that the program is bound to the MCP. This option is invalid if the ON <pack ID> syntax is used.
<program name>	This is the name of the program whose file names you want to list.
ON <family name>	This indicates that the program resides on a diskpack family that has a label specified by <family name>.
<multifile ID>/<file ID>	This specifies the external ID of the file as displayed on the ODT if a NO FILE condition existed. The syntax is identical to that of a FILE statement when label equating a tape or diskpack file. The <file ID> can be masked with a single equal (=) sign. Masking is not permitted in the <multifile ID>.

Examples

```
FN COPY
```

This lists the internal and external file names of the program COPY.

```
FN/MCP COPY
```

This lists the internal and external file names of the MCP-bound program COPY.

See also:

The FILE (Equate File Name with File Identifier or Medium) command on Page 2-167.

FP (Flush Program)

Keyboard Input Message

Access Level: 1

The FP command directs the MCP to purge all I/O requests for a terminating program when peripheral hardware malfunctions are inhibiting the termination. The program normally awaits the completion of processing on an output file after a DP (Dump and Discontinue), DS (Discontinue Program), or STOP (Suspend Program Processing) command.

- <mix number> - FP

Explanation

<mix number> This is the mix number of a terminating program and cannot be 0 or 1.

Caution

Use this command cautiously because all I/O operations pending for the specified mix number at the time of the request are lost when FP is used.

Example

7 FP

This purges all I/O requests for the terminating program associated with mix number 007.

See also:

- The DP (Dump and Discontinue) command on Page 2-144.
- The DS (Discontinue Program) command on Page 2-148.
- The ST (Suspend Program Processing) command on Page 2-364.

FR (Designate Final Reel)

Keyboard Input Message

Access Level: 1

The FR command lets you respond to a magnetic-tape or paper-tape file-required message from the MCP by indicating the input reel just read as the final reel of the file. This command is used primarily for unlabeled files, the end of which cannot be recognized by the MCP.

- <mix number> - FR _____

Explanation

<mix number> This represents the mix number associated with the job that produced the magnetic tape or paper tape file required message.

Details

When a program declares a tape to be labeled, you can use the FR command if you use the UL (Assign Unlabeled File) command to designate the unit on which a particular unlabeled tape file is located. However, you cannot use FR if you use the IL (Assign Labeled File) command, which designates the location of a labeled file.

Example

13 FR

This designates the input reel just read for the file associated with mix number 013 as the final reel of the file.

See also:

- The IL (Assign Labeled File) command on Page 2-192.
- The UL (Assign Unlabeled File) command on Page 2-383.

GENERATE (Schedule DMPALL for Execution)

MCP Control Instruction

Access Level: 1

The GENERATE command schedules the DMPALL program for execution, passing specified parameters to that program through a disk file.

The GENERATE function of DMPALL generates an Assembler language program that reproduces a data file from one hardware type to another hardware type, and optionally changes file ID, record length, blocking factor, or parity.

```
- GENERATE [ /MCP ] [ /OWN ] <specification string>
```

Explanation

GENERATE
GEN

These are synonymous.

/MCP
/OWN

This specifies that either the bound MCP-intrinsic DMPALL (/MCP) or a user-provided DMPALL (/OWN) should be used.

If /MCP is specified, the bound, MCP-intrinsic DMPALL is scheduled regardless of the presence on disk of a user-provided DMPALL.

The /OWN option can be used to prevent the scheduling of the bound, MCP-intrinsic DMPALL when a user-provided DMPALL is required but is not on disk.

If this option is omitted, the GENERATE command causes the MCP to search the disk directory for a user-provided program called DMPALL. If a user-provided DMPALL is resident on disk, it is scheduled for execution; otherwise, the MCP intrinsic DMPALL program is scheduled to generate the routine described in the <specification string>.

<specification string>

The available options and their methods for inclusion in the <specification string> are defined under "DMPALL-File Conversion Utility Program" in the *V Series Systems Operations Guide Volume 3: System Utilities*.

When DMPALL is executed, the semicolon delimiter must be used following the <specification string> when <additional control instructions> such as MEMORY or PRIORITY are desired.

Examples

GENERATE /OWN CRDCRD A B; AFTER VPROG1; CHARGE 99

This executes a user-provided version of DMPALL to generate a program to copy a card file named A to a punch file named B. This function will be performed after PROG1 finishes executing. A charge number of 99 will be inserted into the MCP Run log for these programs.

GEN LIST TAPFIL 200 100; MEM 45

This executes the MCP version of DMPALL to generate a program that will provide a double-spaced listing of the tape file named TAPFIL. This MEM command specifies that this task will be allocated 45 kilodigits of memory to execute.

See also:

- The PERFORM (Schedule DMPALL) command on Page 2-280.
- The DMPALL utility in the *V Series Systems Operations Guide Volume 3: System Utilities*.

GO (Restart a Stopped Program)

Keyboard Input Message

Access Level: 3

The GO command requests the resumption of a program stopped by an ST (Suspend Program Processing) or STOP (Suspend Program Processing) command.

The GO command has the same effect as the START (Restart a Stopped Program) command.

- <mix number> - GO _____|

Explanation

<mix number> This is the mix number of the program to be restarted.

Details

The program restarts its execution at the point where it stopped. It starts with the same values it was using when it stopped.

If you issue GO on a program that is not stopped, the system ignores the command.

If there is not enough memory for the program being restarted the following occur:

- Memory is allocated by priority. If you raise the program's memory priority, it might restart.
- When sufficient memory becomes available, the program will restart.
- A program being restarted will start before a program of the same priority that is waiting in the schedule.
- Programs with lower memory priority are rolled out from memory to disk if this will make enough room for the program that is waiting to restart.

Example

```
119 GO
```

This restarts a program with mix number 119.

See also:

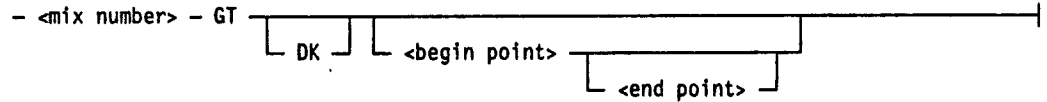
- The ST (Suspend Program Processing) command on Page 2-364.
- The START (Restart a Stopped Program) command on Page 2-365.
- The STOP (Suspend Program Processing) command on Page 2-367.

GT (Initiate Trace)

Keyboard Input Message

Access Level: 3

The GT command initiates the instruction trace for a specified program that is executing.



Explanation

- <mix number>** This designates the mix number of the program to be traced and cannot be 0 or 1.
- DK** This specifies that the trace data should be directed to a printer backup disk file.
- DK has the same effect as using the OU (Direct File To Device) command to send the file to disk or as setting the GTDK system option. The GTDK option can be set through the USE GTDK record in the system configuration file or with an SO (Set System Options) command.
- If this option is omitted, the trace goes to the system default medium, determined by the setting of the GTDK system option.
- <begin point>**
<end point> These allow the operator to specify beginning and ending limits for the trace printing.
- If the **<begin point>** is specified, the trace begins printing when the program reaches the designated point.
- If the **<end point>** is also specified, the printing terminates at that point and does not resume until the **<begin point>** is again reached. The program is still being traced during this period, but the printing is suppressed.
- If you omit **<begin point>** and **<end point>**, the entire program can be traced. Tracing will begin where the program was executing when the GT was entered. This will trace user code only.
- The figure that follows shows the format for the **<begin point>** and **<end point>** specifications.

`<segment number> / <instruction address>`

`<segment number> /
<instruction address>`

These specify the exact segment number and address of an instruction. If either is 0, the corresponding limit is not enabled. A `<segment number>` of 0 means any segment, not just segment 0.

Details

The trace is listed on the first available line printer; if none is available, any printer in use by the designated program is used. If printers are not available, the program is suspended until a printer becomes available or until the operator directs the trace to backup disk through an OU (Direct File to Device) command.

You can use the TRACE (Initiate Trace) command to trace a program when it enters the mix.

The NT (Terminate Trace) command stops the tracing of a program.

Traces can also be started programmatically. The TRACE program branch communicate (BCT) requests that the MCP initiate or terminate a program trace or produce a memory dump of the program. See the *V Series Program Interfaces Programming Reference Manual* for more information.

The interactive DEBUG module has a number of sophisticated debugging features that can be used to debug user programs. DEBUG (Attach Program to a DEBUG Session) and ID (Invoke DEBUG Session) can be used to start DEBUG sessions. GT and TRACE can be used only for user programs.

Examples

```
21 GT DK
```

This starts an instruction trace on the program with mix number 021 and sends the trace data to a printer backup file on disk.

```
14 GT 13/1484 13/14620
```

This starts an instruction trace on the program with mix number 014. The trace should begin at segment 13, address 1484 and end at segment 13, address 14620.

See also:

- The DEBUG (Attach Program to DEBUG Session) command on Page 2-121.
- The ID (Invoke DEBUG Session) command on Page 2-191.
- The NT (Terminate Trace) command on Page 2-243.
- The OU (Direct File to Device) command on Page 2-257.

- The SO (Set System Options) command on Page 2-352.
- The TRACE (Initiate Trace) command on Page 2-381.
- The USE GTDK record in the *V Series Systems Operations Guide Volume 1: Installation*.
- The *V Series Program Interfaces Programming Reference Manual* and the various program language manuals for information about the TRACE command.

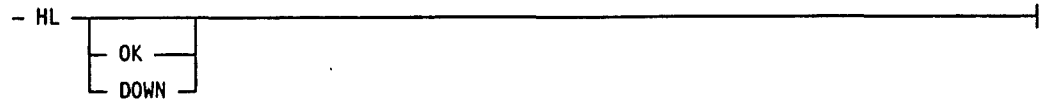


HL (Request a Halt/Load)

Keyboard Input Message

Access Level: 8

The HL command causes a halt/load or halt operation to be performed. You can use HL only at the HLSPO ODT.



Explanation

- | | |
|----|--|
| HL | This initiates a halt/load operation. HL is valid only if there are no jobs running in the mix. |
| OK | This must be entered if jobs are running in the mix and you want to perform a halt/load operation. |

Caution

The HL OK command terminates all jobs and related activities on the system.

- | | |
|------|--|
| DOWN | This causes just the halt portion of a halt/load operation. HL DOWN is valid only if there are no jobs running in the mix. |
|------|--|

Details

A halt/load operation can do the following:

- Load the MCP from the system disk into memory
- Build the tables and lists in memory used by the operating system

A halt/load is usually used to start the machine and to recover from failures.

The HLSPO is an Operator Display Terminal (ODT) that displays halt/load messages and fail codes in the event of system failure. The HLSPO is declared in the UNIT ODT record of the system configuration file or through the UNIT (Add Peripheral Devices) command.

If you enter HL or HL DOWN and there are jobs running in the mix, the message stating that a null mix is required is displayed.

Examples

HL

If there are no jobs running in the mix, this causes a halt/load operation.

HL OK

This causes a halt/load operation regardless of whether jobs are running in the mix.

HL DOWN

This causes the halt portion of a halt/load operation.

See also:

- The UNIT (Add Peripheral Devices) command on Page 2-385.
- "System Initialization Coldstart and Haltload Instructions" and the "UNIT ODT, OCS Record" in the *V Series Systems Operations Guide Volume 1: Installation*.

HN (Identify or Change Hostname)

Keyboard Input Message

Access Level: 0

The HN command can be used to identify and change the hostname of the local processor that supports BNA network architecture.

The hostname is originally supplied during system initialization through the HOSTNAME record of the system configuration file.

```
- HN _____
    |_____|
    |<hostname>|
```

Explanation

HN	This can be used without the <hostname> option to display the current host name.
<hostname>	This identifies the host and can be up to 17 characters long as follows: <ul style="list-style-type: none"> • A number followed by up to 16 additional numbers • A letter followed by up to 16 characters, which can be any combination of numbers, uppercase letters and lowercase letters

You cannot change the hostname if there are running entries in the mix. You must have a null mix. However, you can display the name of the current host at any time.

Details

The hostname is used to identify the processor in the Run log, the ODT log, and the Maintenance log.

Examples

The following examples contain a single line as entered at the ODT followed by the one-line response from the system. The original hostname is RED.

Operator Input	System Response
HN	HN = RED
HN GREEN	HOSTNAME CHANGED FROM RED TO GREEN
HN 123	HOSTNAME CHANGED FROM GREEN TO 123
HN BNAh1	HOSTNAME CHANGED FROM 123 TO BNAh1

Invalid examples

HN 1HOST

This is invalid because you cannot have a hostname that starts with a number and is followed by letters. An invalid syntax message will be displayed.

HN HOST-1

This is invalid because you cannot have a special character such as a hyphen as part of a hostname. An invalid syntax message will be displayed.

See also:

- The HOSTNAME record in *V Series Systems Operations Guide Volume 1: Installation*.
- The *V Series BNA Installation and Operations Guide*.
- The *V Series MCP/VS Programming Reference Manual*.

ID (Invoke DEBUG Session)

Keyboard Input Message

Access Level: 8

The ID command invokes a DEBUG session at the ODT on which it is entered.

- <mix number> - ID _____

Explanation

<mix number> This refers to the mix number of an active job in the mix.

Related Commands

ID differs from the DEBUG command. The ID command invokes a DEBUG session for a program that is already running on the system. The DEBUG command *executes a program* and then attaches the program to an interactive DEBUG session.

The ED and QD commands, like the ID command, manipulate DEBUG sessions. The ED (Enter Debug Session) command allows you to return to a DEBUG session previously started through the DEBUG or ID command. The QD (Query Debug Sessions) command displays the number of active DEBUG sessions on the system with their corresponding session numbers.

Do not confuse the commands used with the interactive DEBUG facility (DEBUG, ED, ID and QD) with the commands used as part of the existing program trace facility (TRACE, GT, or ET).

Example

55 ID

This invokes a DEBUG session for the program associated with mix number 055.

See also:

- The DEBUG (Attach a Program to a DEBUG Session) command on Page 2-121.
- The ED (Enter DEBUG Session) command on Page 2-155.
- The QD (Query DEBUG Sessions) command on Page 2-307.
- The DEBUG utility in the *V Series Systems Operations Guide Volume 3: System Utilities*.

IL (Assign Labeled File)

Keyboard Input Message

Access Level: 5

The IL command tells the MCP the location or name of a labeled file that is needed by a program.

IL lets you designate the location or name of a file to be given to a requesting program. IL is often used to respond to the following system messages:

For Non-Disk Files

- File-required message, such as one of the following:

<file ID> NOT AVAILABLE

NO FILE ON UNIT

- Duplicate-file message, such as the following:

DUP FILE

For Disk And Diskpack Files

- File not in directory message, such as one of the following:

<file ID> NOT ON DISK

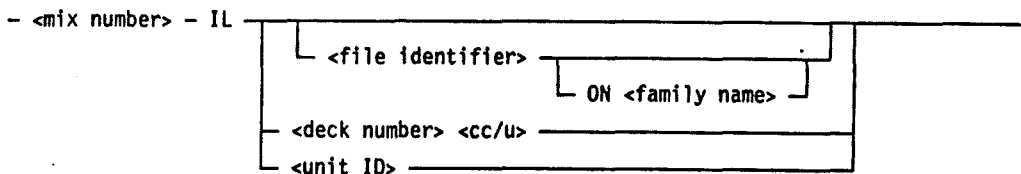
<file ID> NOT ON DPK

- Duplicate file on pack message, such as the following:

DUP FILE ON DPK

The file designated in the IL command is assumed by the MCP to be the file required by the requesting program regardless of the name, reel number, and, within limits, the hardware type.

For non-disk files, use the syntax shown in the following figure.



Explanation

<mix number>	This is the mix number of the program requesting the file.
<file identifier>	This can be included to specify the particular file desired if the indicated unit contains a multifile tape. File identifiers can be up to six characters long. They cannot be made up of all nulls, all blanks, or all zeros. If a <file identifier> is not present for a multifile tape-open, the first file on the tape will be assigned, regardless of its name.
ON <pack name>	This indicates that the file resides on a particular diskpack.
<deck number> <cc/u>	This can be specified if the file to be assigned is present in an active pseudo reader. The <cc/u> indicates the required diskpack.
<unit ID>	This is a 1- to 6-character name that can be used for data communications on MICR files to specify the desired device. The file ID must match the UNIT-ID declared on a UNIT record in the system configuration file.

For disk or diskpack files, use the IL command syntax shown in the following figure. Note that this syntax can be used only in response to a file-not-in-directory or duplicate-file-on-pack message. This syntax can also be used for diskpack files if the diskpack family name is correct but a different file on the family is to be used.

- <mix number> - IL - <file identifier> _____

Explanation

<mix number>	This is the mix number of the program requesting the file.
<file identifier>	This is the name contained in the disk or diskpack directory. File identifiers can be up to six characters long. They cannot be made up of all nulls, all blanks, or all zeros.

Details

If you know at run time that the file needed by the program has a different name or location than the program expects, the FILE (Equate File Name with File Identifier or Medium) command could be used to redirect the program.

Examples

4 IL 3/0

This gives the program associated with mix number 004 the location of the labeled file it needs on channel number 3, unit number 0.

21 IL SYMFIL

This gives the program associated with mix number 021 the name of the labeled file it needs as SYMFIL.

32 IL FILE01 ON DMSPAK

This gives the program associated with mix number 032 the name of the labeled file it needs as FILE01, which resides on the diskpack named DMSPAK.

See also:


- The FR (Designate Final Reel) command on Page 2-179.
- The OF (Optional File Indication) command on Page 2-245.
- The UL (Assign Unlabeled File) command on Page 2-383.

IN (Insert Data into Program)

Keyboard Input Message

Access Level: 3

The IN command loads a data string into a job in the mix at the indicated address.

- <mix number> - IN - <starting address> - <length>  <data string> _____

Explanation

<mix number> This is the mix number of the program into which you want to insert the data string.

Note that you cannot specify a locked program or use a mix number of 0 or 1.

<starting address> This is the program base-relative address at which to start inserting. This can range from 000000 through 999999.

<length> This indicates the number of digits (for UN data) or bytes (for UA data) to be inserted and must be 30 or less.

UN These indicate data class.

UA

UN specifies unsigned numeric data and causes digits and undigits to be inserted. Undigits are the hexadecimal values between 10 and 15. The undigits are A, representing the number 10, through F, representing the number 15.

UA denotes unsigned alphanumeric data. This inserts characters with the <length> in bytes.

If a data classification is not specified, UN is assumed unless the data is contained in quotation marks. UA is optional if the data is enclosed in quotation marks.

<data string> This is the information to be inserted into the program.

If you use special characters (period, comma, and so forth), the data is considered to be in UA format and must be enclosed in quotation marks

Two consecutive quotation marks within the <data string> cause a single quotation mark to be inserted.

Details

The IN command can be used on a program that is already executing, unlike the INSERT (Insert Data into Program) command, which is used prior to a program's execution.

Examples:

```
10 IN 16 2 56
```

This inserts 2 digits of UN or numeric data with a value of 56 into the program with mix number 010 beginning at address 16.

```
7 IN 0 8 UA ABCDEF12
```

This inserts 8 bytes of alphanumeric data with the value of ABCDEF12 into the program with mix number 007 beginning at address 0.

```
7 IN 0 8 ''ABCDEF12''
```

This inserts 8 bytes of alphanumeric data with the value of ABCDEF12 into the program with mix number 007 beginning at address 0.

See also:

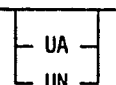
- The INSERT (Insert Data into Program) command on Page 2-197.
- The SW (Set Programmatic Switches) command on Page 2-369.
- The VALUE (Insert Value into a Program) command on Page 2-403.

INSERT (Insert Data into Program)

MCP Control Instruction

Access Level: 1

INSERT places required data into a program prior to its execution.

- INSERT - <starting address> - <length>  <data string>

Explanation

INSERT
IN

These are synonymous.

<starting address>

This is the program base-relative address at which to start inserting. This can range from 000000 through 999999.

Note: In most cases, the <data string> inserted into a program prior to its execution must not terminate at an address greater than 000063. Data inserted into higher base-relative addresses is destroyed by the overlay mechanism generated by most compilers.

<length>

This indicates the number of digits (UN) or bytes (UA) of data to be inserted and must be 30 or less.

UN
UA

These indicate data class.

UN specifies unsigned numeric data and causes digits and undigits to be inserted. Undigits are the hexadecimal values between 10 and 15. The undigits are A, representing the number 10, through F, representing the number 15.

UA denotes unsigned alphanumeric data. This inserts characters with the <length> in bytes.

If a data classification is not specified, UN is assumed unless the data is contained in quotation marks. UA is optional if the data is enclosed in quotation marks.

<data string>

This is the information to be inserted into the program.

If you use special characters (period, comma, and so forth), the data is considered to be in UA format.

Two consecutive quotation marks within the <data string> cause a single quotation mark to be inserted.

INSERT

Details

IN can be used on a program that is already executing, unlike the INSERT (Insert Data into Program) command, which is used at program initiation time.

Examples

```
EXECUTE PROG1; INSERT 0 8 UN 12A00011
```

This executes PROG1 and inserts 8 digits of unsigned numeric data with the value of 12A00011 beginning at address 0.

```
EXECUTE PROG2; INSERT 10 4 UA ABCD
```

This executes PROG2 and inserts 4 bytes of unsigned alphanumeric data with the value of ABCD beginning at address 10.

```
EXECUTE PROG3; INSERT 0 3 UA 'A''B''
```

This executes PROG3 and inserts 3 bytes of unsigned alphanumeric data with the value A"B beginning at address 0.

See also:

- The IN (Insert Data into Program) command on Page 2-195.
- The SW (Set Programmatic Switches) command on Page 2-369.
- The VALUE (Insert Value into a Program) command on Page 2-403.

IR (Initiate DMSII Recovery)

Keyboard Input Message

Access Level: 8

The IR command initiates certain database functions. These include database recovery, reorganization of database files, execution of production database programs (DBPs), purging of database structures, and the loading and dumping of database archive tapes.

For a complete description of the syntax and usage of this command, see the *V Series DMSII Operations Reference Manual Volume 2: Administration*.

()

()

()

KA (Analyze Disk Directory)

Keyboard Input Message

Access Level: 5

The KA function invokes the intrinsic program DSKOUT to generate an analysis of all files found in the disk directory. The following information appears for each file:

- Logical ID
- Area address
- File identifier
- Area numbers, total areas used, and maximum areas declared
- Total number of segments used
- Logical record size
- End of file pointer
- File attributes

You can specify a single file ID, a group file ID, or a subsystem number. The system will analyze only the relevant file or files.

```
- KA [ NOSTANDBY ] [ <file ID> ] [ <subsystem number> ]
```

Explanation

NOSTANDBY

This directs the operating system to capture information for DSKOUT even when the operating system *cannot* perform its normal standby function. The standby function is still attempted, but information is captured regardless of whether the function is successful. For more information about the standby function, see "Details" on the following pages.

<file ID>

This refers to the file ID or masked file ID of files on which an analysis will be performed. The files can reside on any of the existing disk subsystems on the system.

<subsystem number>

This analyzes only the files with areas on that disk subsystem and can be a number from 1 to 8.

Output Options

By default, output for the KA and PA commands is sorted by file ID, and output for the KS (Analyze Disk Space) and PS commands is sorted by address. There are several options, however, for specifying the output. DSKOUT can provide the following:

- Output sorted by total file size
- Output sorted by file ID (alphabetically)
- Output sorted by address (default)
- Output sorted by available area size
- Calculation of wasted space in files, sorted by total file size
- Identification and analysis of code files
- Provision of installation name on printout

These options are specified at execution time with the VALUE statement as follows:

```
VALUE 1 <integer>
```

The following list of options will appear on the ODT if the value of <integer> is 1:

```

DSKOUT=<mix number>          *** ENTER OPTIONS ***
FILESIZE                     SORT BY FILESIZE (KA, PA ONLY)
NAME                         SORT BY NAME (KA, PA ONLY)
ADDRESS                      SORT BY ADDRESS (KA, KS, PA, PS ONLY)
AREASIZE                     SORT BY AVAILABLE AREA SIZE (KS, PS ONLY)
WASTED                      ANALYZE WASTED SPACE (KA, PA ONLY)
CODE                        ANALYZE CODE FILES (KA, PA ONLY)
INSTALL                     ACCEPT INSTALLATION NAME (KA, PA ONLY)
** DSKOUT=                   <mix number> ACCEPT

```

If the value of <integer> is 8, the options will not be displayed on the ODT, but DSKOUT will still wait for an ACCEPT.

The response to the accept should be any valid option or combination of options, in string format. Valid option delimiters are a comma, a slash, and a space. For example:

```

<mix number> AX 'NAME, ADDRESS'
<mix number> AX 'FILESIZE ADDRESS CODE'
<mix number> AX 'AREASIZE/INSTALL'

```

If invalid input is received, DSKOUT displays an error message on the ODT, redisplay the valid options, and then waits for another ACCEPT. This will continue until you enter valid options or use the DS (Discontinue Program) command to terminate the program. Certain output options can be used only with specific commands.

The following list shows the commands that can be used with each option:

OPTION	COMMANDS
FILESIZE	KA, PA
NAME	KA, PA
ADDRESS	KA, KS, PA, PS
AREASIZE	KS, PS
WASTED	KA, PA
CODE	KA, PA
INSTALL	KA, KS, KP, PA, PS, PP

Note: The CODE option is very time consuming, because each file is opened to have the program parameter block and all file parameter blocks read.

The default option is ADDRESS for KS/PS and NAME for KA/PA.

Details

When you enter a KA command, the operating system can respond with the message:

KBD IGNORED: DISK OR PACK MAINT IS IN PROGRESS

This message means the operating system cannot perform a standby function as part of the KA command. A standby function suspends updates to the disk directory and other data structures long enough to obtain an accurate "snapshot" of the structures. This "snapshot" is passed to DSKOUT for analysis. The operating system needs exclusive access to the disk directory for an accurate DSKOUT analysis.

To respond to this situation, you can do either of the following:

- Wait until a later time when disk activity is less.
- Use the NOSTANDBY option of the KA command.

The NOSTANDBY option captures information for DSKOUT even when the operating system *cannot* perform the standby function. The standby function is attempted, but information is captured regardless of whether it is successful.

Using NOSTANDBY can create missing or duplicate disk messages that are not correct. This happens if an update occurs as the "snapshot" is taken. If NOSTANDBY is used and missing or duplicate disk conditions are encountered, DSKOUT flags the resulting messages with NOSTANDBY warnings.

These warnings mean the condition might be caused by non-exclusive access rather than by an actual missing or duplicate disk. Before you act on any missing or duplicate disk messages with a NOSTANDBY warning, enter another KA command without NOSTANDBY. Enter the command when less disk activity is taking place.

Examples

KA =A=BCD

This causes the system to analyze the files that fit the masked file ID of =A=BCD.

KA HFILE

This causes the system to analyze HFILE.

KA 5

This causes the system to analyze the files with areas on disk subsystem number 5.

KA NOSTANDBY 7

This causes the system to analyze the files with areas on disk subsystem number 7. The analysis is generated even if the normal standby function of the KA command is unsuccessful.

See also:

- The KP (Print Disk Segments) command on Page 2-205.
- The KS (Analyze Disk Space) command on Page 2-207.
- The PA (Analyze Pack Directory) command on Page 2-259.
- The PP (Print Pack Sectors) command on Page 2-287.
- The PS (Analyze Pack Space) command on Page 2-303.

KP (Print Disk Segments)

Keyboard Input Message

Access Level: 5

KP invokes the intrinsic program DSKOUT to print selected segments from a disk subsystem. The disk ID, starting segment address, and number of segments specify which segments to print.

- KP - <disk ID> - <begin segment address> <number of segments>

Explanation

- <disk ID> This specifies a previously declared disk unit and can be a number from 1 through 255.
- <begin segment address> This specifies the beginning point for the printing. The address must be stated as an integer and must be between 0 and the drive capacity or the declared limit.
- <number of segments> This indicates the number of segments you want to print. If you omit this option, the system assumes a value of 1.

Examples

KP 3 10

This invokes the DSKOUT intrinsic to print one disk segment beginning with segment 10 from disk ID 3.

KP 2 1557 25

This invokes the DSKOUT intrinsic to print 25 disk segments beginning with segment 1557 from disk ID 2.

KP 1 1489 50; PR 5

This invokes the DSKOUT intrinsic to print 50 disk segments beginning with segment 1489 from disk ID 1 and assigns the DSKOUT intrinsic a setting of 5 for the memory priority, I/O priority, processing priority and scheduling priority.

See also:

- The KA (Analyze Disk Directory) command on Page 2-201.
- The KS (Analyze Disk Space) command on Page 2-207.
- The PA (Analyze Pack Directory) command on Page 2-259.
- The PP (Print Pack Sectors) command on Page 2-287.
- The PS (Analyze Pack Space) command on Page 2-303.

KS (Analyze Disk Space)

Keyboard Input Message

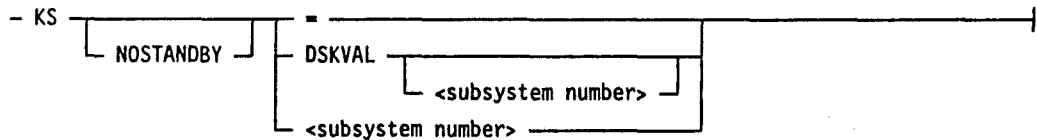
Access Level: 5

KS invokes the intrinsic DSKOUT to analyze the allocation of disk space. Files, available areas, and missing and duplicate disk space are listed. The following are listed for each file:

- Logical ID
- Area addresses
- File identifier
- Area number and size of area
- File attributes

For available areas, the logical ID, area address, the string "A V A I L A B L E", and the size of the available space are listed. For missing or duplicate disk space, the logical ID, address of the area, the string "MISSING DISK" or "DUPLICATE DISK", and size of the area are listed.

You can generate an analysis of only a specified subsystem or analysis of only available space on disk.



Explanation

NOSTANDBY	This directs the operating system to capture information for DSKOUT even when the operating system <i>cannot</i> perform its normal standby function. The standby function is still attempted, but information is captured regardless of whether the function is successful. For more information about the standby function, see Details on the following pages.
=	This is a masking character that can be used to cause an analysis of all disks on the system.
DSKAVL	This causes an analysis of only the unassigned disk space on the system or on the specified subsystem.
<subsystem number>	This causes an analysis of only the specified disk subsystem.

Output Options

Various output options are available for alternate sorting and analysis. See the KA (Analyze Disk Directory) command for a full description of these options and their applications.

Details

When you enter a KS command, the operating system can respond with the message:

```
KBD IGNORED: DISK OR PACK MAINT IS IN PROGRESS
```

This message means the operating system cannot perform a standby function as part of the KS command. A standby function suspends updates to operating system data structures long enough to obtain an accurate "snapshot" of them. This "snapshot" is passed to DSKOUT for analysis. The operating system needs exclusive access to the data structures for an accurate DSKOUT analysis.

To respond to this situation, you can do either of the following:

- Wait until a later time when disk activity is less.
- Use the NOSTANDBY option of the KS command.

The NOSTANDBY option captures information for DSKOUT even when the operating system *cannot* perform the standby function. The standby function is attempted, but information is captured regardless of whether it is successful.

Using NOSTANDBY can create missing or duplicate disk messages that are not correct. This happens if an update occurs as the "snapshot" is taken. If NOSTANDBY is used and missing or duplicate disk conditions are encountered, DSKOUT flags the resulting messages with NOSTANDBY warnings.

These warnings mean the condition might be caused by non-exclusive access rather than by an actual missing or duplicate disk. Before you act on any missing or duplicate disk messages with a NOSTANDBY warning, enter another KS command without NOSTANDBY. Enter the command when less disk activity is taking place.

Examples

```
KS =
```

This invokes the DSKOUT intrinsic to analyze disk space on all the disks on the system.

```
KS DSKAVL; PR = 9; LOCK
```

This invokes the DSKOUT intrinsic to analyze only the unassigned disk space on the system; assigns the job a memory, schedule, I/O, and processing priority of 9; and locks the job, thus preventing it from being discontinued or modified.

KS NOSTANDBY 6

This invokes the DSKOUT intrinsic to analyze disk space on disk subsystem 6. The analysis is generated even if the normal standby function of the KS command is unsuccessful.

See also:

- The KA (Analyze Disk Directory) command on Page 2-201.
- The KP (Print Disk Segments) command on Page 2-205.
- The PA (Analyze Pack Directory) command on Page 2-259.
- The PP (Print Pack Sectors) command on Page 2-287.
- The PS (Analyze Pack Space) command on Page 2-303.

KX (Obtain User Disk Space)

Keyboard Input Message

Access Level: 0

The KX command removes disk files to make disk space available. This is generally done in response to a no-user-disk message.

The KX command lists the files that can be removed. The files are ordered as they appear in the directory. For each file found, a line indicates the name of the file, segments per area, maximum number of contiguous disk segments, and the subsystem on which the file resides. If the SECURITY option is set, the usercode of each file is also displayed. You can then keep or remove the file.

```
- KX _____
  |-----|
  | - - - |
  | <size> |
```

Explanation

KX

KX =

These are synonymous and display a list of removable files in the directory. Removable files are those not in use and whose security attributes allow library maintenance (the LIBMAINT file attribute is set). See the *V Series System Software Security Installation and Operations Guide* for more information.

<size>

This is an integer that specifies that only files containing at least the indicated number of contiguous disk segments are to be listed.

A KX command without the <size> option has zero as the number of contiguous segments and hence will list all files.

Responses to KX

There are four possible responses to each line of output from the KX request:

- A null response (ETX) leaves the listed file on disk and continues searching for another file to remove.
- RM removes the listed file from disk. However, the file might be opened between the time it is listed and the time the RM is entered and, therefore, it cannot be removed. If the file was successfully removed, the MCP continues to search for files. If the file has not been removed, a message is displayed saying that the file is not removed, and the MCP continues to search for files.

- KX <size> restarts the KX with a new size. The listed file remains on disk. Although <size> is required, it can be zero.
- END terminates the KX attempt. The listed file remains on disk.

AUTO KX

If you have run out of disk space and the MCP tries to add more data to the disk, the MCP might initiate automatically a KX operation. This is known as AUTO KX, and acts similarly to the KX command in that you have to analyze the disk directory and remove files to obtain the needed disk space.

If a response of END is given to an AUTO KX and insufficient disk space has been made available, the AUTO KX is reinitiated.

Caution

Do not perform a halt/load while an AUTO KX is in progress. If AUTO KX is invoked due to an MCP log transfer and a halt/load is performed to terminate the KX, it might corrupt disk directories and data files.

Examples

KX =

This lists all removable files. This can be viewed as a KX specifying zero as the number of contiguous segments.

KX 4000

This lists only files with at least 4000 contiguous disk segments.

See also:

- The KS (Analyze Disk Space) command on Page 2-207.
- The RX (Return Deleted Space) command on Page 2-336.
- The WX (Display Removed Space) command on Page 2-425.

LABEL1 (Change Tape Label Format)

MCP Control Instruction

Access Level: 0

The LABEL1 command changes the format of an installation-defined magnetic tape label.

The LABEL1 instruction must be entered through a card reader, and the punched card must contain an invalid character in column 1. The word LABEL1 can appear anywhere in columns 2 through 8.

- LABEL1 - <label parameters> _____

Explanation

<label parameters> These are the parameters you can enter to customize the tape label.

The record is 67 columns long. The format of the tape label record is listed in Table 2-6.

Note that if multifile identifiers are specified, the length of the file identifier must be 6 characters or less. However, if the length of the MFID is 0, the file identifier can be up to 12 characters long. File identifiers with lengths of 10, 11, and 12 are specified by the letters A, B, and C, respectively.

Table 2-6. Tape Label Record Format

Column Positions	Position Contents	Description
1-8	LABEL1	The control word LABEL1 can appear anywhere in the field.
9	Blank	Skip this column.
10-19	1-10 characters	Text of beginning of file label ID.
20-21	0-99	Position of beginning of file label ID field.
22-23	1-10	Length of beginning of file label ID field.
24-25	0-99	Position of multifile ID field.
26	0-C (C=12)	Length of multifile ID field (0 = none).
27-28	0-99	Position of file ID field.
29	1-C (C=12)	Length of file ID field.

Table 2-6. Tape Label Record Format

Column Positions	Position Contents	Description
30-31	0-99	Position of reel number field.
32	1-9	Length of reel number field.
33-34	0-99	Position of creation date field.
35-36	0-99	Position of purge date field.
37	3-9	Length of purge date field (a value of less than 5 implies a retention period instead of purge date).
38-39	0-99	Position of volume ID field.
40	1-9	Length of physical volume ID field.
41-42	0-99	Position of end of reel or end of file identifier.
43	1-5	Length of end of reel or end of file label identifier.
44-48	1-5 characters	Text of end of reel label ID.
49-53	1-5 characters	Text of end of file label ID.
54-56	20-998	Minimum length of label records.
57-58	0-99	Number of label records (0=variable).
59	0-1	1=tape mark after end labels.
60	0-1	1=tape mark after beginning labels.
61-62	0-99	Position of block count field.
63	0-9	Length of block count field (0=none).
64-65	0-99	Position of record count field (end labels only).
66-67	0-12 (0=none)	Length of record count field.

Example

LABEL1 **BEGIN***001018659642368545114745*ENDR*ENDF20001114653107

This changes the format of an installation-defined tape label.

Columns 1 through 8 contain LABEL1; column 9 is blank.

Columns 10 through 19 contain **BEGIN*** as the beginning of file label identifier. Columns 20 and 21 contain 00 as the position of the beginning of the file label identifier. Columns 22 and 23 contain 10 as the length of the beginning of file label identifier.

LABEL1

Columns 24 and 25 contain 18 as the position of the multifile ID; column 26 contains 6 as the length of the multifile ID.

Columns 27 and 28 contain 59 as the position of the file ID; column 29 contains 6 as the length of the file ID.

Columns 30 and 31 contain 42 as the position of the reel number; column 32 contains 3 as its length.

Columns 33 and 34 contain 68 as the position of the creation date; columns 35 and 36 contain 54 as the position of the purge date; and column 37 contains 5 as the length of the purge date.

Columns 38 and 39 contain 11 as the position of the volume ID; column 40 contains 4 as its length.

Columns 41 and 42 contain 74 as the position of the EOR or EOF identifier; column 43 contains the length of the EOR or EOF identifier. Columns 44 through 48 contain *ENDR as the text of the EOR label identifier; columns 49 through 53 contain *ENDF as the text of the EOF label identifier.

Columns 54 through 56 contain 200 as the minimum length of label records; columns 57 and 58 contain 01 as the number of label records.

Columns 59 and 60 each contain a 1 to specify a tape mark after end labels and after beginning labels, respectively.

Columns 61 and 62 contain 46 for the position of the block count; column 63 contains 5 as the length of the block count.

Columns 64 and 65 contain 31 as the position of the record count; columns 66 and 67 contain 07 as the length of the record count.

See also:

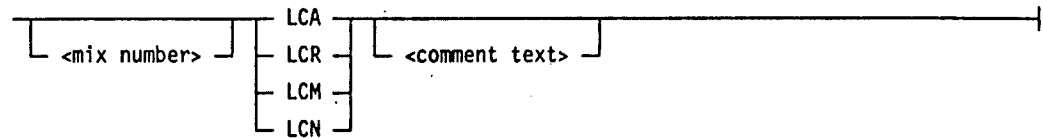
- The AC (Purge and Place Tape Number in Label) command on Page 2-2.
- The CN (Display Tape Number) command on Page 2-78.
- The SN (Purge and Place Tape Number in Label) command on Page 2-350.
- The TM (Write Tape Mark) command on Page 2-377.
- The INSTALLATION LABEL1 record in the *V Series Systems Operations Guide Volume 1: Installation*.

LC (Insert Log Comment)

Keyboard Input Message

Access Level: 1

The LC command inserts a comment into a system log file.



Explanation

<mix number>	This is the mix number of a currently active, waiting, or stopped job. You cannot use a mix number of 0 or 1, which indicates the MCP. The <mix number> can be omitted if you do not want to associate the comment with a particular job.
LCA	This inserts the comment in all logs.
LCR	This inserts the comment in the Run log and the ODT log.
LCM	This inserts the comment in the Maintenance Log and the ODT log.
LCN	This inserts the comment in the Network log.
<comment text>	This contains the text of the comment and can be up to 75 characters long. The <comment text> must be enclosed in quotation marks if it contains non-alphanumeric characters.

Details

You can use this command only if the requested log is in use. See the USE LOGGING records in the *V Series Systems Operations Guide Volume 1: Installation*.

The comment is placed in a record type 5/0 in the Run log and the Maintenance log, if specified.

Examples

LCA START OF SHIFT

This places this comment in all logs. Note that the comment does not have to be enclosed in quotation marks because it contains only alphanumeric characters.

12 LCR 'GIVEN WRONG FILE - DSED'

This places this comment in the Run log. The quotation marks are required because the comment contains a hyphen. This comment is associated with the job with mix number 012.

LCN STATION X2594 DISABLED FOR RELOCATION

This places this comment in the Network log. Quotes are not required.

LCM '2/3 TAKEN OUT OF USE - PARITY ERRORS'

This places this comment in the Maintenance Log. The quotation marks are required because the comment contains a slash.

See also:

- The USE RLOG Record, USE SLOG Record, USE MLOG Record, and USE NLOG Record in the *V Series Systems Operations Guide Volume 1: Installation*.
- The *V Series System Software Logging Operations Reference Manual*.

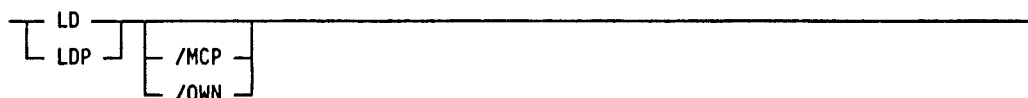
LD, LDP (Create Pseudo Card Deck)

Keyboard Input Message

Access Level: 7

The LD command schedules the LDCNTL intrinsic program, which reads card files and converts them into pseudo card files on disk or diskpack.

The LDCNTL program can also be scheduled through a CV (Convert Punch Backup File to Pseudo Card Deck) or an EXECUTE (Execute Library Program) command.



Explanation

LD directs the pseudo decks to disk; LDP directs them to diskpack.

/MCP executes the version bound to the MCP. /OWN executes a free-standing user-written version.

If this option is omitted, the user-coded version is initiated if present on disk; otherwise the MCP intrinsic is initiated.

Examples

```
LD; AFTER MYPROG; PRIORITY 5
```

This starts LDCNTL after a program called MYPROG finishes and assigns it a memory, processing and I/O, and schedule priority of 5.

See also:

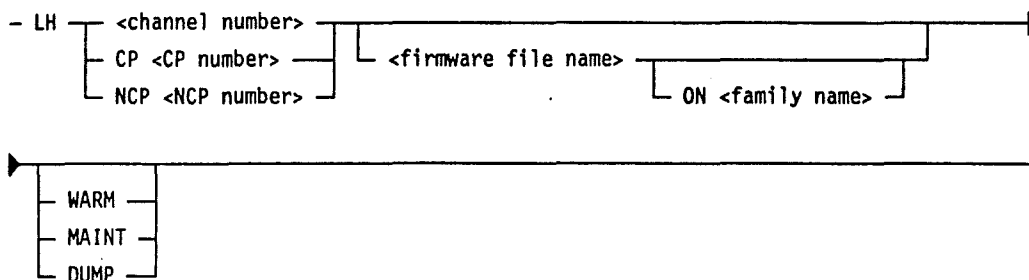
- The CV, CVP (Convert Punch Backup File to Pseudo Card Deck) commands on Page 2-100
- The DA (Deactivate Pseudo Reader) command on Page 2-103.
- The EXECUTE (Execute Library Program) command on Page 2-162.
- The OL (Display Peripheral Status) command on Page 2-248.
- The RN, RNP (Activate Pseudo Reader) commands on Page 2-324
- The LDCNTL utility in the *V Series Systems Operations Guide Volume 3: System Utilities*.

LH (Load Firmware to Channel Controller/DCP)

Keyboard Input Message

Access Level: 8

The LH command invokes the utility program LDHOST to download firmware to a programmable controller, a Data Communication Processor (DCP), or a UNILINE device. In the case of a B 974 DCP, the LH command invokes the utility program B974LD. In the case of communications devices connected through the V Series Communication System (VCS), LH invokes the utility program CPLOAD.



Explanation

- <channel number>** This specifies the channel to which the controller is connected. It can be a secondary channel on an exchange. This option is not used if the device on the channel is connected through the V Series Communication System (VCS).
- CP <CP number>** This loads firmware to the specified Communication Processor (CP) connected through VCS. Examples of CPs that can be connected through VCS include a CP 3680 Front End Processor or a TELCOM DLP. This option is used for data communications devices that have a physical connection to individual stations.
- NCP <NCP number>** This loads firmware to the communications processor (CP) connected to the specified Network Communications Port (NCP) that is connected through VCS. This option functions the same as the CP <CP number> option.
- <firmware file name>** This is the name of the firmware file to be loaded. The firmware file must be on disk when the LH command is issued.
- This option is not required if standard firmware or a DCP, as specified in the Network Definition Language source file, is being host loaded to the controller.

Standard firmware includes that specified on the DLP record of the system configuration file.

For the CP <CP number> or NCP <NCP number> options, the firmware file name is the name of the VCS initialization file.

- ON <family name>** This option is valid only when used with the NCP <NCP number> or CP <CP number> options of the LH command. The option indicates the diskpack family where the firmware file for the NCP or CP resides.
- WARM** This can be used only with DCPs. It specifies that the internal DCP tables are not to be reloaded. The WARM option can allow recovery from an error condition without restarting the entire data comm network.
- MAINT** This can be used with any hardware type to specify that the channel is to be left inhibited at the completion of the load. It allows the loading of diagnostic firmware (for example, a diskpack exerciser) to one controller on an exchange while continuing normal operations on the other exchange channels.

Note: The WARM and MAINT options must be used in separate commands. They cannot be used together.

- DUMP** This option is valid only when used with the NCP <NCP number> or CP <CP number> option of the LH command. The DUMP option causes the memory of the data communications device to be stored in a file before firmware is loaded. The file resides on the V Series system. The default value does not request a memory dump.

Details

Table 2-7 shows host load files and their usage.

Table 2-7. Host Load Files and Their Usages

Name	Usage
HSTLAD	Loads firmware to a B 9389 controller for 680 or 682 diskpack by means of SEQ-HT.
HSTLKC	Loads firmware to a B 9384 or B 9385 controller for 225 pack by means of HT-DLP.

Table 2-7. Host Load Files and Their Usages (cont.)

Name	Usage
HSTLQH	Loads firmware to a B 9387 controller for 206/207/677/659 pack by means of HT-DLP.
HSTLUD	Loads firmware to a B 9387 controller for 206/207/677/659 pack by means of SEQ-HT. If the controller is not correctly updated for the HSTLUD version, the firmware is "changed" to HSTLUC version.
USP2BV	Loads UNILINE SPO firmware.
UST2BH	Loads UNILINE terminal firmware.
IPPFMW	Loads Loadable Operating System (LOS) firmware to an IPP.
LSDIAG	Load Loadable Diagnostics to an IPP.

The MD4 and MD8 disk and diskpack does not utilize a controller and, therefore, does not require any controlware to be loaded.

Uses

A DCP can be host loaded when the DCP option is reset, the DCP is marked "LH REQUIRED", or the DCP is not in use.

If a B 974 or B 874 DCP is not working properly because its data communication code has become corrupted, the MCS programs will receive indication that the DCP is not responding. The LH command will reload the code into the B 974 or B 874 DCP.

You might want to use the LH command if the NDL source has been recompiled and is to be loaded into one particular B 974 DCP. This would enable different versions of the NDL to be used on different DCPs.

Details for the B 974 DCP

If the LH command initializes a B 974 DCP, the B 974 dumps its S-memory to the host disk with the file name DC cc sP, where cc is the channel number of the DCP, and s is the system number of the host. The dump, if one exists, uploads to the host with the file name DC dnn P, where d is the DCP number of the B 974 DCP, and nn is the dump number of the B 974 DCP.

Example

```
LH 2 HSTLQH
```

This invokes LDHOST to load the programmable controller on channel 2.

See also:

LDHOST in the *V Series Systems Operations Guide Volume 3: System Utilities*.

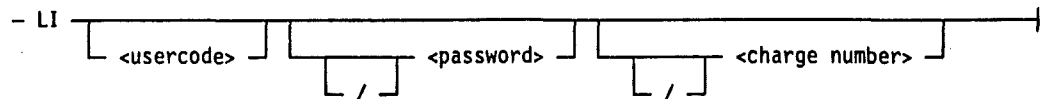
LI (Log In)

Keyboard Input Message and MCP Control Instruction

Access Level: 0

The LI command logs an ODT or card reader into the system. This command must be entered from the device that is being logged in.

The syntax is the same for both the LI keyboard input message and the LI MCP control instruction. It is shown in the following figure.



Explanation

<usercode>	This refers to a code that identifies a user to the system.
<password>	This refers to a group of characters by which the system determines the legality of a usercode.
<charge number>	This refers to a number used to assign charges for system use.

Note: The <usercode>, <password>, and <charge number> are all optional. Slashes are required only if leading parameters are omitted (for example, LI //90023).

System Security

System security refers to system access control and disk file security functions. System access control restricts access to the system and defines each user's capabilities on the system. Disk file security controls programmatic and operator access to permanent disk and diskpack files. In addition, disk file security allows different file access mode characteristics, such as public or private.

With system security, users identify themselves to the system before performing system functions. Identification is contained in an access code, which can consist of a usercode, password, and charge number combination.

The user enters the usercode, password, and charge number access combination when logging in to the system or entering the BEGINUSER (Beginning Of Job Stream) or USER (Assign Access Code) commands.

After log-in, the MCP checks each command entered with the access level associated with that user. If the command is permitted, it is performed, but if it is invalid, an appropriate message is displayed.

See also:

- The BEGINUSER (Mark Beginning of Job Stream) control instruction on Page 2-45.
- The ENDUSER (Mark End of Job Stream) control instruction on Page 2-158.
- The LO (Log Out) command on Page 2-225.
- The USER (Assign Access Code) command on Page 2-401.
- The *V Series Security Installation and Operations Guide* for information about <usercode>, <password>, and <charge number>.

LN (Transfer and Print Log)

Keyboard Input Message

Access Level: 8

The LN command closes the current system log file, directs logging information into a new log file, and starts the log analysis program. The log analysis program produces a printer backup file containing a report on the log.



Explanation

LNA	This transfers and analyzes all log files—Run log, Maintenance log, ECM log, and ODT log.
LNE	This transfers and analyzes the ECM log. The Environmental Control Module log is created only on V 500 series systems and is stored on the hard disk of the system console.
LNM	This transfers and analyzes the Maintenance log.
LNN	This transfers and analyzes the Network log.
LNR	This transfers and analyzes the Run log.
LNS	This transfers and analyzes the ODT log.

Use

Log transfer happens automatically when a log file becomes full or at LOGXFERTIME. The LN or TL (Transfer Log) command can be used to close the current log file.

The LN command transfers the log and starts the log analysis program. LN always starts the log analysis program. TL starts the log analysis program only if the AUTO option is set in the USE record of the log file in the system configuration file.

The LN command combines the functions of a TL (Transfer Log) command and an EXECUTE (Execute Library Program) request of the appropriate log analysis program.

Details

The system configuration file can contain a USE record for the RLOG and SLOG (that is, the USE RLOG and USE SLOG records). If a USE record of a log is present, the log will be created.

If the AUTO option is set on a USE record of a log, the appropriate log analysis and reporting program is started whenever that log is transferred, whether it is transferred automatically when it fills up, at LOGXFERTIME, or manually by the TL or LN command.

If the WRAP option is set on a USE record of a log, log files are renamed each time they transfer by incrementing a number in the file name from 0 through 9. For example, the name of the MLOG would change from MLOGs1 to MLOGs9 (where *s* is the system number). If this option is not set, the file names do not change, and one log file will overwrite the other file at log transfer time. If this option is not used, the files will be renamed MLOGs0 to MLOGs1.

Examples

LNM

This transfers and analyzes the Maintenance log.

LNA

This transfers and analyzes all logs.

See also:

- The TL (Transfer Log) command on Page 2-375.
- The USE LOG and LOGXFERTIME records in the *V Series Systems Operations Guide Volume 1: Installation*.
- The *V Series System Software Logging Operations Reference Manual* for complete details on system logging.

LO (Log Out)

Keyboard Input Message and MCP Control Instruction

Access Level: 0

The LO command logs a peripheral device off the system.

The syntax shown in the following figure is the same for both the LO keyboard input message and the LO MCP control instruction.

- LO _____

See also:

- The BEGINUSER (Mark Beginning Of Job Stream) control instruction on Page 2-45
- The ENDUSER (Mark End Of Job Stream) command on Page 2-158.
- The LI (Log In) command on Page 2-221.
- The USER (Assign Access Code) command on Page 2-401.

LOAD (Move Files from Library Tape to Disk or Diskpack)

MCP Control Instruction

Access Level: 5

The LOAD command moves files from library tape to disk or diskpack and then replaces any existing file with any file on the tape having the same file identifier.

It is recommended that you use the COPY command instead of the LOAD command. The LOAD command will become unavailable at a future release.

The LOAD command starts a program called LOADMP, if the files are to be moved from disk, or a program called PACKUP, if files are to be moved from diskpack. The user must supply the LOADMP and PACKUP programs because Unisys does not supply them. If the appropriate program is not present on disk, the LOAD command is ignored.

See Appendix B for LOAD syntax if you have user-coded LOADMP and/or PACKUP utilities.

See also:

- The COPY (Invoke File Transfer Utility) command on Page 2-86.
- Appendix B.

LOCK (Lock Program)

MCP Control Instruction

Access Level: 1

The LOCK command prevents a job from being discontinued or modified.

— LOCK —

Details

The LOCK command can be used with any library maintenance or system command that causes the initiation of a program.

You cannot use the following commands on a locked program:

- DP (Dump and Discontinue)
- DS (Discontinue Program)
- IN (Insert Data into Program)
- QT (Quit Program Operation)
- RS (Remove Job from Schedule)
- SK (Skip Backup Records)
- SW (Set Programmatic Switches)

To use one of these commands, you must first unlock the program by using the UP (Unlock Program) command.

Examples

```
EXECUTE PROGA; LOCK
```

This executes and locks a program called PROGA.

```
PFM LIST FILEA N; CG 27; PR 5; LOCK
```

This schedules the MCP bound intrinsic DMPALL to print the contents of FILEA in a numeric output format; assigns the job a charge number of 27; gives it a memory, I/O and processing, and schedule priority setting of 5; and locks the program.

See also:

- The LP (Lock Program) command on Page 2-228.
- The UP (Unlock Program) command on Page 2-398.

LP (Lock Program)

Keyboard Input Message

Access Level: 1

The LP command locks a specified program, preventing it from being discontinued or modified until it is unlocked by the UP (Unlock Program) command.

- <mix number> - LP

Explanation

<mix number> This is the mix number of the program you want to lock.

Details

The following commands are invalid for locked programs:

- DS (Discontinue Program)
- DP (Dump and Discontinue)
- IN (Insert Data into Program)
- QT (Quit Program Operation)
- RS (Remove Job from Schedule)
- SK (Skip Backup Records)
- SW (Set Programmatic Switches)

Example

54 LP

This locks the job associated with mix number 54.

See also:

- The LOCK (Lock Program) command on Page 2-227.
- The UP (Unlock Program) command on Page 2-398.

MEMDUMP (Abnormal EOJ Memory Dump)

MCP Control Instruction

Access Level: 1

The MEMDUMP command causes program memory to be dumped if the job terminates abnormally. However, the MEMDUMP command is not executed on any jobs terminating because of a memory parity error.

— MEMDUMP —

Details

If a program experiences a fault or attempts an illegal action upon a file, the MCP terminates the program. If the program has been executed with the MEMDUMP command or if the TRMD is set, a program fault causes the program to abort with a memory dump unless termination was due to a memory parity error.

Once the file is dumped, it can be printed and analyzed by using the PM (Print Memory Dump) command.

Example

```
EXECUTE APROG; MEMDUMP
```

This executes a program called APROG and specifies that program memory will be dumped if the file terminates abnormally.

See also:

- The DM (Dump And Continue) command on Page 2-142.
- The DP (Dump and Discontinue) command on Page 2-144.
- The PM (Print Memory Dump) command on Page 2-284.
- The USE TRMD record in the *V Series Systems Operations Guide Volume 1: Installation*.

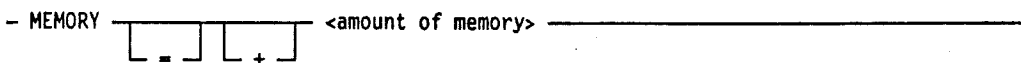
MEMORY (Allocate Memory for Program Execution)

MCP Control Instruction

Access Level: 3

When a program is compiled, the amount of memory it needs in order to execute is specified by the programmer or, by default, is determined by the compiler. The MEMORY command overrides this requirement.

The MEMORY command is ignored if it specifies less memory than the amount chosen when the program was compiled.



Explanation

MEMORY
MEM

These are synonymous and allocate memory for the execution of a program.

=

This is optional and does not influence the command.

+

This adds the specified amount of memory to the default minimum. Multiple MEMORY + <amount of memory> statements are cumulative.

<amount of memory>

This indicates the amount of memory required if used without the plus sign. If the <amount of memory> is used with the plus sign, it indicates the amount of memory to add to the default minimum.

If less than four digits are used to indicate the amount of memory, they indicate thousands of digits (kd). That is, 1 equals 1000 digits of memory; 22 equals 22,000 digits of memory; 333 equals 333,000 digits of memory.

If four to six digits are used, they indicate digits of memory. If that figure is not a multiple of 1,000, it is changed to the next highest multiple of 1,000.

Examples

EXECUTE PAYROL; MEM = 30000

This starts a program called PAYROL and specifies that it needs 30,000 digits of memory to execute.

PERFORM MTPMTP A 1000 10 B 1000 10; MEM 44

This starts the MCP DMPALL utility with 44,000 digits of memory to perform a media conversion. A file with the file identifier of A, with an input record length of 1,000 bytes, and a blocking factor of 10, will be copied to a file with a file identifier of B, a record length of 1000 bytes, and a blocking factor of 10.

COMPILE BTEST PASCAL; MEM + 80

This compiles a Pascal program called BTEST, allocating to the Pascal compiler 80,000 digits more than the default minimum amount of memory. (Note that MEM on a COMPILE command does not cause any change in the size of the compiled program.)

MERGE (Transfer Files from Multiple Sources, Resolving Duplicates)

MCP Control Instruction

Access Level: 5

MERGE invokes the SYSTEM/COPY utility to transfer files from any number of sources to up to 10 destinations. Duplicate files are resolved by the MERGE function.

See the COPY (Invoke File Transfer Utility) command for details on syntax and how to use this command.

See also:

- The COPY (Invoke File Transfer Utility) command on Page 2-86.
- SYSTEM/COPY in the *V Series Systems Operations Guide Volume 3: System Utilities*.

MOVE (Move Files to New Media)

MCP Control Instruction

Access Level: 5

MOVE invokes the SYSTEM/COPY utility to transfer one or more files from one medium to another and remove them from the original medium.

See the COPY (Invoke File Transfer Utility) command, for details on syntax and how to use this command.

See also:

- The COPY (Invoke File Transfer Utility) command on Page 2-86.
- SYSTEM/COPY in the *V Series Systems Operations Guide Volume 3: System Utilities*.

MR (Remove Duplicate File)

Keyboard Input Message

Access Level: 5

The MR command removes a new file from disk or diskpack to resolve a DUP LIB situation. The old file will remain intact.

A DUP LIB situation occurs when a program is trying to create a file but a file with that name already exists.

The MR command removes the output file, allowing the "old" file to remain on disk or diskpack and the program to resume processing.

- <mix number> - MR _____

Explanation

<mix number> This is the mix number of the task that is creating the file you want to remove.

Details

The MR command removes the "new" file and leaves the "old" file, while the RM (Remove Duplicate File) command removes the "old" file and leaves the "new" file.

The RMOV option, if set, can resolve DUP LIB conditions on disk or diskpack by indicating that the old file is to be removed.

Example

The following is a scenario of messages displayed on the ODT in which the MR command is used. (These messages would be displayed on your system only if the OPEN, CLOS, BOJ, and EOJ system options were set.)

ODT Display	Comment
==> EXECUTE JOB1	The operator starts a program called JOB1.
BOJ JOB1=012 012586 9:15	JOB1 starts to execute and is assigned a mix number of 012.
DSK FILE FILEA OPEN OUT JOB1=012	JOB1 opens a file called FILEA. It is a disk file and has been opened for output.
DSK FILE FILEA CLOSED RELEASED JOB1=12	JOB1 closes the file FILEA.

ODT Display	Comment
==> EXECUTE JOB2	The operator starts a program called JOB2.
BOJ JOB2=018 103185 9:20	JOB2 starts to execute and is assigned a mix number of 018.
DSK FILE FILEA OPEN OUT JOB2=018	JOB2 creates a file that happens to be called FILEA and that is a disk file.
**DUP LIB FILEA JOB2=018	JOB2 tries to close the file FILEA. The MCP detects that there is already a file called FILEA. JOB2 waits.
==> SPO 018 MR	The operator enters the MR command. This will remove the version of FILEA that has been created by JOB2. The "old" version of FILEA, created by JOB1, will remain on disk.
EOJ JOB2=018 9:30 IN 1:36 CHG 2:22	JOB2 continues executing until it finishes.

See also:

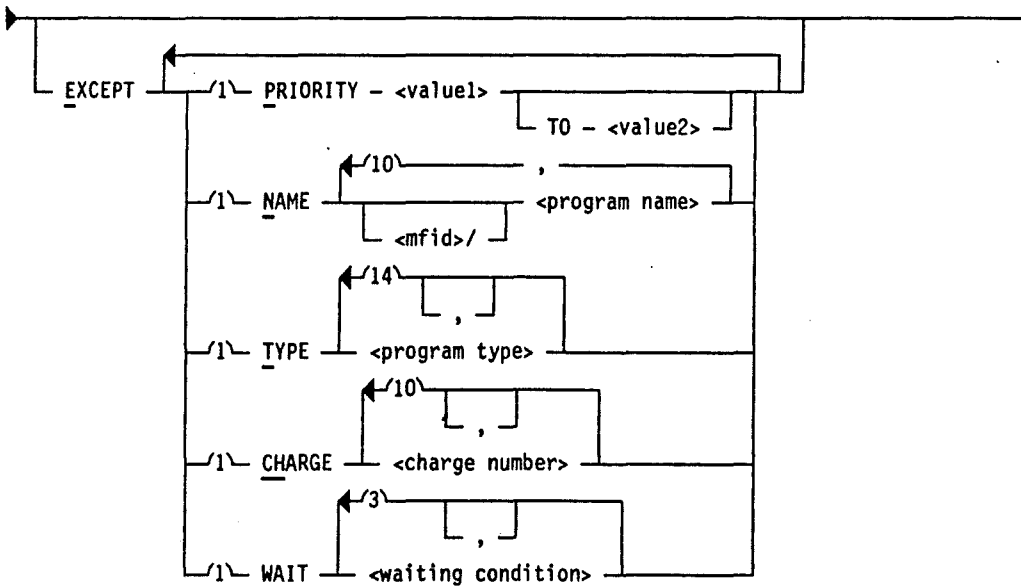
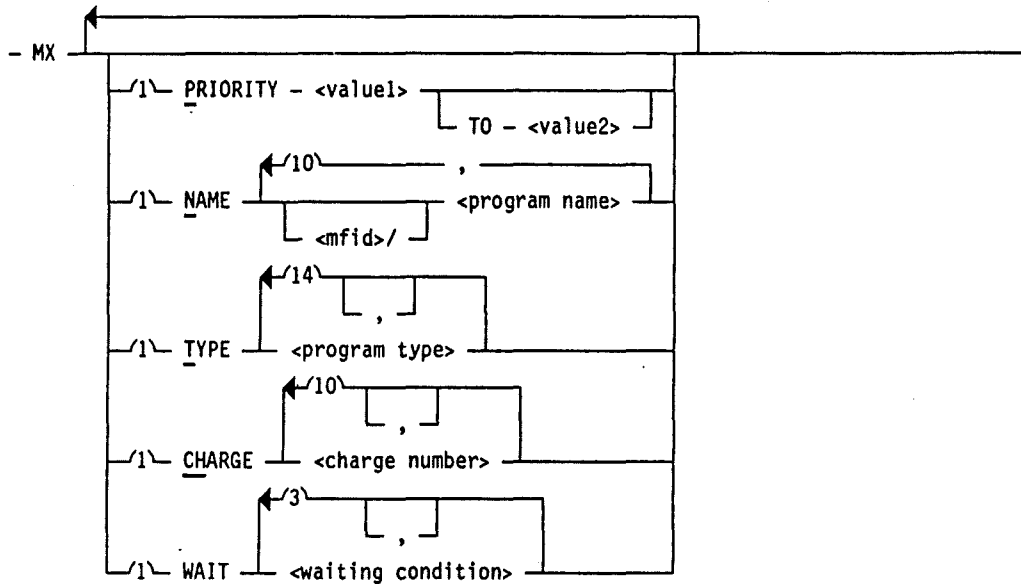
- The RM (Remove Duplicate File) command on Page 2-322.
- The REMOVE (Delete Files) command on Page 2-315.
- The USE RMOV record in the *V Series Systems Operations Guide Volume 1: Installation*.

MX (Display Jobs in Mix)

Keyboard Input Message

Access Level: 0

The MX command displays a list, sorted by job priority, of the jobs currently in the mix. These jobs are active, waiting, or stopped. You can list a subset of the jobs in the mix by using syntax options of the command.



Explanation

PRIORITY <value1> TO <value2>	Selects jobs based on processor priority. You have the option of selecting a single priority or a range of priorities. The values of <value1> and the optional <value2> must be integers in the range of 1 through 9, and <value1> must be less than or equal to <value2>.																												
NAME <multiprogram name>/ <program name>	<p>Selects jobs based on the program name and (optionally) on the multiprogram name. You can enter a maximum of 10 name selections, separated by commas.</p> <p>You can also select jobs based on part of a name by using an equal sign for any unknown characters in the name. For example, the following command displays all programs in the mix beginning with the three characters DMP, but does not display the program DMPDOUT.</p> <p>MX NAME DMP= EXCEPT NAME DMPDOUT</p>																												
TYPE <program type>	<p>Selects jobs based on the type of program. You can enter a maximum of 14 type selections. The available program types are as follows:</p> <table> <tr> <td data-bbox="787 1029 933 1060"><u>BNAHNDL</u></td> <td data-bbox="990 1029 1339 1060">Program is a BNA handler.</td> </tr> <tr> <td data-bbox="787 1071 933 1102"><u>COMPILER</u></td> <td data-bbox="990 1071 1388 1102">Program generates object code.</td> </tr> <tr> <td data-bbox="787 1113 950 1144"><u>COMPSHAR</u></td> <td data-bbox="990 1113 1453 1186">Program that generates object code, executed as a timesharing task.</td> </tr> <tr> <td data-bbox="787 1197 868 1228"><u>COPY</u></td> <td data-bbox="990 1197 1453 1228">Program is SYSTEM/COPY intrinsic.</td> </tr> <tr> <td data-bbox="787 1239 909 1270"><u>DMPALL</u></td> <td data-bbox="990 1239 1372 1270">Program is DMPALL intrinsic.</td> </tr> <tr> <td data-bbox="787 1281 925 1312"><u>DMSPROG</u></td> <td data-bbox="990 1281 1453 1354">Program is a DMSII control program (for example, DBP).</td> </tr> <tr> <td data-bbox="787 1365 909 1396"><u>DSKOUT</u></td> <td data-bbox="990 1365 1404 1438">Program is DSKOUT intrinsic or diskpack SQUASH intrinsic.</td> </tr> <tr> <td data-bbox="787 1449 925 1480"><u>EXECUTE</u></td> <td data-bbox="990 1449 1266 1480">Program is executing.</td> </tr> <tr> <td data-bbox="787 1491 941 1522"><u>HANDLERS</u></td> <td data-bbox="990 1491 1477 1564">Program is any type of handler (BNA, WFL, time sharing).</td> </tr> <tr> <td data-bbox="787 1575 909 1606"><u>LOADMP</u></td> <td data-bbox="990 1575 1372 1606">Program is LOADMP intrinsic.</td> </tr> <tr> <td data-bbox="787 1617 852 1648"><u>MCS</u></td> <td data-bbox="990 1617 1445 1690">Program is Message Control System (MCS).</td> </tr> <tr> <td data-bbox="787 1701 909 1732"><u>PACKUP</u></td> <td data-bbox="990 1701 1372 1732">Program is PACKUP intrinsic.</td> </tr> <tr> <td data-bbox="787 1743 958 1774"><u>SHAREHNDL</u></td> <td data-bbox="990 1743 1412 1816">Program is a timesharing handler program (for example, CANDE).</td> </tr> <tr> <td data-bbox="787 1827 860 1858"><u>SORT</u></td> <td data-bbox="990 1827 1453 1858">Program is SORT: or SORT. intrinsic.</td> </tr> </table>	<u>BNAHNDL</u>	Program is a BNA handler.	<u>COMPILER</u>	Program generates object code.	<u>COMPSHAR</u>	Program that generates object code, executed as a timesharing task.	<u>COPY</u>	Program is SYSTEM/COPY intrinsic.	<u>DMPALL</u>	Program is DMPALL intrinsic.	<u>DMSPROG</u>	Program is a DMSII control program (for example, DBP).	<u>DSKOUT</u>	Program is DSKOUT intrinsic or diskpack SQUASH intrinsic.	<u>EXECUTE</u>	Program is executing.	<u>HANDLERS</u>	Program is any type of handler (BNA, WFL, time sharing).	<u>LOADMP</u>	Program is LOADMP intrinsic.	<u>MCS</u>	Program is Message Control System (MCS).	<u>PACKUP</u>	Program is PACKUP intrinsic.	<u>SHAREHNDL</u>	Program is a timesharing handler program (for example, CANDE).	<u>SORT</u>	Program is SORT: or SORT. intrinsic.
<u>BNAHNDL</u>	Program is a BNA handler.																												
<u>COMPILER</u>	Program generates object code.																												
<u>COMPSHAR</u>	Program that generates object code, executed as a timesharing task.																												
<u>COPY</u>	Program is SYSTEM/COPY intrinsic.																												
<u>DMPALL</u>	Program is DMPALL intrinsic.																												
<u>DMSPROG</u>	Program is a DMSII control program (for example, DBP).																												
<u>DSKOUT</u>	Program is DSKOUT intrinsic or diskpack SQUASH intrinsic.																												
<u>EXECUTE</u>	Program is executing.																												
<u>HANDLERS</u>	Program is any type of handler (BNA, WFL, time sharing).																												
<u>LOADMP</u>	Program is LOADMP intrinsic.																												
<u>MCS</u>	Program is Message Control System (MCS).																												
<u>PACKUP</u>	Program is PACKUP intrinsic.																												
<u>SHAREHNDL</u>	Program is a timesharing handler program (for example, CANDE).																												
<u>SORT</u>	Program is SORT: or SORT. intrinsic.																												

TIMESHARE Program is executed as time sharing task.

WFLHNDL Program is a Work Flow Language (WFL) handler.

UTILITY Program is any type of intrinsic (DMPALL, LOADMP, PACKUP, DSKOUT, SYSTEM/COPY).

CHARGE <charge number>

Selects jobs based on the charge number the job is executed under. You can enter as many as 10 charge number selections and 10 charge number exceptions.

You can also select jobs based on part of a charge number by masking the charge number using the equal sign as a masking character. Use an equal sign for any unknown characters in the name. For example, the following command displays jobs executed under a charge number beginning with the characters 66, but does not display jobs executed under charge numbers 660001 or 669999.

MX CH 66= EXCEPT CH 660001, 669999

WAIT <waiting condition>

Selects jobs that are waiting for a specific condition to be satisfied (for example, waiting for hardware, waiting for memory and so forth). You can enter as many as three waiting conditions and three waiting condition exceptions.

The available types of waiting conditions are:

DEVICE Waiting for hardware device to become available

EVENT Waiting for a software event to be signaled (for example, STOQUE, CRCR, and so forth)

OPERATOR Waiting for operator to perform an action

SPACE Waiting for memory, disk space or diskpack space

EXCEPT

All text following the keyword EXCEPT is treated as exceptions to the selections entered previously. You must group all of the exceptions together after the keyword EXCEPT; you cannot enter a selection and an exception followed by another selection.

See also:

- The AJ (Display Active Jobs) command on Page 2-14.
- The WJ (Display Waiting Jobs) command on Page 2-411.
- The WS (Display Schedule) command on Page 2-423.

NET (Network Control Command)

Keyboard Input Message

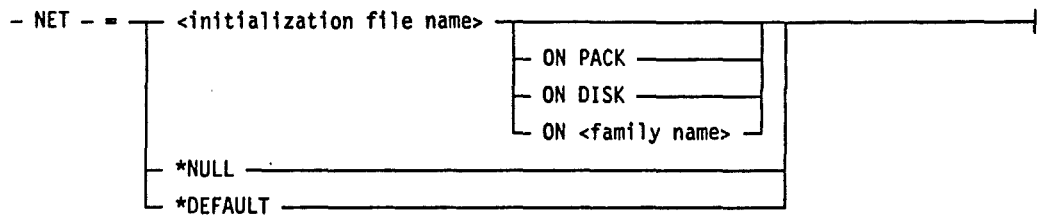
Access Level: 8

The NET command displays the status of a BNA network, or changes the mode of the local system to or from the Network mode.

The following figure shows the syntax used to display network status.

-- NET -----|

The following figure shows the syntax used to change the network configuration information used for any subsequent network initialization.



Explanation

<initialization file name> This designates the specified file as the next initialization file. The next initialization file is used for any subsequent initialization of the network, unless an initialization file is specified when the network is initialized.

ON PACK
ON DISK
ON <family name>

These give the location of the specified initialization file.

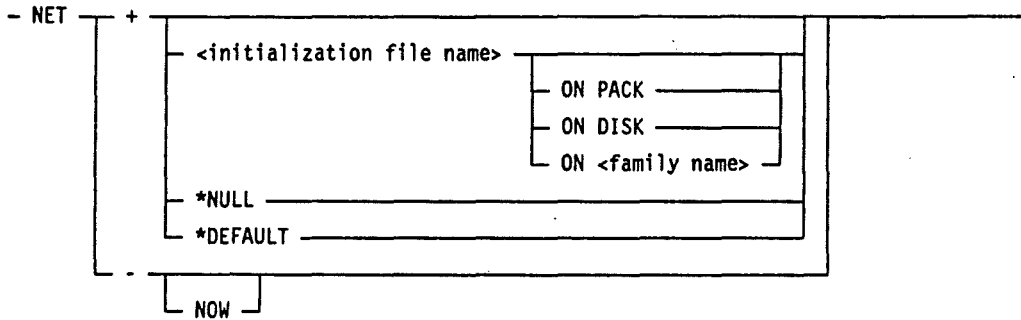
***NULL**

This instructs the system to have no next initialization file. When the network is brought up, the operator must enter BNA commands from the ODT.

***DEFAULT**

This makes the file BNAINI on diskpack the next initialization file.

The following figure shows the syntax used to change the network mode.



Explanation

- + This puts the local system into Network mode, using the BNA commands in the next initialization file.
- + <initialization file name> This puts the local system into Network mode, using the BNA commands in the named initialization file.
- ON PACK
ON DISK
ON <family name> These give the location of the specified initialization file.
- *NULL This instructs the system not to use any initialization file. When the network is brought up, the operator must enter BNA commands from the ODT.
- *DEFAULT This instructs the system to use the default initialization file. When the network is brought up, the system uses the file BNAINI on diskpack.
- This puts the local system in isolated mode, taking it out of the network. This takes effect after all subports that were in communication with remote ports finish their dialogs and are deactivated.
- NOW This puts the local system in isolated mode immediately, deactivating all subports that were in communication with remote subports without waiting for dialogs to finish.

Details

An initialization file, or *init* file, contains a series of BNA commands that bring the local system into the network.

At any one time, one init file is designated as the *next init file*. When a NET + command is done, the BNA commands for bringing up the network are taken from the current *next init file*.

Examples

NET +

This puts the local system in Network mode, using the next initialization file.

NET + INITFI ON DISK

This puts the local system in Network mode, using the initialization file INITFI on disk.

See also:

The V Series BNA Installation and Operations Guide.

NL (Resume at New Location)

Keyboard Input Message

Access Level: 3

The NL command resumes the processing of a program that has been suspended (awaiting a DS or DP command).

- <mix number> - NL - <program address> _____

Explanation

<mix number>	This designates the program that is to be resumed. You cannot use a mix number of 0 or 1.
<program address>	This is the base-relative address at which point the processing of the program is to resume. It must be an even value within the address range of the suspended program.

Caution

This command should be used only when the reason for suspension of the program has been thoroughly investigated. The MCP has no means to determine whether the new location is acceptable or other possible conflicts. The MCP can determine only that the address is within the base and limit of the program.

Example

10 NL 15532

This resumes processing of the program associated with mix number 10 at program address 15532.

See also:

- The ST (Suspend Program Processing) command on Page 2-364.
- The STOP (Suspend Program Processing) command on Page 2-367.
- The USE TERM and USE TRMD records in the *V Series Systems Operations Guide Volume 1: Installation*.

NT (Terminate Trace)

Keyboard Input Message

Access Level: 3

The NT command manually stops the instruction trace of a program.

- <mix number> - NT

Explanation

<mix number> This is the mix number of the program being traced.

Details

Instruction traces can be started with a GT (Initiate Trace) command if the program is executing, with a TRACE (Initiate Trace) command prior to a program's execution, or programmatically through a TRACE branch communicate (BCT). The TRACE BCT can also be used to terminate a program trace.

Example

33 NT

This stops the tracing of the program with mix number 033.

See also:

- The GT (Initiate Trace) command on Page 2-183.
- The TRACE (Initiate Trace) command on Page 2-381.
- The *V Series Program Interfaces Programming Reference Manual* and the various program language manuals for information about TRACE.

NW (Networking Command Prefix)

Keyboard Input Message

Access Level: 8

Most BNA network architecture commands must be preceded by the NW command.

- NW - <networking command> _____|

Explanation

<networking command> See the *V Series BNA Installation and Operations Guide* for full details on BNA commands.

Exceptions

The BNA commands that cannot be preceded by NW are

- AT
- CA
- HN
- NET

Example

NW AWAITCALL BY JMSTATN

This places the specified station called JMSTATN into a state to receive a call.

See also:

The *V Series BNA Installation and Operations Guide*.

OF (Optional File Indication)

Keyboard Input Message

Access Level: 1

The OF command is used to respond to a "NO FILE", "FILE NOT ON DSK", or "FILE NOT ON DPK" message. It causes the waiting program to continue without the file. The requested file must have been declared an optional file in the program.

-- <mix number> -- OF

Explanation

<mix number> This is the mix number of the program you wish to continue executing.

Use

This command is used when all the following conditions are true:

- A job suspends its execution and waits because a file it needs is missing. This condition produces a "NO FILE", "FILE NOT ON DSK", or "FILE NOT ON DPK" message on the ODT.
- The missing file is one that has been declared OPTIONAL in the program, meaning that it is not needed every time the program runs.
- The OF command causes the program to continue executing, ignoring the missing file.

Example

12 OF

This informs the MCP that the optional file being sought can be ignored and the program with mix number 012 can continue to execute.

See also:

- The DP (Dump and Discontinue) command on Page 2-144.
- The DS (Discontinue Program) command on Page 2-148.
- The IL (Assign Labeled File) command on Page 2-192.
- The OK (Continue Processing Suspended Program) command on Page 2-246.
- The UL (Assign Unlabeled File) command on Page 2-383.
- The various programming language reference manuals for information about declaring optional files.

OK (Continue Processing Suspended Program)

Keyboard Input Message

Access Level: 1

The OK command directs the MCP to continue processing a suspended program. The program might be suspended because of a lack of resources or a resource conflict.

The OK command can be used after the operator has resolved a NO USER DSK condition. In most cases, the MCP automatically recognizes that the program need no longer wait and resumes processing without operator intervention.

Do not initiate the OK command until the problem has been resolved.

```

- <mix number> - OK _____|
  | EOR _____|
  | EOF _____|
  | <subsystem number> _____|

```

Explanation

<mix number>

This is the mix number of the suspended program.

EOR
EOF

This can be used when an input file on magnetic tape has no ending label or has an irrecoverable parity error on the ending label.

The EOR option sends the job an "End-Of-Reel" message and causes the job to continue.

The EOF option sends the job an "End-Of File" message and causes the job to continue.

<subsystem number>

This can be used only if the specified program is in a NO USER DSK condition because insufficient space is available on the disk subsystem indicated in the ODT message. The operator can cause the remainder of the file to be allocated on a different subsystem by including the <subsystem number>, which can have a value from 0 through 8. A value of 0 indicates the default subsystem.

Examples

3 OK

This directs the MCP to continue processing the program with mix number 003.

35 OK 2

This directs the MCP to continue processing the program with mix number 35, which lacks sufficient space on its disk subsystem. The MCP will try to allocate the needed file space on subsystem 2.

18 OK EOR

This directs the MCP to continue processing the program with mix number 018 by sending the program an "End-Of-Reel" message.

See also:

- The FR (Designate Final Reel) command on Page 2-179.
- The IL (Assign Labeled File) command on Page 2-192.
- The OU (Direct File to Device) command on Page 2-257.
- The RM (Remove Duplicate File) command on Page 2-322.
- The UL (Assign Unlabeled File) command on Page 2-383.

OL (Display Peripheral Status)

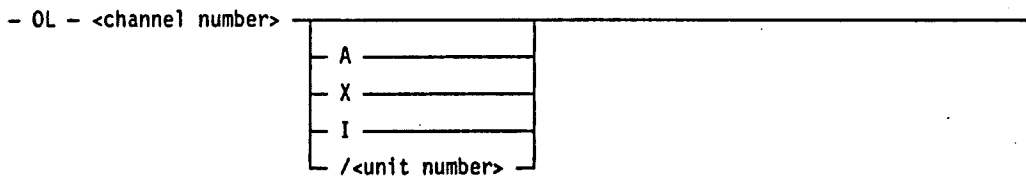
Keyboard Input Message

Access Level: 0

The OL command displays the status of a channel, peripheral unit or group of peripheral units, or pseudo card readers.

Channels

You can display the status of a channel by using the syntax shown in the following figure.



Explanation

- `<channel number>` This is the number of the channel whose status will be displayed. If only the `<channel number>` is supplied, the MCP displays the status of that channel.
- `A` This displays the status of all units on the specified channel.
- `X` This displays the status of all unassigned units on the specified channel.
- `I` This displays the status of all assigned units in use on the specified channel.
- `/ <unit number>` This obtains the status of a specific unit.

Examples

OL 3/2

This displays the status of channel 3, unit number 2.

OL 6A

This displays the status of all units declared for channel 6.

Peripheral Units

You can display information about a group of hardware devices by using the syntax shown in the figure which follows.

- OL - <hardware mnemonic>



Explanation

<hardware mnemonic> This is the mnemonic code for the hardware device whose status you want to display.

The following list shows the <hardware mnemonic>s permitted. These mnemonics resemble those specified in the UNIT record of the system configuration file. In addition, the identifier MTP causes the MCP to list the status of all magnetic tape units.

Hardware Mnemonic	Description
CRD	80-Column Card Reader
DCP	Data Communications Processor
DPK	Diskpack (Pack)
DSK	Disk
GCR	GCR Magnetic Tape Drive
IPP	Intelligent Laser Printing System
ISC	Inter-System Connect
MPE	Phase-Encoded (PE) Magnetic Tape Drive
MT9	Non-Return Zero Magnetic Tape Drive
MTC	Magnetic Cartridge Tape Drive
MTP	Magnetic Tape Drives (all)
NST	Non-Status Device
OCS	Operator Control Station
ODT	Operator Display Terminal
PCH	80-Column Card Punch
PCR	Pseudo Card Reader
PRN	Buffered Printer
RJE	Remote Job Entry Device
S4A	9137 Type Reader Sorter

Hardware Mnemonic	Description
S4B	9138 Type Reader Sorter
TC5	TC 500 Device
TWX	Teletypewriter through a UNILINE DLP
TPR	Train Printer
VDD	B 20 or ET Series Workstation or a B 9352 or TD 800 Series Terminal

X This displays the status of all unassigned units of the designated <hardware mnemonic>.

I This displays the status of the units of the designated <hardware mnemonic> that are being used.

Magnetic Tape

If you try to display the status of a magnetic tape unit that is in use, you will receive one of the following messages:

IN USE (I)	The tape unit is opened input
IN USE (O)	The tape unit is opened output
IN USE	The tape unit is closed and still attached to the job

If a tape unit is GCR/PE capable, the response to an OL MTP command also displays the density of the tape. For example:

```
06/1 GCR (1600) Y85W4/FILE 8 RL# 001 (I) SYSTEM/COPY=41
```

The density of the tape is 1600.

Disk and Diskpack

The OL command for disk and diskpack can be OL DPK, OL DSK, or OL <cc/u>. The first two commands show information for all disks or diskpacks; the latter command displays information for a particular device.

A typical system response to an OL command used to display disk or diskpack status is as follows:

```
34/3 DPK L/680163 INDEX 1 ID 123 (680S) SHRD R RESERVED WLO
. . . 1651644 SEGS AVAIL, 0.2% I/O - IN USE BY 2 FILES
```

The following describes each of the items shown in the preceding example.

Element in Sample Response	Description
34/3	This indicates the channel number (34) and unit number (3) of the device.
DPK	This indicates that the device is a diskpack. If the device is a disk, the mnemonic DSK is displayed instead of DPK.
L/680163	This is the pack name (L) and pack serial number (680163).
INDEX 1	This indicates that this diskpack is a Version 2 diskpack and is part of a family. The index is the family index of the pack within the family. A family index of 1 indicates that the pack is the base pack of the family.
ID 123	This is the ID number of the device.
(680S)	This indicates the type of device. In this example, the device is a Memorex 3680 sequential mode pack. A response of "UNKN" (unknown) indicates that the device has not been successfully classified.
SHRD	This indicates that the device is a shared device.
R	This indicates that the pack is restricted.
RESERVED	This indicates that the pack is currently reserved.
WLO	This indicates that Write Lockout has been enabled for this device.
1651644 SEGS AVAIL	This indicates the total number of disk segments available.

Element in Sample Response	Description
0.2% I/O	This is an estimate of the percentage of time that the device is in use. It is recomputed every 10 seconds in the following manner: the elapsed times between the execution of each Initiate I/O (IIO) instruction and the completion of that I/O are totalled, giving the approximate total time that the unit is in use during the 10-second period; the total time is then expressed as a percentage of a 10-second interval.
IN USE BY 2 FILES	This indicates the number of file areas open on the device, not necessarily the number of files.

Note: The SEGS AVAIL is the total space available on the device; it does not take into account disk fragmentation.

Examples

OL MTP

This displays the status of all magnetic tape units.

OL SOR I

This displays the status of the in-use or assigned sorter units.

OL CRD X

This displays the status of all unassigned card readers.

Network Communication Ports

You can display status information on any Network Communication Ports (NCPs) on the system. NCPs are used as part of the V Series Communication System (VCS). There is no special syntax for the OL NCP command. The response to the OL NCP command is shown in the figure that follows.

```

      ①          ②          ③          ④
  Ⓐ 16/0 NCP CP3680 0010 IN USE NCP-IR=0100

      ⑤          ⑥          ⑦          ⑧
  Ⓑ CP   0010   CP3682          RDY    I/O

      ⑨ READ:    3.7 MSG/S    0.31 KB/S  WRITE:    3.7 MSG/S    0.61 KB/S

  Ⓒ NODE 0001   HOSTC          NOT RDY  I/O

      READ:    0.0 MSG/S    0.00 KB/S  WRITE:    0.0 MSG/S    0.00 KB/S

```

Figure 2-1 . System Response to the OL NCP Command

Entries in the OL NCP response are displayed in the order in which items are declared in the VCS initialization file.

There are three types of entries in the response: NCP entry (labeled A in Figure 2-1), CP entry (labeled B in Figure 2-1), and node entry (labeled C in Figure 2-1). NCP entries represent the NCPs connected to the system. Communication Processor (CP) entries represent hardware devices with physical connections to individual stations. Currently, there is one CP entry for each NCP entry. Node entries represent other host systems that can be connected to the NCP. There can be several node entries for each NCP entry. Each of the circled numbers in Figure 2-1 corresponds to an entry in the following table.

Number	Description
①	This is the channel number and unit number declared for the NCP.
②	This a mnemonic representing the type of NCP connected to the channel. Current NCP types include CP3680 and TELCOM.
③	This a 4-digit number assigned to the NCP in the VCS initialization file. This number is declared on the UNIT NCP record in the system configuration file.
④	This is the status, name and task number of the Independent Runner (IR) responsible for serving the NCP. This item is not displayed if VCS is not currently running on the system.

Number	Description
⑤	This is a 4-digit number assigned to the CP in the VCS initialization file.
⑥	This is a 20-character name assigned to the CP in the VCS initialization file.
⑦	This indicates whether the CP is ready.
⑧	This indicates what input/output mode the CP is currently enabled for. Possible values include I/O, IN, and OUT.
⑨	This is CP status information showing the messages per second and the kilobytes per second transmitted through the CP for both read and write operations. The averages are calculated when the OL NCP command is entered, based on the period of time since the last MCP status operation (status interval 40 seconds).

Details

To calculate the total bandpass of an NCP, add the read and write transfer information for all CPs and nodes connected to the NCP. The result can be compared to the maximum bandpass of the NCP for performance capacity planning.

See also:

- The UNIT (Add Peripheral Devices) command on Page 2-385.
- NCP initialization information in the *V Series Communication System Installation Reference Manual*.

Pseudo Card Readers

You can display the active pseudo card readers on the system by using the syntax shown in the following figure.



Explanation

- PCR** This displays the status of all pseudo card readers if the optional elements are not specified.
- <deck number>** This displays the status of only the indicated pseudo card deck.
- <file identifier>** This displays the status of all pseudo card readers that contain a deck of the specified name.

Details

Use the CD, CDP (List Inactive Pseudo Decks) command to display inactive pseudo decks.

If you want to deactivate a pseudo card deck, use the DA, DAP (Deactivate Pseudo Reader) command.

Examples

```
OL PCR CARDFL
```

This displays the status of all pseudo card readers that contain a deck with the name CARDFL.

```
OL PCR 3
```

This displays the status of pseudo card readers with active deck number 3.

See also:

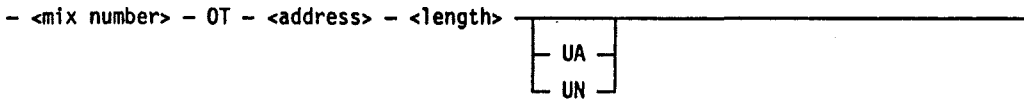
- The CD, CDP (List Inactive Pseudo Decks) commands on Page 2-68.
- The DA, DAP (Deactivate Pseudo Reader) commands on Page 2-103.
- The SHOW (Display System Information) command on Page 2-343.

OT (Display Program Data)

Keyboard Input Message

Access Level: 3

The OT command displays on the ODT up to 30 characters of a job's memory.



Explanation

<mix number> This specifies the mix number of the program from which the data is displayed. You cannot use a mix number of 0 or 1, since these are reserved for the use of the operating system.

<address> This designates the program base-relative location at which the data begins.

<length> This specifies the number of bytes (for UA data) or digits (for UN data) to be displayed. The length of the displayed data can range from 1 through 30.

UA UN These indicate data class.

UA

UN UA denotes unsigned alphanumeric, which causes data to be displayed as characters with the <length> in bytes. (One byte is 8 bits.)

UN denotes unsigned numeric, which causes data to be displayed as a string of digits with the <length> in digits. (One digit is 4 bits.)

Example

11 OT 38 2 UN

This displays a field that is two unsigned numeric digits long and begins at base-relative address 38 from the program with mix number 011.

OU (Direct File to Device)

Keyboard Input Message

Access Level: 5

The OU command routes an output file to a designated device. Within limits, this command can be used to change the hardware medium for a file.

```
- <mix number> - OU - DK _____|
                   | PK _____|
                   | <cc/u> _____|
```

Explanation

<mix number>	This is the mix number of a program that requires an output device. The <mix number> cannot be 0 or 1, since these are reserved for use by the MCP.
DK	This routes the file to disk.
PK	This routes the file to diskpack.
<cc/u>	This routes the file to the specified channel number and unit number of a magnetic tape drive. The device specified by <cc/u> cannot be in a saved status.

Details

The OU command is normally used in response to a "<hardware mnemonic> REQD <file name> <job name> = <mix number>" message that has been generated because of the following:

- A lack of available resources
- System option settings that do not permit automatic file assignment

The system options that affect file assignment, such as designating the medium where printer backup files will be directed, include APBD, APCR, PBD, PBP, PBPS, PBT, PCD, PCP, and PRN. These options can be set through the USE records of the system configuration file or with the SO (Set System Options) command. These options can be reset with the RO (Reset System Options) command.

If the PBT option is set, a file directed to a GCR (Group Coded Recording) tape drive will be written in GCR mode if a scratch tape is mounted. If the tape mounted is not a scratch tape, the already established mode and density are used. This does not apply to MTC tape drives.

Note: *The OU command cannot be used to direct portions of a file to different pack families. Once a file is opened on a family, all subsequent portions of the file must be placed on the same family.*

Examples

15 OU 3/2

This sends an output file with mix number 015 to the tape drive with channel number 3, unit number 2.

20 OU DK

This sends an output file with mix number 020 to disk.

See also:

- The FM (Direct File to Device—Special Forms Response) command on Page 2-175.
- The RO (Reset System Options) command on Page 2-327.
- The SO (Set System Options) command on Page 2-352.
- The USE records in the *V Series Systems Operations Guide Volume 1: Installation*.

PA (Analyze Pack Directory)

Keyboard Input Message

Access Level: 5

The PA command invokes the intrinsic program DSKOUT to generate an analysis of all files found in the diskpack directory. An analysis of the following is given:

- Area addresses
- File identifier
- Total number of segments used
- Logical record size
- Records per block
- Area size
- Area numbers, total areas used, and maximum areas declared
- End of file pointer
- File attributes

You can specify a single file ID or group file ID, and the system analyzes only the relevant file or files.

```

- PA _____ <cc/u> _____
   |_____| |_____| |_____|
   NOSTANDBY <family name> <file ID>
                                     <masked file ID>
  
```

Explanation

NOSTANDBY

This directs the operating system to capture information for DSKOUT even when the operating system *cannot* perform its normal standby function. The standby function is still attempted, but information is captured regardless of whether the function is successful. For more information about the standby function, see "Details" on the following pages.

<cc/u>

This is the channel number and unit number of the diskpack to be analyzed.

<family name>

This is the family name of the pack family on which an analysis will be performed. All files resident on the specified family are analyzed. This option is valid only for Version 2 packs. For more information on pack family versions, refer to "Pack Subsystems" in the *V Series Systems Operations Guide Volume 3*.

=
<file ID>
<masked file ID>

This is the file identifier or masked file identifier of files on which an analysis will be performed. If an equal sign (=) is specified, then an analysis is performed on all files on the family.

Output Options

Output options that provide alternate sorting and analysis are available. See the KA (Analyze Disk Directory) command for a full description of these options and their applications.

Details

When you enter a PA command, the operating system can respond with the following message:

```
KBD IGNORED: DISK OR PACK MAINT IS IN PROGRESS
```

This means the operating system cannot perform a standby function as part of the PA command. A standby function suspends updates to the diskpack directory and other data structures long enough to obtain an accurate "snapshot" of the structures. This "snapshot" is passed to DSKOUT for analysis. The operating system needs exclusive access to the diskpack directory for an accurate DSKOUT analysis.

To respond to this situation, you can do either of the following:

- Wait until a later time when disk activity is less.
- Use the NOSTANDBY option of the PA command.

The NOSTANDBY option captures information for DSKOUT even when the operating system *cannot* perform the standby function. The standby function is attempted, but information is captured regardless of whether it is successful.

Using NOSTANDBY can create missing or duplicate disk messages that are not correct. This happens if an update occurs as the "snapshot" is taken. If NOSTANDBY is used and missing or duplicate disk conditions are encountered, DSKOUT flags the resulting messages with NOSTANDBY warnings.

These warnings mean the condition might be caused by non-exclusive access rather than by an actual missing or duplicate disk. Before you act on any missing or duplicate disk messages with a NOSTANDBY warning, enter another PA command without NOSTANDBY. Enter the command when less disk activity is taking place.

Examples

PA 10/12 ABC=D=

This produces an analysis of all files with the masked file ID ABC=D= on the diskpack located on channel number 10, unit 12.

PA 3/1 AFILE

This produces an analysis of AFILE on channel number 3, unit 1.

PA NOSTANDBY USERPK =

This produces an analysis of all files on the family named USERPK. The analysis is generated even if the normal standby function of the PA command is unsuccessful.

See also:

- The KA (Analyze Disk Directory) command on Page 2-201.
- The KP (Print Disk Segments) command on Page 2-205.
- The KS (Analyze Disk Space) command on Page 2-207.
- The PP (Print Pack Sectors) command on Page 2-287.
- The PS (Analyze Pack Space) command on Page 2-303.

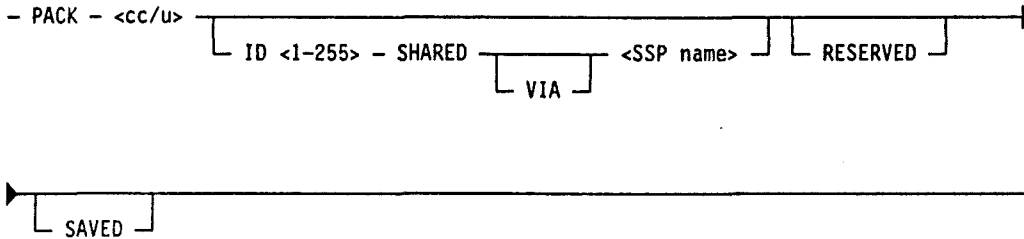
PACK (Add a Pack)

MCP Control Instruction

Access Level: 8

The PACK command adds a diskpack to the system environment.

The syntax for the PACK command is identical to that for the PACK record of the system configuration file.



Explanation

- | | |
|--|---|
| <cc/u> | This is the channel number and unit number of the diskpack to be added. |
| ID <1-255> | This assigns an ID number to the diskpack. A diskpack ID number can range from 1 through 255. Do not use the same ID number twice on a given system. Be sure that each system in a multi-system shared configuration uses the same ID number to reference the drive. |
| SHARED <SSP name>
SHARED VIA <SSP name> | This indicates the name of the SSP that connects the system to other systems in a multi-system shared configuration. |
| RESERVED | This causes the device to be initially reserved when the system is cold started or at subsequent halt/loads. Reserved status can be altered using the UR (Inhibit or Uninhibit Unit) command. |
| SAVED | This causes the device to be initially saved when the system is cold started or at subsequent halt/loads. The device must be readied with the RY (Ready Peripheral Device) command in order to access it. Note that the RY command does not alter the initial saved status of the diskpack. |

Example

PACK 05/2 SAVED

This adds a diskpack to the system environment on channel 5, unit number 2. When the system is initialized the diskpack will be in a saved status until readied with an RY command.

See also:

- The DISK (Add a Disk) command on Page 2-128.
- The UNIT (Add Peripheral Devices) command on Page 2-385.
- The PACK record in *V Series Systems Operations Guide Volume 1: Installation*.

PATCH (Patch File)

MCP Control Instruction

Access Level: 5

The PATCH command overwrites a specified section of data with new data. The area can be in an MCP code file, a program code file, in a disk or diskpack file, or at a known sector on a disk.

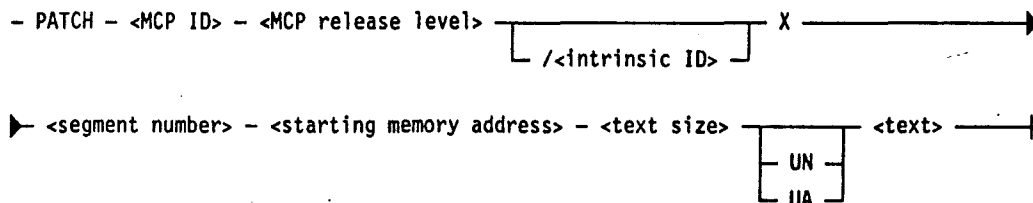
Note that MCP code files can reside only on disk. Any attempt to patch such a file on diskpack will result in a syntax error.

Patches to files on diskpack will be allowed only if the relevant area resides on the base diskpack. Continuation diskpack areas cannot be patched or displayed.

Note: The text given must be the specified length or a syntax error will result. If UN <text> is specified, then only numbers and the hexadecimal characters 0 through F are allowed. If UA <text> is specified, and any special characters are present, the entire string must be enclosed in quotation marks. All patches are logged in the RLOG.

Patching a Running MCP Code File

The following syntax diagram shows the syntax used to patch the running MCP code file.



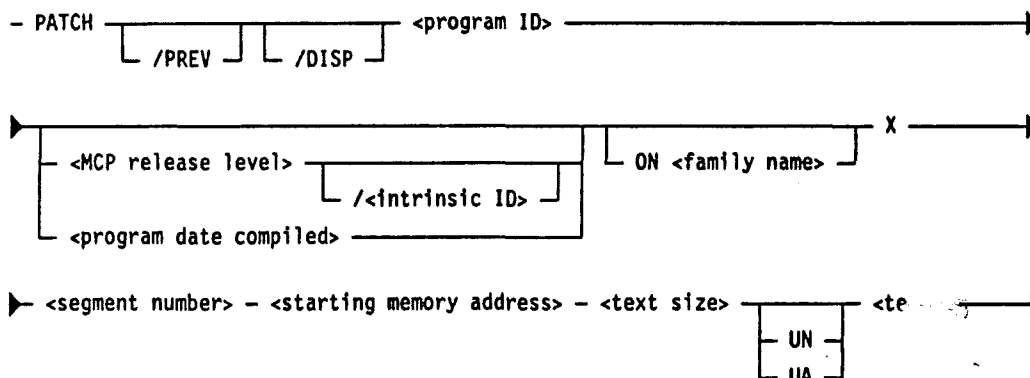
Explanation

<MCP ID>	This specifies the ID of the running MCP code file.
<MCP release level>	This is the MCP/VS release number.
/<intrinsic ID>	This specifies the name of an MCP intrinsic to be patched.
X <segment number>	This is the actual segment where the data is located. Valid numbers are 0 through 999.
<starting memory address>	This is the actual memory address within the segment. This term must be decimal digits.

<text size>	This is the amount of data to be patched. The maximum size is 30 digits or bytes.
UN UA	This indicates the format of the data to be patched. UN denotes numeric, UA denotes alphanumeric. If the text is declared as UN, the record can contain only 0 through 9 and A through F. The default is UN.
<text>	This is the data to be placed in the MCP or MCP intrinsic.

Patching a Program or a Non-Running Code File

The following syntax diagram shows the syntax used to patch a program or a non-running code file.



Explanation

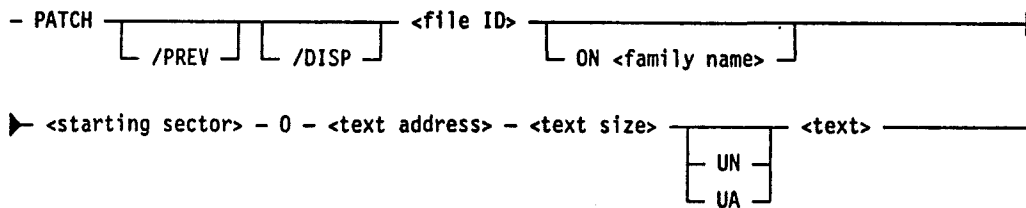
/PREV	This displays the data present at the location specified before the patch was applied.
/DISP	This displays the data after the patch is applied to ensure correctness.
<program ID>	This is the name of the program to be patched.
<MCP release level>	This is the MCP/VS release number.
/<intrinsic ID>	This is the name of an MCP intrinsic to be patched. The <intrinsic ID> is valid only if <program id> specifies MCP.

PATCH

- <program date compiled> This is the date the program was compiled. The format of the date is Dmmddy, where the following is true:
- | | |
|----|---|
| D | This is distinguishes the date specification. |
| mm | This is the month expressed as a two digit number (01 for January, 02 for February, and so on). |
| dd | This is the day of the month expressed as a two digit number. |
| yy | This is the last two digits of the year. |
- ON <family name> This is the name of the family where the program is located. A family name of DISK is the same as omitting the ON <family name> clause.
- X <segment number> This is the actual segment where the data is located. Valid numbers are 0 through 999.
- <starting memory address> This is the actual memory address within the segment. This term must be decimal digits.
- <text size> This is the amount of data to be patched. The maximum size is 30 digits or bytes.
- UN
UA This indicates the format of the data to be patched. UN denotes numeric, UA denotes alphanumeric. If the text is declared as UN, the record can contain only 0 through 9 and A through F.
- The default is UN.
- <text> This is the data to be placed in the program.

Patching Disk or Diskpack Files

The following syntax diagram shows the syntax used to patch a file on disk or diskpack.

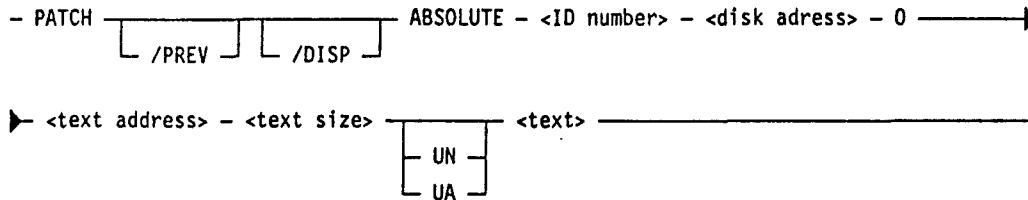


Explanation

/PREV	This displays the data present at the location specified before the patch was applied.
/DISP	This displays the data after the patch is applied to ensure correctness.
<file ID>	This is the name of the file to be patched.
<MCP release level>	This is the MCP/VS release number.
ON <family name>	This is the name of the diskpack where the file is located.
<starting sector>	This is the sector address of the start of the patch location within the file.
0	This separates the <starting sector> value from the <text address> value.
<text address>	This must be less than 200 for disk, or 360 for diskpack. This address is always expressed in digits, relative to zero. The text address causes zero-relative digit displacement from the beginning of the sector.
<text size>	This is the amount of data to be patched. The maximum size is 30 digits or bytes.
UN UA	This indicates the format of the data to be patched. UN denotes numeric, UA denotes alphanumeric. If the text is declared as UN, the record can contain only 0 through 9 and A through F. The default is UN.
<text>	This is the data to be placed in the file.

Patching Disk Sectors

The following syntax diagram shows the syntax used to patch disk sectors.



Explanation

- | | |
|-------------------------------------|---|
| /PREV | This displays the data present at the location specified before the patch was applied. |
| /DISP | This displays the data after the patch is applied to ensure correctness. |
| ABSOLUTE <ID number> <disk address> | This specifies the disk ID number of the device, and the physical disk address. The disk ID number can be any number from 1 through 255. The disk address is an address of 1 to 12 digits and must be expressed in decimal. |
| 0 | This separates the <disk address> value from the <text address> value. |
| <text address> | This must be less than 200 for disk, or 360 for diskpack. This address is always expressed in digits, relative to zero.

The text address causes zero-relative digit displacement from the beginning of the sector. |
| <text size> | This is the amount of data to be patched. The maximum size is 30 digits or bytes. |
| UN
UA | This indicates the format of the data to be patched. UN denotes numeric, UA denotes alphanumeric. If the text is declared as UN, the record can contain only 0 through 9 and A through F.

The default is UN. |
| <text> | This is the data to be placed on the disk or diskpack. |

Example

```
PATCH /DISP ABSOLUTE 4 367299 0 100 5 UN 33251
```

This patches a disk sector on the disk with ID number 4 at disk address 367299 with UN data 33251 and displays the data after the patch is applied.

Verifying Patch Records

You can verify patch records by using the PATCHK program. This applies to patching running MCP code files, program and non-running MCP code files, and patching disk or diskpack files.

To utilize PATCHK, you must create patch records on cards or on card images through an online editor such as CANDE or DMPALL/CTLDCK.

Records consist of 80 bytes; the blocking factor can be any value. Only the first 72 bytes of text will be read. Do not enter an invalid punch in column 1, or a question mark in column 1 of a CANDE file that is to become a pseudo deck, until after PATCHK has verified the records.

Copy on the card the checksum provided to you by your Unisys customer representative. PATCHK reads the patch records and calculates a checksum. Errors are detected by comparing the two checksums. PATCHK prints a report listing the errors.

PATCHK uses only the following terms to calculate the checksum:

Explanation

<program ID> <MCP release level> <intrinsic ID>	If this term is enclosed in parentheses, the parentheses are not included in the checksum.
ON <family name>	This entire term is included in the checksum.
X <segment number>	PATCHK includes the X in the checksum. Leading zeros are ignored; for example, X05 is equal to X5. However, X00 is equal to X0.
<starting memory address>	Leading zeros are ignored; however, if this field is zero, zero will be included in the checksum.
<starting sector>	Leading zeros are ignored; however, if this field is zero, zero will be included in the checksum.
0	This literal is included in the checksum as a single digit, regardless of additional leading zeros.
<text address>	Leading zeros are ignored; however, if this field is zero, zero will be included in the checksum.

PATCH

- <text size>** Leading zeros are ignored. This field must be expressed in decimal digits and must be greater than or equal to 0 and less than 30. If the **<text size>** does not equal the actual length of the text, PATCHK flags the record as an error.
- UN** The term UA is accumulated in the checksum; the
UA term UN is not.
- <text>** All characters entered are included in the checksum, regardless of field type.
- If this field begins with a quotation mark, the field is converted to a string (which includes embedded spaces, quotation marks, and special characters) before the checksum is calculated. If the string is declared as UA, the record can contain only the characters 0 through 9 and A through F.
- If the length of the **<text>** is not equal to **<text size>**, the record is flagged as an error.

Executing the PATCHK Program

If the PATCHK records are in a CANDE file, you can execute PATCHK as follows:

```
- EX PATCHK ( <nnnn00>,"<xxxx>" _____ ) _____  
                |_____|  
                |_____| <family name> |_____|  
                |_____| PACK |_____|  
                |_____| DISK |_____|
```

Explanation

- <xxxx>** This is the 4-character file ID of the CANDE file.
- <nnnn>** These are the first four digits of the CANDE usercode.

A card deck containing invalid punches in column 1 can be submitted to PATCHK as follows:

```
? EX PATCHK  
? FILE PCHCRD CTLDCK  
? DATA CTLDCK  
  <Patch Deck>  
? ENDCTL
```

The following is an example of a patch deck:

? .SORT: (19) MERGE PHASE RUN CONTROL FAILURE (011990)	T11877 734#001#
?PATCH MCP 2.10 /(SORT:) X042 14782 18 110808011196008672.53%	T11877 734 1-01
? .DIR I/O CLOSE PATCH	R68#004#
?PATCH MCP 2.10 X038 16128 26 05A5020008004D000000D00538.F2	R68 1-04
?PATCH MCP 2.10 X038 16154 20 0107075E055240D00538.FB	R68 2-04
?PATCH MCP 2.10 X038 16174 8 27006446.C3	R68 3-04
?PATCH MCP 2.10 X038 9920 8 27016128.C2%	R68 4-04
? .SORT: (27) NOT ENOUGH MEMORY (SORT BUFFERS)	736#002#
?PATCH MCP 2.10 /(SORT:) X042 14826 24 040606010938008590011196.D6	736 1-02
?PATCH MCP 2.10 /(SORT:) X042 14850 26 03060601119600867427014910.72%	736 2-02
? .MORE TAPESORT PROBLEMS	737#003#
?PATCH MCP 2.10 /(SORT:) X028 9886 6 013400.39	737 1-03
?PATCH MCP 2.10 /(SORT:) X033 10028 6 013400.B6	737 2-03
?PATCH MCP 2.10 /(SORT:) X035 12494 6 013400.29%	737 3-03

PATCHK Program Error Messages

These paragraphs list the PATCHK error messages and provide a sample patch record showing each error.

DATA AFTER 'ABSOLUTE' NOT NUMERIC

The ID number, disk address, text address, or text size contains a non-numeric character.

PATCH ABSOLUTE F 12000 0 400000 6 UA HAAAA

DATA NOT LENGTH SPECIFIED

The text entered in the patch record does not match the text length in the PATCH command.

PATCH MCP 2.02 X23 24610 6 32F00000.9F

DISK ADDRESS/SECTOR # NOT NUMERIC

The disk address to be patched contains a non-numeric character.

PATCH ABSOLUTE 1 120F0 0 400000 6 UA HAAAA

FILE NAME > 6 CHARS

The name of the program or operating system code file is longer than the maximum 6 characters.

PATCH MCPMCPMCP 2.02 X23 24610 8 32F00000.9F

PATCH

INVALID CHECKSUM CHK = 77

An incorrect character has been entered in the patch record.

PATCH MCP 2.02 X23 24610 8 32F00000.CE

MEMORY ADDRESS NOT NUMERIC

The starting memory address contains a non-numeric character.

PATCH MCP 2.02 X23 2461D 8 32F00000.9F

MEMORY AREA> 99

The memory area subscript of the segment number is greater than 99.

PATCH MCP 2.02 X23.120 24610 8 32F000000.9F

MEMORY AREA NOT NUMERIC

The memory area subscript of the segment number contains a non-numeric character.

PATCH MCP 2.02 X23.F 24610 8 32F000000.9F

MEMORY AREA NOT PRESENT

The memory area subscript of the segment number is missing.

PATCH MCP 2.02 X23. 24610 8 32F00000.9F

RELEASE LEVEL > 4 CHARS

The release level of the operating system code file is longer than the maximum 4 characters.

PATCH MCP 2.021 X23 24610 8 32F00000.9F

SEGMENT NO. LENGTH > 3 OR < 1

The segment number of a program or operating system code file.

PATCH MCP 2.02 X2345 24610 8 32F00000.9F

TEXT LENGTH <1 OR > 30

The text size entered is not within the limits allowed by the PATCH command. Usually the text size is too large.

PATCH MCP 2.02 X23 24610 124 32F00000.9F

TEXT LENGTH NOT NUMERIC

The text size contains a non-numeric character.

PATCH MCP 2.02 X23 24610 A 32F00000.9F

'UA' NOT SPECIFIED

The text to be patched contains a non-numeric character, but the PATCH command does not specify UA data.

PATCH MCP 2.02 X23 24610 8 _ 32X00000.9F

UNKNOWN DATA AFTER '/'

The PATCH option entered after the slash is not a valid PATCH option.

PATCH/PREB MCP 2.02 X23 24610 8 32F00000.9F

ZERO LITERAL NOT PRESENT

The character 0 (zero) is missing from a PATCH command that patches a file or a disk sector.

PATCH GUESS 12000 _ 400000 6 UA HAHAHA

See also:

The PATCH record in the *V Series Systems Operations Guide Volume 1: Installation*.

PB (Print Backup File)

Keyboard Input Message

Access Level: 1

The PB command starts a user-provided program to print backup files. However, it is recommended that you use the PRINT (Print Files) command instead of the PB command.

If the printer backup file resides on tape, the program PBTOUT is scheduled. If the file resides on disk or diskpack, PBDOUT is scheduled. The user must supply the PBTOUT and PBDOUT programs; Unisys does not supply them. If the appropriate program is not present on disk, the PB command is ignored.

See Appendix B for PB syntax.

See also:

- The PRINT (Print Files) command on Page 2-291.
- Appendix B.

PC, PCP (Create Punch Backup File)

Keyboard Input Message

Access Level: 1

The PC command schedules the execution of the PCHOUT utility, which creates punched output of disk or diskpack punch backup files.

You can specify either an MCP or user-coded version of PCHOUT.

Using the MCP Intrinsic

```

┌ PC ────────────┐ ┌───────────┐ ┌───────────┐ ┌───────────┐ ┌───────────┐
└ PCP ───────────┘ └ /MCP ─────────┘ └ SAVE ─────────┘ └ <copies> ─────────┘
<file number>

```

Explanation

PC PCP PC punches backup files from disk; PCP punches backup files from diskpack.

/MCP This specifies the MCP intrinsic PCHOUT.

If you specify /MCP, the bound MCP intrinsic is scheduled regardless of the presence of a user-provided PCHOUT on disk.

If you omit this option and there is a user-provided PCHOUT on disk, the user-provided PCHOUT will be executed.

<file number> This is the number of the punch backup file on disk or diskpack to be made into punched card output.

SAVE This indicates that the punch backup file is not to be removed after the punched output is created.

<copies> This indicates the number of copies of the punched output file you want. One copy is the default.

User-Coded PCHOUT

A user-coded version of PCHOUT can include parameter entries for PC and PCP that resemble the equal sign (=) and asterisk (*) parameters of the PRINT command. The MCP PCHOUT program ignores these parameters.

```

┌ PC ────────────┐ ┌───────────┐ ┌───────────┐ ┌───────────┐ ┌───────────┐ ┌───────────┐
└ PCP ───────────┘ └ /OWN ─────────┘ └ * ───────────┘ └ <file number> ─────────┘ └ SAVE ─────────┘ └ <copies> ─────────┘

```

Explanation

- /OWN** Using the /OWN option prevents the scheduling of the bound MCP intrinsic when a user-provided PCHOUT is required but is not on disk.
- If no program with this name is in the disk directory, the bound MCP intrinsic PCHOUT is scheduled.
- =** This indicates that all punch backup files on the specified medium will be punched.
- *** This informs the PCHOUT program that an AX response will provide further specifications, such as skipping lines or restarting, after operation begins.
- After locating the specified file, PCHOUT displays "ENTER SPECS" and issues an ACCEPT. The response to the ACCEPT must be in one of the formats shown under the PRINT (Print Files) command.

Details

Backup files created by WFL, CANDE, RJE, and BNA cannot be punched by the PC command and cannot be accessed by programs that open "@00000" and "*00000". This feature prevents files from being accidentally removed before they can be printed (RJE), viewed (CANDE), printed with the PRINT command (WFL job), or routed back to a remote host (BNA).

Examples

PC /MCP 20010 SAVE

This schedules the MCP bound intrinsic PCHOUT to create punched output of disk punch backup file number 20010. The punch backup file will be saved after the output has been created.

PCP /OWN 00104 SAVE 5; PRIORITY 5

This schedules a user-coded version of PCHOUT to create punched output of a diskpack punch file with the number 00104. The PCHOUT program has a memory, processing, I/O, and schedule priority of 5. The backup file will be saved after the output has been created. Five copies will be punched.

See also:

- The CV , CVP (Convert Punch Backup File to Pseudo Card Deck) command on Page 2-100.
- The RO (Reset System Options) command on Page 2-327.
- The SO (Set System Options) command on Page 2-352.
- The TO (Display System Option Status) command on Page 2-379.

- The USE PCH, PCD, and PCP records in the *V Series Systems Operations Guide Volume 1: Installation*.
- The *V Series MCP/VS Programming Reference Manual* for information about PCHOUT.

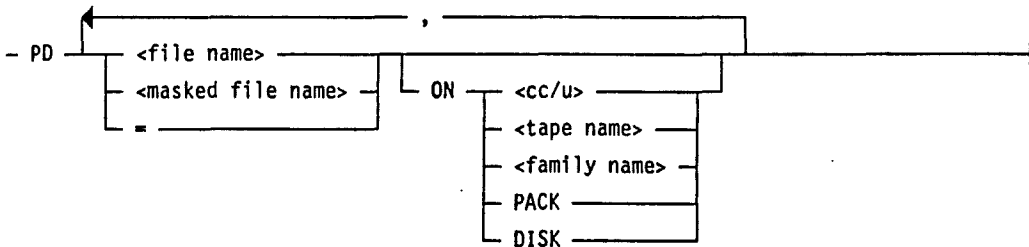
PD (Print Directory)

Keyboard Input Message

Access Level: 0

The PD command lists the names of files on disk, diskpack, or LOADMP/PACKUP tape.

This command reads directories; it does not look for the actual files.



Explanation

<file name>
<masked file name>
=

This indicates the name of the file. You can use a unique file name, a masked file name, or an equal sign (=). A single equal sign lists all files. A masked file name is a string of characters used to list one or more files. An equal sign (=) is used as a masking (wildcard) character. For example, A= lists all files beginning with A.

ON

If the ON clause is missing, or if the ON clause specifies DISK, the PD will be done on disk (100-byte media).

<cc/u>

This specifies the channel number and unit number of the diskpack or tape drive to be searched for files. Both base and continuation areas are included.

If a diskpack is specified using the channel number and unit number of the diskpack, the results returned will differ depending on the version level of the diskpack. For Version 2 diskpacks, the *-B* (for base), and the *-C* (for continuation) suffixes have been replaced by a count of the number of areas used by the file on the specified device. Also, if the specified diskpack is the base pack of a family and the specified file exists on the family but has no areas assigned to the specified diskpack, the system returns a value of 0 (zero) for the count of the number of areas used by the file.

Disk, diskpack, and tape PDs can be mixed in the same command. However, if no location is given, the default is disk.

Note: For more information on pack family versions, refer to "Pack Subsystems" in the *V Series Systems Operations Guide Volume 3*.

<tape name>	This specifies a tape that was not created by the SYSTEM/COPY utility. For tapes created using the SYSTEM/COPY utility, use the DIR (List File Directory) command.
<family name>	This causes the base area of all diskpacks with this name to be searched.
PACK	This indicates all unrestricted diskpacks.
DISK	This indicates disk.

Examples

PD A, B, C

This lists the files A, B, C contained in the disk directory.

PD = ON DISK

This lists all files in the disk directory.

PD A= ON 3/12

This lists all files beginning with A on channel number 3, unit number 12.

PD FILE01 ON SYSTEM

This lists a file called FILE01 contained in the directory on a diskpack called SYSTEM.

See also:

The DIR (List File Directory) command on Page 2-125.

PERFORM (Schedule DMPALL)

MCP Control Instruction

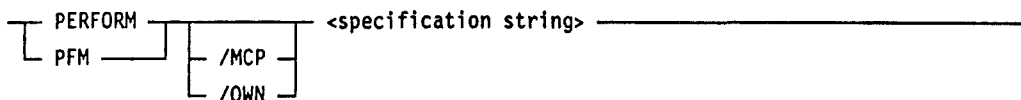
Access Level: 1

The PERFORM command schedules the DMPALL utility, which does the following:

- Prints the contents of a disk, diskpack, magnetic tape, or card file.
- Reproduces a data file from one hardware type to another.

If there is a free-standing user-supplied program called DMPALL on disk, that program is started. If not, the MCP intrinsic DMPALL utility is started.

Parameters are passed to DMPALL through a pass file on disk.



Explanation

PERFORM

These are synonymous.

PFM

/MCP

If /MCP is specified, the bound, MCP intrinsic DMPALL is scheduled regardless of the presence of a user-provided DMPALL on disk.

/OWN

The /OWN option can be used to prevent the scheduling of the bound, MCP intrinsic DMPALL when a user-provided DMPALL is required and is not on disk.

<specification string>

See the PERFORM option of the DMPALL intrinsic in the *V Series Systems Operations Guide Volume 3: System Utilities* for the available options and the method of specifying their inclusion in the <specification string>.

Examples

```
PERFORM LST1 A DSK UA
```

This produces a listing in UA format of a disk file named A.

PERFORM /MCP CRDCRD A B; CHARGE 99

This schedules the MCP version of DMPALL to reproduce a card file named A to a new file named B and inserts a user charge number of 99 into the Run log for the task.

PFM /OWN DSKDSK D F 100 10 500 20; CHARGE 34

This schedules a user-provided version of DMPALL to create a new file; changes the file id; specifies record size, blocking factor, records per area, and number of areas of a disk file; and inserts a user charge number of 34 into the Run log for the task.

See also:

- The GENERATE (Schedule DMPALL for Execution) command on Page 2-180.
- The DMPALL utility in the *V Series Systems Operations Guide Volume 3: System Utilities* and the *V Series Programmer's Reference Guide*.

PG (Purge Magnetic Tape)

Keyboard Input Message

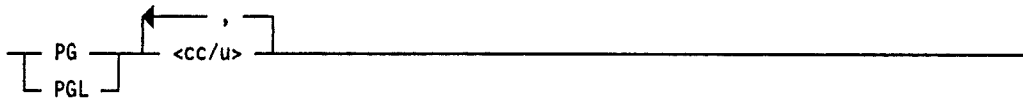
Access Level: 7

The PG command purges a magnetic tape on a unit that is ready, has a write ring, and is not in use.

The purge operation retains the label type and physical tape number (if any) currently written on the tape. If the tape was previously unlabeled, a Unisys standard label is written.

There are two formats of the PG command. The first format purges a tape. The second format lets you assign a job to the tape after purging and/or override a program's tape type request.

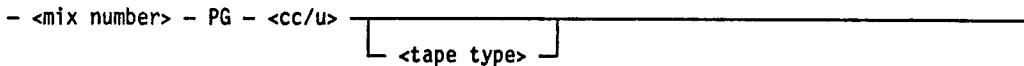
Purging a Tape



Explanation

- PG This purges the tape and leaves it ready and available to the system.
- PGL This indicates that the tape is to be locked after being purged.
- <cc/u> This is the channel number and unit number where the magnetic tape is located.

Assigning a Tape to a Job After Purging



Explanation

- <mix number> This is the mix number of the program to which the purged tape is to be assigned.
If a mix number is specified and the selected job is waiting for an output tape, the tape is purged and assigned the highest tape density allowed by the program and the drive.

<tape type> This allows an override of the tape type requested by a user program. The tape type can be MT9 (NRZ format), MPE, GCR, or MTC.

Details

If you do not specify a <mix number> or a <tape type> and you use a GCR-capable drive, scratch tapes are written in GCR mode.

Examples

PGL 14/1, 14/2

This purges and then locks the magnetic tapes on unit number 1 and unit number 2 on channel number 14.

41 PG 6/3 GCR

This purges a GCR magnetic tape on channel number 6, unit number 3 and assigns it to the job with mix number 041.

See also:

- The AC (Purge and Place Tape Number in Label) command on Page 2-2.
- The RP (Ready and Purge Magnetic Tape) command on Page 2-329.
- The RY (Ready Peripheral Device) command on Page 2-338.
- The SN (Purge and Place Number in Tape Label) command on Page 2-350.

PM (Print Memory Dump)

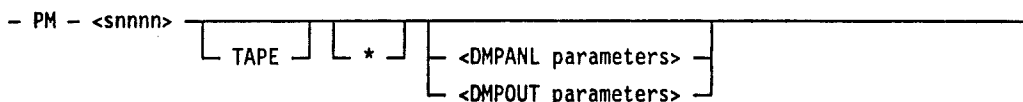
Keyboard Input Message

Access Level: 1

This command causes the analysis and printout of memory dump files by invoking one of two bound intrinsic programs: DMPOUT or DMPANL. DMPOUT analyzes a dump of specified user files; DMPANL analyzes a dump of the MCP file.

This command allows you to limit the scope of the printout for either program.

The following figure shows the general syntax for invoking these two programs.



Explanation

<snnnn>

This is the system number (*s*) of the system (0-3), and the dump file number *nnnn*.

To invoke DMPANL, enter PM 1 or PM s0001, where *s* is the number of a system whose MCP file you want to analyze. PM 1 prints the MCP memory dump file of the current system. PM s0001 prints the MCP memory dump file of the system indicated by the system number *s*.

To invoke DMPOUT, enter PM <snnnn>, where *nnnn* is the number of the user dump file and can range from 2 through 9999. The system number *s* can be omitted if the user dump file is present on the system from which the command was entered.

TAPE

This indicates that the source of the memory dump is on tape and is valid only for DMPANL.

*

This is an input option that allows you to enter multiline input in the form of AX commands: <mix number> AX <parameter>.

For example:

```
5 AX FILE, DATAPAGE END
```

You must terminate the last AX command with the word END. If you enter <mix number> AX HELP, the system will display the parameters for the program (DMPOUT or DMPANL, depending on which you invoked).

<DMPANL parameters> These allow you to limit the printout to specific tables and memory areas. See the DMPANL utility in the *V Series Systems Operations Guide Volume 3: System Utilities* for information on the parameters of the DMPANL utility.

<DMPOUT parameters>

See "Generating and Analyzing User Memory Dumps" in the *V Series MCP/VS Programming Reference Manual* for information on the parameters of the DMPOUT utility.

Examples

PM 1 TBL

This prints the system tables with no raw memory dump.

PM 1 DCP MIX

This prints all information relating to DCPs, as well as the Mix Table.

PM 23 DATAPAGE SAVE

This prints the MCP data page for the user dump file 23, and leaves the dump file on disk.

See also:

- The DMPANL utility in the *V Series Systems Operations Guide Volume 3: System Utilities*.
- Discussions of user memory dumps in the *V Series MCP/VS Programming Reference Manual*.

PO (Power Off Diskpack)

Keyboard Input Message

Access Level: 7

The PO command powers off the designated diskpack unit once all active files resident on the diskpack have been closed.

- PO - <cc/u> _____

Explanation

<cc/u> This is the channel number and unit number of the diskpack.

Restrictions

This command is not valid for MD4, MD8, or 680 diskpack types.

Also, the following conditions must be met before the diskpack can be powered off:

- The diskpack cannot be SHARED.
- The diskpack cannot have any areas in use or be assigned to any task.
- The diskpack cannot be in use by direct I/O.
- The diskpack must have been previously saved with a SV (Save a Peripheral Unit) command.
- The diskpack must be ready. A diskpack can be made ready through the RY (Ready Peripheral Device) command.

Example

PO 19/1

This powers off the diskpack located at channel number 19, unit 1.

See also:

- The RY (Ready Peripheral Device) command on Page 2-338.
- The SV (Save a Peripheral Unit) command on Page 2-368.

PP (Print Pack Sectors)

Keyboard Input Message

Access Level: 5

PP invokes the intrinsic program DSKOUT to print selected sectors from a diskpack. The starting address and numbers of sectors given at execution time specify which sectors to print.

```
- PP - <cc/u> - <begin address> _____
                               |_____
                               |<number of sectors>|
```

Explanation

<cc/u>	This is the channel number and unit number location of the diskpack.
<begin address>	This specifies the beginning point for the printing. The address must be in the range 0 through 16700000 (or the diskpack limit) and must be specified in decimal format.
<number of sectors>	This indicates the number of sectors to be printed. If this element is omitted, a value of 1 is assumed. If specified, the value must be less than 10000000.

Output Options

Various output options that provide alternate sorting and analysis are available. A full description of these options and their applications appear under the KA (Analyze Disk Directory) command.

Examples

```
PP 10/11 2500 10
```

This prints 10 sectors from the diskpack located at channel number 10, unit number 11, with the starting address of 2500.

```
PP 1/2 6
```

This prints 1 sector from the diskpack located at channel number 1, unit number 2, with the starting address of 6.

See also:

- The KA (Analyze Disk Directory) command on Page 2-201.
- The KP (Print Disk Segments) command on Page 2-205.
- The KS (Analyze Disk Space) command on Page 2-207.
- The PA (Analyze Pack Directory) command on Page 2-259.
- The PS (Analyze Pack Space) command on Page 2-303.

PR (Change Priority)

PRM (Change Memory Priority)
 PRP (Change Processing and I/O Priority)
 PRS (Change Scheduling Priority)

Keyboard Input Message

Access Level: 3

The PR command changes the memory, processing and I/O, and schedule priorities of a program in the mix or schedule. You can change all the priorities to a common setting at the same time or change each priority individually.

```

- <mix number> PR _____ <priority>
                |
                | PRM
                | PRP
                | PRS
  
```

Explanation

<mix number>	This is the mix number of the program whose priority is to be changed.
PR	This changes and provides a common setting for the memory priority, processing and I/O priority, and scheduling priority of a program.
PRM	This changes the memory priority of the specified program.
PRP	This changes the processing and I/O priority of the specified program.
PRS	This changes the scheduling priority of the specified program. The scheduling priority can be changed only while the program is in the scheduling and not yet running.
=	This is optional and does not influence these commands.
<priority>	This is a number ranging from 1 to 9, with 1 the lowest priority and 9 the highest. The MCP assigns a default priority of 4 unless directed otherwise by certain control commands and defaults.

Examples

2 PR 5

This causes the program whose mix number is 2 to have a memory, processing and I/O, and schedule priority of 5.

4 PRM 3

This causes the program whose mix number is 4 to have a memory priority of 3.

3 PRP 9

This causes the program whose mix number is 3 to have a processing priority of 9, the highest priority.

2 PRS = 6

This causes the program whose mix number is 2 to have a schedule priority of 6.

See also:

- The PRIORITY (Set Common Priority) command on Page 2-301.
- The CONTROL PRIORITY record in the *V Series Systems Operations Guide Volume 1: Installation*.

PRINT (Print Files)

MCP Control Instruction

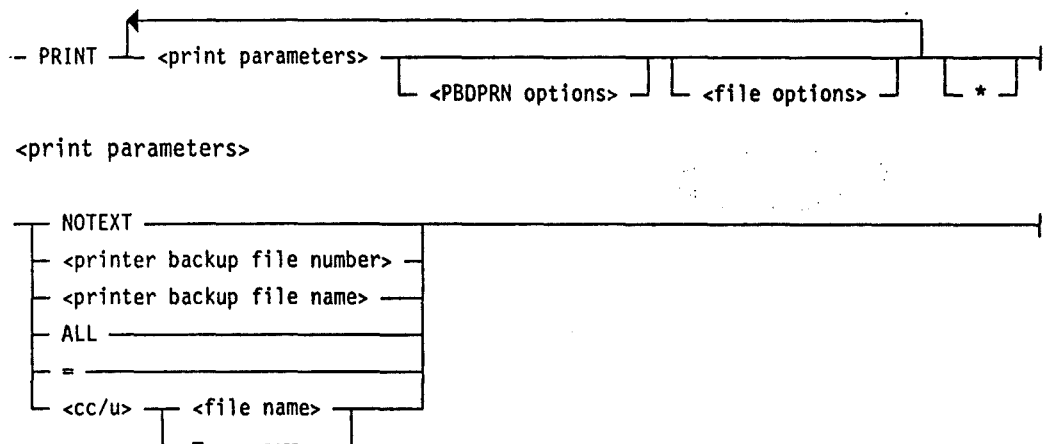
Access Level: 1

PRINT invokes the PBDPRN system utility that prints files on various printing devices.

PBDPRN can do the following:

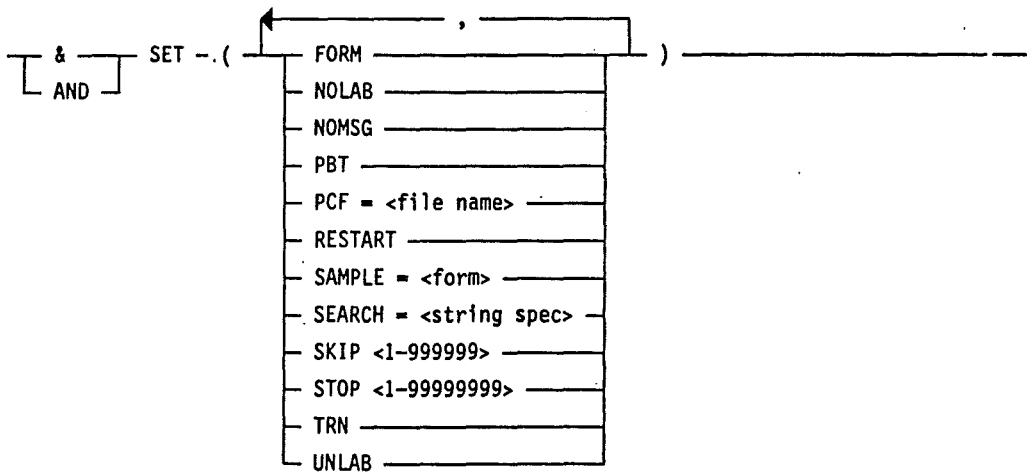
- Print printer backup files by number or name
- Print printer backup files from tape
- Create printer backup tapes
- Search the backup file for a specified string of text and begin printing when it finds it
- Start printing at a specified position in a file and stop at another position
- Translate all lowercase letters to uppercase letters as it prints
- Skip lines for the printout
- Stop printing after a specified number of lines
- Restart the printing of a print job that had been stopped by the QT (Quit Program Operation) command

The following figure shows the syntax for the PRINT command.

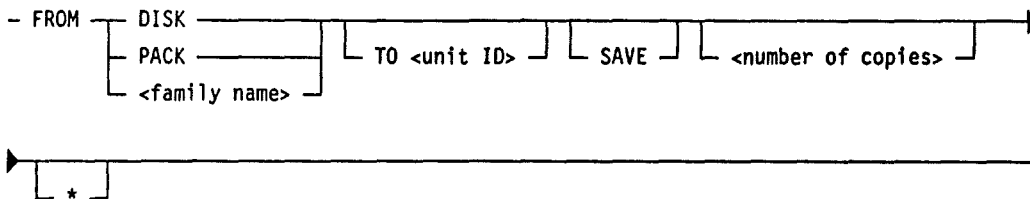


PRINT

<PBDPRN options>



<file options>



Explanation

NOTEXT

This prints a sample of a form or a printer control file (PCF) produced by the Forms Manager program. This option is used only with an Image Page Printer (IPP).

<printer backup file number>
<printer backup file name>

This prints an individual file according to its backup file number or backup file name.

ALL
=

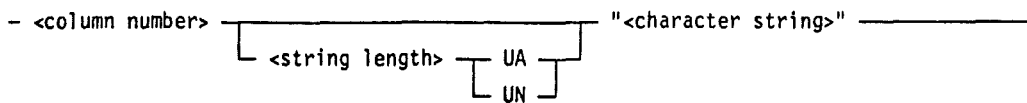
This causes PBDPRN to print all printer backup files. Unless DISK or PACK is specified, this option prints all the printer backup files on both disk and diskpack.

<cc/u>
<cc/u> <file name>
<cc/u> =

PBDPRN can print printer backup files from tape if you specify the channel number and unit number of the tape drive. Entering only the <cc/u> prints all printer backup files from the specified tape. If a file name is used, the system prints only that file. If the equal sign (=) is used, the system prints all the printer backup files on that tape.

AND SET & SET	These are synonymous. The options following AND SET (& SET) must be within parentheses. If you are using more than one option, separate each with a comma.
FORM	This directs output to the printer previously specified in the MCP file equate control card. For example, FILE PRINT = SHORT/OUTPUT.
NOLAB	This stops a block letter label from being printed on the cover page of the printout. However, the line that indicates the backup file number, its name, and the file label information will be printed.
NOMSG	This stops PBDPRN from displaying messages on the system console when it finishes printing copies of a file.
PBT	This directs the printer backup listing to tape.
PCF = <file name>	This must be used to print backup files on an Image Page Printer (IPP). When NOTEXT is specified, an entire printer control file (PCF), which can contain multiple forms, is printed as a sample.
RESTART	<p>This restarts a print job that has been stopped with a QT command or as a result of some IPP VSID message protocol errors that terminated PBDPRN.</p> <p>This option tells PBDPRN to start printing at the beginning of the page it was printing when it was stopped. If it does not find the beginning of a page, it resumes at the beginning of the last line it was printing when it was stopped. This option does not work for files on printer backup tapes.</p> <p>For this option, use the same syntax used to start the job originally, but include the option RESTART.</p>
SAMPLE = <form>	This prints sample forms in the process of being created by the Forms Manager program. It prints only the form, no text. It is used only with an IPP and is reserved for use by the Forms Manager program.
SEARCH <string spec>	This tells PBDPRN to search a column of text and to start printing the file where it finds a matching string of characters. The string cannot exceed 99 characters. The following figure illustrates the syntax for the string specifications.

PRINT



- | | |
|--|---|
| <column number> | This instructs PBDPRN which column to search. The column number must be in the range of 1 through 132. Note that the sum of the column number and the string length cannot exceed 133. |
| <string length> | This indicates the number of characters and blanks in the string. The string cannot be longer than 99 characters, including blanks. |
| UA
UN | These represent data types. UA indicates that the string is a representation of bytes as seen on a printed listing. |
| "<character string>" | This causes PBDPRN to search for this string of characters. The string cannot be longer than 99 characters, including blanks. |
| SKIP <1-999999> | This tells PBDPRN to begin printing at the specified line number and can be used with the STOP option to print only part of a file. The line number must be in the range 1 through 999999. |
| STOP <1-99999999> | This tells PBDPRN to stop printing at the specified line number and can be used with the SKIP option to print only part of a file. The line number must be in the range 1 through 99999999. |
| TRN | This tells PBDPRN to translate all lowercase letters to uppercase letters. |
| UNLAB | This suppresses the printing of a label or other types of information on the cover page of the printout. |
| FROM DISK
FROM PACK
FROM <family name> | FROM DISK indicates that the printer backup file is on disk.
FROM PACK indicates that the file is on diskpack.
PBDPRN searches for the file in the following order: |

- 1) The primary backup diskpack
- 2) The secondary backup diskpack (if defined)
- 3) All unrestricted diskpacks

Primary and secondary backup diskpacks are declared during system initialization and can be displayed or changed using the SHOW or ALTER commands.

FROM <family name> indicates that the printer backup file is stored on a specific family. If a FROM clause is not included in the PRINT command, PBDPRN searches both disk and diskpack by default.

TO <unit ID>	allows you to use a particular printer. If you are using an IPP, you must use the option PCF = <file name>. If this option is not used, PBDPRN prints the file on the first available line printer.
SAVE	stops PBDPRN from removing the printer backup file after it is printed.
<number of copies>	This allows you to print more than one copy. Up to 99 copies can be printed for each print job. This option is ignored if PBDPRN is printing backup files from a tape.
*	This tells PBDPRN to go into a waiting accept condition. PBDPRN will display ENTER SPECS on the ODT and prompt you with an ACCEPT. You can then issue parameters to PBDPRN with the MCP system command AX. Use of the AX command in conjunction with the PRINT command is described under "Using AX to Enter Print Specifications".

Using AX to Enter Print Specifications

If an asterisk (*) was the last character of a PRINT command, PBDPRN displays the message ENTER SPECS on the ODT. You use the AX command to enter print specifications. The following discusses the three possible responses to PDBPRN.

Using AX to Specify a Skip Count

The following figure shows the syntax used to specify the number of records to skip before beginning printing.

- <mix number> - AX - <skip count> _____|

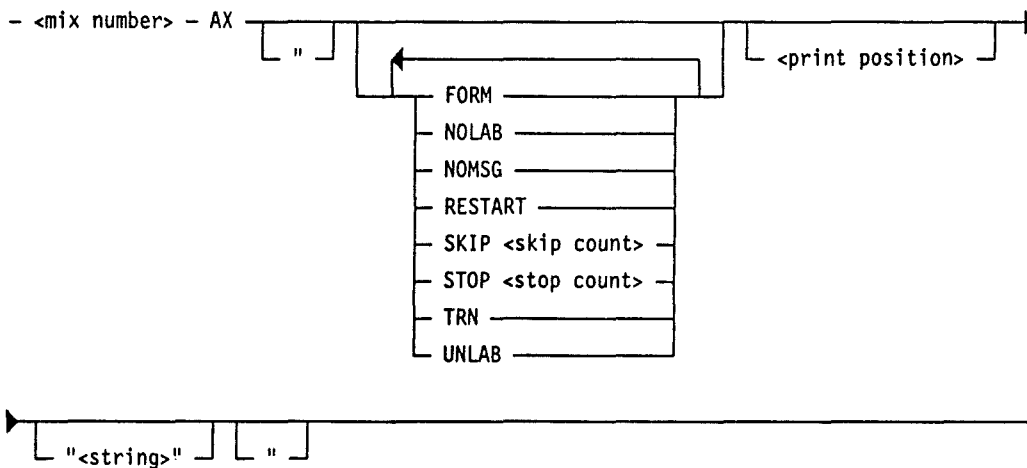
PRINT

Explanation

- <mix number>** This refers to the mix number of the job you want to respond to.
- <skip count>** This specifies the number of records to skip before printing begins. This must be a one- to six-digit unsigned integer.

Using AX to Specify Print Options

The following figure shows the syntax used to specify print options to PBDPRN.



Explanation

- <mix number>** This refers to the mix number of the job you want to respond to.
- FORM** This forces the output to be declared special forms.
- NOLAB** This suppresses the printing of block-printed labels. The first line (specifying file-name, and so forth) still appears unless the file name is absent or labels are omitted.
- NOMSG** PBDPRN normally displays a message on the ODT indicating that it has completed printing a copy of the file when multiple copies are requested. This option suppresses these messages.
- RESTART** This can be used to resume printing of a file that was previously ended with a QT (Quit Program Operation) command. When this option is specified, PBDPRN searches backward from the last line printed, looking for a line containing a "SKIP TO CHANNEL ONE"

code. If such a line can be located within 100 lines of where the printing of the file was ended, printing is resumed immediately following this line; if not, printing resumes with the line following the last one printed when the printing of the file was ended with a QT command.

SKIP <skip count>	This specifies that PBDPRN should skip the specified number of records before starting to print the file. If multiple SKIP entries are present, only the last is used. However the syntax of each SKIP entry is checked for being within the file limits.
STOP <stop count>	This terminates the printing of the file at the specified record number. If SAVE is not specified as part of the the PC command, the file is removed at the completion of printing. This option can be used in conjunction with the SKIP option to print only part of a file.
TRN	This translates all lowercase alphabetic characters to uppercase.
UNLAB	This suppresses printing of labels completely, and is equivalent to the effect of declaring "LABELS OMITTED" on the backup file.
<print position> "<string>"	This specifies that printing is to begin when the specified string is found at the specified print position. The print position is a 1- to 3-digit unsigned integer, less than 133, specifying the column position in which the string starts. The first print position is one. The length of the string added to the print position cannot exceed 133. The string must be enclosed in quotation marks and it must follow COBOL rules. For example, an embedded quotation mark must be represented by two consecutive quotation marks. "ISN'''T" represents the string ISN'T, which is five characters long.

Using AX to Specify Search Strings

The following figure shows the syntax used to specify search strings to PBDPRN.

- <mix number> - AX - <print position> - <length> - UA - <string> _____|

Explanation

<mix number> This refers to the mix number of the job you want to respond to.

PRINT

<print position>	This specifies the first position of the record at which to begin searching for the specified string. The print position is a 1- to 3-digit unsigned integer, less than 133, specifying the column position in which the string starts. The first print position is one.
<length>	This is an integer specifying the length of the string to be searched for. The length of the string added to the print position cannot exceed 133.
UA	This indicates that an alphanumeric compare will be performed.
<string>	This is the actual data for which the file is searched. It is not enclosed in quotes and begins at the first non-blank character following UA and continues to the end of the accept message. It must be exactly the same length as specified by the <length> parameter. If the string contains any special characters, the entire ACCEPT message must be enclosed in quotes.

Examples

PRINT 12345

This prints the printer backup file @12345 on any available line printer.

PRINT 12345 AND SET (SKIP 100, STOP 1000)

This prints the printer backup file @12345 starting at line 100, and stops printing at line 1000. Because no printer was specified, the file will be printed on any available line printer.

PRINT 6/1 =

This prints all the backup files from the tape on channel number 6, unit number 1. Because no printer was specified, the file will be printed on any available line printer.

PRINT BKFILE & SET (SEARCH 10 'SAM') TO SPEC 5

This searches columns 10-12 of each line and prints from the first occurrence of the string SAM to the end of the file. The system will use the printer SPEC and print five copies.

PRINT 12345 AND SET (PCF=DOPCPF)

This prints the file @12345 on the IPP using DOPCPF form file.

PRINT NOTEXT & SET (PCF=FORMSS)

This prints a sample of a form created by the Forms Manager program, printing only the form, with no text.

Note: DOCPCF and MCPDCF are two printer control files available for general use on the image printer. DOCPCF will format printer backup data in portrait mode (80 characters by 60 lines). MCPDCF will format printer backup data in landscape mode (132 characters by 60 lines), as well as in duplex mode. To create other forms and PCFs, you can use the Forms Manager program.

Executing PBDPRN

PBDPRN can be executed with the MCP system command EX by using the following syntax:

```
EX PBDPRN
```

After it is executed, PBDPRN sends the following message:

```
PBDPRN=<mix number> ENTER SYNTAX. AX 'END' TO TERMINATE INPUT.
```

Auto Printing

An MCP file equation command can be used to direct printer backup files to the Image Page Printer (IPP) automatically.

The following paragraphs describe the different syntax used to send files automatically to a line printer versus an IPP.

```
?FILE LINE = SPEC/OUTPUT FORM AUTO.
```

A sample command to output this file to an IPP:

```
?FILE LINE = DOCPCF/OUTPUT FORM AUTO.
```

Note: DOCPCF is the unit ID for an IPP.

PBDPRN System Software

The B9290-30 IPP utilizes a high-level software protocol, known as Virtual Static Image Device (VSID), that communicates with the IPP. The PBDPRN utility uses the VSID protocol to communicate with the IPP unit.

The system software that runs and manages the IPP unit consists of a resident operating system (ROS), loadable operating system (LOS), and the loadable diagnostics. The ROS is a PROM-based module that resides on the GASP card in the IPP unit. The LOS is a software module that is downloaded from the host

system. The loadable diagnostics is the software that is used to inform the user of system errors regarding the IPP.

Use the following procedure at the ODT to download the software necessary for IPP operation.

1. Set the IPP option on the system with the following ODT command:

```
SO IPP
```

2. Enter the following command to check whether an IPP has been added as a peripheral device:

```
OL IPP
```

If an IPP has not been added to the system, enter the following command to add an IPP to the system:

```
UNIT <cc/u> MCPPCF IPP
```

In the preceding syntax, MCPPCF is the unit ID recognized by PBDPRN and also found in the PRINT command syntax diagram, and where IPP is the unit ID recognized by the system.

3. To download the LOS, the maintenance panel on the IPP console should first display the following message:

```
RESIDENT OPERATING SYSTEM V <version number>
```

Enter the following command to load software to the IPP:

```
LH <cc>
```

In the preceding syntax <cc> is the channel number of the IPP unit. When the load is complete, the maintenance panel on the IPP console should display the following message:

```
''LOADABLE OPERATING SYSTEM V <version number>
```

4. The IPP unit is now ready to print.

See also:

The PBDPRN utility in the *V Series Systems Operations Guide Volume 3: System Utilities*.

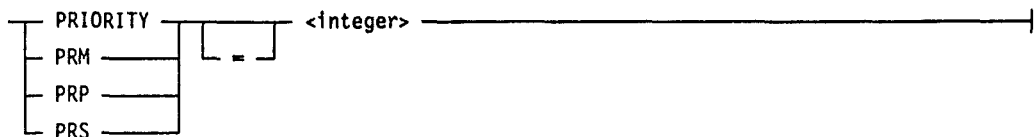
PRIORITY (Set Priority)

PRM (Set Memory Priority)
 PRP (Set Processing and I/O Priority)
 PRS (Set Schedule Priority)

MCP Control Instruction

Access Level: 3

The PRIORITY command assigns a common setting of the memory priority, processing and I/O priority, and scheduling priority for a program at the time of scheduling. To set the different types of priority individually, use the PRM, PRP, and PRS commands.



Explanation

PRIORITY PR	These are synonymous. Do not confuse this command with the PR (Set Priority) command, which is a keyboard input message that changes the priority of a job already in the mix or schedule. This PR command sets the priority of a job before it enters the mix.
PRM	This specifies the memory priority to be assigned to a given program.
PRP	This specifies the processing and I/O priority to be assigned to a given program.
PRS	This specifies the schedule priority to be assigned to a given program.
=	This is optional and does not influence the command.
<integer>	This denotes the priority to be assigned. Priorities can range from 1 to 9, with 1 the lowest priority and 9 the highest. The MCP assigns a default PRIORITY of 4 unless directed otherwise by certain control commands and defaults.

Details for PRM

The PRM command indicates the priority to be given a program when the system allocates memory space. The PRM command applies only after the MCP determines that a program's schedule priority qualifies it for the mix.

PRIORITY

If a program has sufficient schedule priority to enter the mix and sufficient memory is not available, the memory priorities of all jobs in the mix are compared with the memory priority of the job in the schedule.

If any jobs in the mix have a memory priority less than the memory priority of the job in the schedule, the MCP moves the program from memory to disk, to create space for the higher-memory-priority job. The MCP marks these jobs as **PRIORITY STOPPED**. Priority-stopped jobs take precedence over jobs of equal or lower memory priority when memory becomes available.

In addition, if a job in the mix requires memory for any reason (such as, disk file header space, or printer backup space), jobs with a lower memory priority can be priority-stopped to make memory available.

Details for PRP

The MCP uses the PRP priority to determine the processing reinstate priority to be given to a program while in the mix. This priority does not affect job entrance into the mix from the schedule. While jobs are in the mix, however, the PRP priority establishes the order of reinstatement when the jobs are ready to run.

Note: A processor-bound program with a high process priority can prevent all jobs with a lower priority from being reinstated by the MCP.

Details for PRS

The PRS command allows you to assign a schedule priority to a program. This schedule priority establishes which jobs receive precedence to enter the mix. The program is not affected after it has entered the mix.

No job enters the mix, regardless of PRP and PRM priorities or memory availability, until all jobs of higher schedule priority have been initiated. Jobs with equal schedule priorities are initiated (assuming memory is available) based first on memory priority, then length of time in the schedule.

Examples

```
COMPILE B PASCAL; PRIORITY = 5
```

This compiles the program named B and assigns it a priority setting of 5.

```
EX C; PRM 7
```

This executes a program named C and assigns it a memory priority setting of 7.

See also:

- The PR (Change Priority) command on Page 2-289.
- The CONTROL PRIORITY record in the *V Series Systems Operations Guide Volume 1: Installation*.

PS (Analyze Pack Space)

Keyboard Input Message

Access Level: 5

PS starts the intrinsic program DSKOUT to perform a detailed analysis of the space allocation on the designated diskpack.

```
- PS [ NOSTANDBY ] <cc/u> [ = ] [ DSKAVL ]
```

Explanation

NOSTANDBY

This directs the operating system to capture information for DSKOUT even when the operating system *cannot* perform its normal standby function. The standby function is still attempted, but information is captured regardless of whether the function is successful. For more information about the standby function, see "Details" on the following pages.

<cc/u>

This is the channel number and unit number of the diskpack.

=

This causes analysis of the entire diskpack.

DSKAVL

This causes an analysis of the available space on the diskpack only.

Details

Files, available areas, missing, and duplicate diskpack are all listed. Each file's area addresses, file identifier, area size, area number, and file attributes are given.

For available areas, the area address, the string "A V A I L A B L E", and the size of the available space are given.

For missing or duplicate diskpack, the address of the area, the string "MISSING PACK" or "DUPLICATE PACK", and size of the area are given.

When you enter a PS command, the operating system can respond with the message:

KBD IGNORED: DISK OR PACK MAINT IS IN PROGRESS

This means the operating system cannot perform a standby function as part of the PS command. A standby function suspends updates to operating system data structures long enough to obtain an accurate "snapshot" of the structures. This

snapshot is passed to DSKOUT for analysis. The operating system needs exclusive access to these structures for an accurate DSKOUT analysis.

To respond to this situation, you can do either of the following:

- Wait until a later time when disk activity is less.
- Use the NOSTANDBY option of the PS command.

The NOSTANDBY option captures information for DSKOUT even when the operating system *cannot* perform the standby function. The standby function is attempted, but information is captured regardless of whether it is successful.

Using NOSTANDBY can create missing or duplicate disk messages that are not correct. This happens if an update occurs as the snapshot is taken. If NOSTANDBY is used and missing or duplicate disk conditions are encountered, DSKOUT flags the resulting messages with NOSTANDBY warnings.

These warnings mean the condition might be caused by non-exclusive access rather than by an actual missing or duplicate disk. Before you act on any missing or duplicate disk messages with a NOSTANDBY warning, enter another PS command without NOSTANDBY. Enter the command when less disk activity is taking place.

Output Options

Various output options that provide alternate sorting and analysis are available. A full description of these options and their application appear under the KA (Analyze Disk Directory) command.

Examples

PS 10/1 =

This invokes the DSKOUT intrinsic to analyze only the space allocation on the diskpack on channel 10, unit number 1.

PS 19/3 DSKAVL; CHARGE 6610

This invokes the DSKOUT intrinsic to analyze only the available space on the diskpack on channel 19, unit number 3. This job will be assigned a charge number of 6610.

PS NOSTANDBY 14/0 =

This invokes DSKOUT to analyze only the space allocation on the diskpack on channel 14, unit 0. The analysis is generated regardless of the outcome of the normal standby function.

See also:

- The KA (Analyze Disk Directory) command on Page 2-201.
- The KP (Print Disk Segments) command on Page 2-205.
- The KS (Analyze Disk Space) command on Page 2-207.
- The KX (Obtain User Disk Space) command on Page 2-210.
- The PA (Analyze Pack Directory) command on Page 2-259.
- The PP (Print Pack Sectors) command on Page 2-287.

()

()

()

QD (Query DEBUG Sessions)

Keyboard Input Message

Access Level: 8

The QD command identifies the DEBUG sessions present on the system and displays the 4-digit session number of each.

- QD _____

See also:

- The DEBUG (Attach Program to DEBUG Session) command on Page 2-121.
- The ED (Enter DEBUG Session) command on Page 2-155.
- The ID (Invoke DEBUG Session) command on Page 2-191.
- The DEBUG utility in the *V Series Systems Operations Guide Volume 3: System Utilities*.

QT (Quit Program Operation)

Keyboard Input Message

Access Level: 1

The QT command can do the following:

- Stop the printing or punching of a backup file
- Interrupt the DMPALL or DMPANL utility and cause it to go to a normal EOJ
- Terminate a timesharing program, so you do not have to discontinue it with a DS (Discontinue Program) command
- Terminate a Work Flow Language (WFL) job that has the RESTART option enabled

```

- <mix number> - QT _____|
      |-----|
      |  END  |
      |-----|
      |  ALL  |
      |-----|
  
```

Explanation

<mix number> This is the mix number of the program to be stopped.

You can use QT only on the following:

- DMPALL, DMPANL, DMPOUT, DSKOUT, LDCNTL, PBDPRN, and PCHOUT utility programs
- Jobs from the CANDE shares mix
- Any restartable WFL job

Note: *You cannot use the QT command on a locked program. If you want to use QT on a restartable WFL job, you must first unlock the job with the UP (Unlock Program) command.*

END This causes the specified intrinsic to stop printing or punching at the end of the current file and saves or purges the files as requested in the PRINT (Invoke PBDPRN), PC (Punch Backup File), or PM (Print Memory Dump) command.

END terminates timesharing in shared areas only, and only if no other users are present.

ALL This causes the designated intrinsic to immediately stop printing or punching all files, leaving all files not previously purged on disk or diskpack.

ALL terminates timesharing if no users are present and does not allow further log-ins.

Details

If you omit the optional END and ALL specifications, the print or punch utilities immediately stop printing or punching the current file. The utilities go to the next file, if one has been requested, and leave the previous file on disk.

If you are quitting a timesharing program and omit the END or ALL option, the timesharing process terminates immediately.

If the QT command is used to terminate a restartable WFL job, the WFL job removes its own entry from the restart file and then terminates itself. It cannot then be inadvertently restarted by a subsequent EXECUTE (Execute Library Program) command.

Examples

3 QT

This stops processing the program with mix number 003.

5 QT ALL

This stops the program with mix number 005 from printing or punching all files, leaving all files not previously purged on disk. If the program is a CANDE job and no users are present, timesharing will be ended and no user will be able to log in.

See also:

- The DS (Discontinue Program) command on Page 2-148.
- The PC (Create Punch Backup File) command on Page 2-275.
- The PRINT (Print Files) command on Page 2-291.
- The PM (Print Memory Dump) command on Page 2-284.
- The UP (Unlock Program) command on Page 2-398.
- The *V Series CANDE Installation and Operation Guide*.
- The *B 2000/B 3000/B 400/V Series WFL Compiler Programming Reference Manual*.

QWIKPOOL (Set Program Overlay Buffer Pool)

MCP Control Instruction

Access Level: 8

QWIKPOOL sets the total size of the user program overlay buffer pool. The pool is used to store program overlays. QWIKPOOL overrides the buffer size set at cold start by the system default or by the LIMIT QWIKPOOL record.

You can enter the QWIKPOOL command only when the QWIK option is reset. See RO (Reset System Options) for information about resetting the QWIK option.

- QWIKPOOL - <integer> _____

Explanation

<integer> This specifies the pool size in kilodigits (kd). The default setting is 400 kd.

Details

The user QWIK pool is memory dedicated to storing the most recently used user program overlays, so that they will not have to be accessed from disk if they are needed again. This feature usually increases performance. You can limit the amount of memory to be used by the QWIK pool with the LIMIT QWIKPOOL record of the system configuration file. You can set the QWIK pool option with the USE QWIK record or by using the SO (Set System Options) command. You can reset it by using the RO (Reset System Options) command.

User QWIK memory differs from MCP QWIK memory in that MCP QWIK memory is memory dedicated to storage of the most recently used MCP overlays and is set by the LIMIT QWKMEM record in the system configuration file.

Example

RO QWIK; QWIKPOOL 600; SO QWIK

This resets the QWIK system option and then sets the total size of the user program memory buffer pool to 600 kd. Then the QWIK option is reenabled.

See also:

- The QWKMEM (Set MCP Overlay Buffer Pool) command on Page 2-311.
- The RO (Reset System Options) command on Page 2-327.
- The SO (Set System Options) command on Page 2-352.
- The WO (Display Active User QWIK Operation Counts) command on Page 2-418.
- The LIMIT QWIKPOOL and USE QWIK records in the *V Series Systems Operations Guide Volume 1: Installation*.

QWKMEM (Set MCP Overlay Buffer Pool)

MCP Control Instruction

Access Level: 8

The QWKMEM command allocates a fixed amount of memory exclusively for MCP overlay use.

- QWKMEM - <integer>

Explanation

<integer>

This specifies the amount of memory in kilodigits (kd) to be reserved for MCP use and can have any value from 0 kd through 999 kd.

Details

MCP QWIK memory differs from user QWIK memory in that MCP QWIK memory is memory dedicated to storage of the most recently used MCP overlays and is set by the LIMIT QWKMEM record in the system configuration file.

Example

```
QWKMEM 60
```

This designates 60 kd of memory for MCP use only.

See also:

- The QWIKPOOL (Set Program Overlay Buffer Pool) command on Page 2-310.
- The LIMIT QWKMEM record in the *V Series Systems Operations Guide Volume 1: Installation*.

RA (Remove After-Linkage)

Keyboard Input Message

Access Level: 1

The RA command removes after-linkage for the indicated task, allowing it to enter the mix.

- <mix number> - RA

Explanation

<mix number> This must be the mix number of a task that is in the schedule and cannot enter the mix because it is to be executed after another task.

Details

An after-linkage is a state that exists between two tasks when the AFTER (Hold Program) command is used.

For example:

```
EX TASK1
EX TASK2 AFTER TASK1
```

First, TASK1 is executed. Then, an AFTER command is issued which specifies that TASK2 is to wait in the schedule until TASK1 finishes executing. An after-linkage now exists between the two tasks. If TASK1 does not go to a normal EOJ and the AFTR option is reset (off), you can do one of the following:

- Use the RA command to remove the after-linkage if you want TASK2 to enter the mix.
- Use the RS (Remove Job From Schedule) command to remove TASK2 from the schedule.

Example

```
27 RA
```

This removes the after-linkage of a task in the schedule with mix number 027.

See also:

- The AFTER (Hold Program) command on Page 2-12.
- The DS (Discontinue Program) command on Page 2-148.
- The RS (Remove Job from Schedule) command on Page 2-333.
- The USE AFTR record in system configuration file in the *V Series Systems Operations Guide Volume 1: Installation*.

RD, RDP (Remove Pseudo Decks)

Keyboard Input Message

Access Level: 5

RD removes pseudo card decks from the system. However, currently active pseudo decks are not affected.

Removing Pseudo Decks from Disk

```
- RD [ <system number> ] [ A ] [ == ] [ <deck number> ]
```

Explanation

RD	This indicates that the decks to be removed are on disk.
<system number>	This selects pseudo decks created by a particular system in a multi-system shared configuration. If <system number> is not supplied, the default is the system from which you enter the RD command.
A	The letter A selects pseudo decks created by all systems.
===	Three equal signs select all pseudo decks.
<deck number>	This removes the pseudo deck with the specified pseudo deck number. Pseudo decks are designated as #snnnn. The s represents the system number and is not required unless the deck was created by a system other than the one from which you are entering the command. The nnnn is a 4-digit number with leading zeros assumed. Do not include the pound sign in the command syntax.

Removing Pseudo Decks from Diskpack

```
- RDP [ <system number> ] [ A ] [ <cc/u> ] [ == ] [ <deck number> ]
```

The <system number>, A, ==, and <deck number> syntax elements have the same meaning when used with the RDP command as they do when used with the RD command.

Explanation

RDP This indicates that the decks to be removed are on diskpack.

<cc/u> This is a channel number and unit number and can be used to designate that only files from a specific diskpack are to be removed. You can specify <cc/u> only in an RDP command.

Details

The RD command removes only the pseudo decks that are not currently active.

You can display on the ODT the inactive pseudo card decks with the CD, CDP (List Inactive Pseudo Decks) command. An active deck can be deactivated by the DA, DAP (Deactivate Pseudo Reader) command, after which it can be removed by the RD command.

Examples

RD 5

This removes pseudo deck 0005 from the system on which the command is entered.

RD 20035

This removes pseudo deck 0035 from system 2.

RD 3 12

This removes pseudo deck 0012 from system 3.

RD A ==

This removes all inactive pseudo decks from all systems.

RDP 6/0 ==

This removes all inactive pseudo decks from the diskpack that is channel 6, unit 0.

See also:

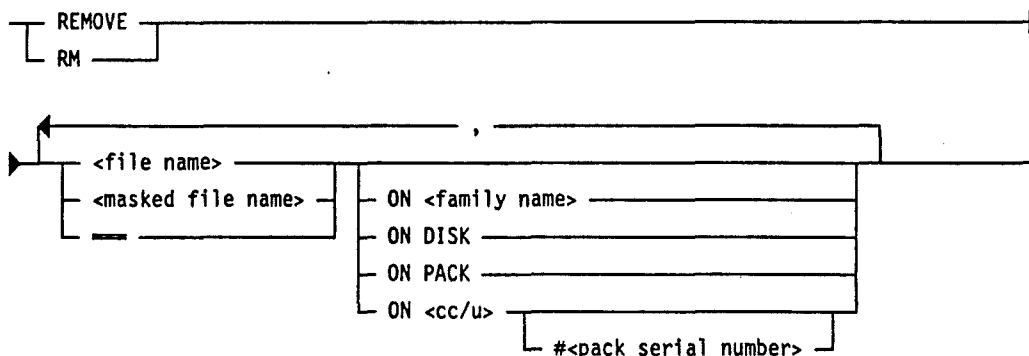
- The CD, CDP (List Inactive Pseudo Decks) command on Page 2-68.
- The DA, DAP (Deactivate Pseudo Reader) command on Page 2-103.
- The RN, RNP (Activate Pseudo Reader) command on Page 2-324.

REMOVE (Delete Files)

MCP Control Instruction

Access Level: 5

The REMOVE command deletes files from the disk or diskpack directory, making the file space available.



Explanation

REMOVE
RM

RM is an abbreviation for REMOVE. Do not confuse this abbreviation with the RM keyboard input message, which is used to handle a DUP LIB condition.

<file name>
<masked file name>
===

This is the name of the file to be removed. Equal signs (=) can be used as masking characters. Three or more equal signs (===) refers to all files.

ON <family name>

This is the name of the diskpack family on which the file is located. For example, ON MYPACK.

ON DISK

This indicates that the file is on 100-byte-per-sector disk. This is the default.

ON PACK

This indicates that the file is located on 180-byte-per-sector diskpack. All nonrestricted diskpacks will be searched for the file.

ON <cc/u>

This indicates that the base area of the file is on a diskpack drive with the specified channel number and unit number. For example, ON 07/3.

If the diskpack specified by <cc/u> is a Version 2 diskpack, it must be the base pack of a pack family. It is invalid to specify a continuation pack of a Version 2 diskpack family.

REMOVE

Note: For more information on pack family versions, refer to "Pack Subsystems" in the V Series Systems Operations Guide Volume 3.

ON <cc/u> # <pack serial number> This removes the areas of the file that are on a diskpack drive with the specified channel number and unit number and whose base area is a diskpack with the specified serial number. Masking characters are not allowed in the file name.

Caution

Be careful with the REMOVE <file name> ON <cc/u> # <pack serial number> syntax, as it is possible to remove part of a file and leave a partial file on the diskpack. This syntax is provided for recovery from diskpack failures, and not for normal use.

General Restrictions

Files in use are not removed. Files for which library maintenance procedures are not allowed are not removed. This includes certain system files and files designated as "NO LIB MAINTENANCE" with an ALTER FILE keyboard command or an ALLOCATE system configuration record.

A null mix is required to remove all files from disk or diskpack.

A comma (,) is required between file identifiers if the REMOVE command is entered from the ODT, or from the card reader with a SPO command.

Security Restrictions

This restriction applies only if the SECURITY record is present in the system configuration file.

If a file has a security type of PRIVATE or GUARDED, the file can be removed only by those allowed access to the file, or by users with LIBMAINT security privileges.

System security is discussed in the *V Series System Security Installation and Operations Guide*.

Duplicate Files On Diskpack

If a masking character is used in the file name, and files are found with duplicate names, all the duplicate files are removed, if possible.

If no masking characters are used in the file name and if two or more files with the same name are found on different families, no files are removed and the DUP FILE message is displayed on the ODT. You can then use the ON <family name> syntax to remove the file you want.

If two or more files with the same name are found on the same diskpack family, you can use the ON <cc/u> syntax to remove the file you want.

File Areas On Diskpack

Diskpack files are divided into file areas. Each area occupies contiguous space on the diskpack. The areas of a file are not necessarily contiguous to one another. In a diskpack family with multiple diskpacks, the areas of a file can be on different diskpacks in the family. However, a file is never split between several families. The purpose of breaking a file into areas is to allow more efficient use of the diskpacks.

The first area in a file is known as area 1 or the base area. If one diskpack in a multidiskpack family has been designated as the master diskpack, the base area of all files on that family will go on the master diskpack.

If the ON <family name> syntax is used, the base area of the file will be sought on the given diskpack family. If the base area is not found, no areas are removed, and the NO FILE message is displayed on the ODT.

If the ON <cc/u> syntax is used, only those files are removed whose base areas are on the given diskpack unit. If the base area is not found, no areas are removed, and the NO FILE message is displayed on the ODT.

If a file is being removed but some of its continuation areas cannot be found because of missing diskpacks, the available continuation areas will be removed. The base area will not be removed, but will be flagged that a remove is in progress. No further access is allowed to the file. When the remaining continuation areas (and the base area) are present, enter another REMOVE command. When all continuation areas are removed, the base area will be removed also.

The ON <cc/u> # <pack serial number> syntax is intended for the case in which the diskpack containing a file's base area is destroyed. It is the only way to remove continuation areas of a file without having access to the base area. Any areas on the given diskpack will be removed, whether they are the base area or continuation areas.

Details

This command works by removing the name and attributes of the file from the disk or diskpack directory and header and returning the space occupied by the areas of the file.

Examples

```
REMOVE FILEX
```

This removes the file named FILEX from disk.

REMOVE

REMOVE RS= ON PACK

This removes any files whose names begin with the letters RS that are on any nonrestricted diskpack.

REMOVE R=S ON PACKA

This removes any files whose names are five characters long, start with the letter R, end with the letter S, and are on the family named PACKA.

REMOVE FILEA ON MYPACK, BOB11 ON DISK

This removes the file named FILEA from the family named MYPACK and also removes the file named BOB11 from disk.

REMOVE AFILE ON 06/2

This removes the file named AFILE from the diskpack that is unit 2 on channel 6.

See also:

- The KX (Obtain User Disk Space) command on Page 2-210.
- The MR (Remove Duplicate File) command on Page 2-234.
- The RM (Remove Duplicate File) command on Page 2-322.
- The ALLOCATE record in *V Series Systems Operations Guide Volume 1: Installation*.
- System security is discussed in the *V Series System Security Installation and Operations Guide*.

RERUN (Log a Rerun Operation)

MCP Control Instruction

Access Level: 1

The RERUN command is used with a job initiation command, such as EXECUTE or COMPILE. It marks the job as a rerun operation in the system Run log (RLOG). The RERUN command has no effect on the scheduling or execution of any program.

— RERUN —

Details

This command sets the one-bit in the supplementary execution code field (RL-SXE, which is digit 71) of the Job Schedule record (type 3/0), the Short Job Schedule record (type 4/0), and Beginning-of-Job record (record type 6/0) for the job in the RLOG.

If one program starts another program using the ZIP facility, the RERUN command can be included in the ZIP syntax. In this way, an entire system of programs can be correctly flagged in the RLOG, as well as the one originally executed.

Examples

```
EXECUTE APROG; RERUN
```

This starts the program called APROG and marks it as a rerun operation.

```
COMPILE MYPROG WITH COBOL9; RERUN
```

This compiles the program called MYPROG and marks the compilation as a rerun operation.

See also:

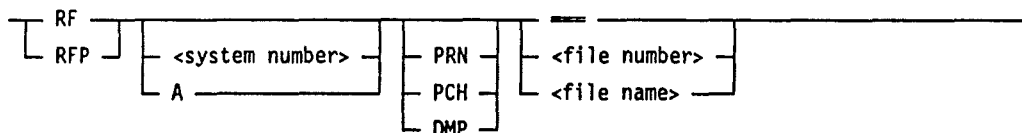
- The TEST (Mark a Test Operation) command on Page 2-371.
- The Run log in the *V Series System Software Logging Operations Reference Manual*.
- The ZIP facility is discussed in the various language manuals.

RF, RFP (Remove Backup File from Disk or Pack)

Keyboard Input Message

Access Level: 5

The RF command removes printer backup files, punch backup files, and memory dump files from disk or diskpack.



Explanation

RF removes files from disk. RFP removes files from diskpack.

<system number>
A
The <system number> indicates that only files created by that system are to be removed. The A means all systems. The default is the system through which the RF or RFP command is entered.

PRN
PCH
DMP
These select the type of file to be removed. The choices are as follows:

- PRN to remove printer backup files
- PCH to remove punch backup files
- DMP to remove memory dump files

Note that you cannot use RFP with dump files, as dump files are not written to diskpack. If you do, the message "****KBD IGNORED: INV SYNTAX" will be displayed.

===
This removes all files. If you use PRN, PCH, or DMP, all files of that type are removed. If you do not supply a file type, files of all three types are removed.

<file number>
This is the backup file number or dump file number of the file to be removed. The symbol (\$, @, or *) that prefixes the file number is not included. Since the first digit of the file number is the system number of the system that created the file, this number can be dropped if the file was created by the system from which the command is entered, or if the system number is otherwise given. Leading zeros can be dropped.

<file name>

This is the backup file name (internal name) of a printer backup file, punch backup, or dump file. For PRN and PCH files, the name is a multifile identifier or file identifier; for DMP files, the name is a program name that can be of the form multiprogram ID/program ID.

Restrictions

This command does not remove files that are in use.

Examples

RF PRN ==

This removes all printer backup files from disk.

RFP PRN 32

This removes the diskpack printer backup file whose backup file number is @s0032, where s denotes a system number. If the command is entered at system 2, for example, the file @20032 is removed.

RFP PRN 20032

This removes the diskpack printer backup file whose backup file number is @20032. This is a file created on system 2, no matter which system the command is entered from.

RFP PRN RPRT

removes the diskpack printer backup file whose name is RPRT.

RF DMP 2

This removes the dump file whose number is \$s0002, where s denotes a system number. If the command is entered at system 2, for example, the file \$20002 is removed.

RF PRN ABC=

This removes all disk printer backup files whose names begin with the letters ABC.

Note: The intrinsic BFRF either lists or lists and removes backup files according to a wide range of options.

See also:

- The BF, BFP (List Backup Files on Disk or Pack) command on Page 2-47.
- The BFRF (Search, List, Remove Backup Files) command on Page 2-49.

RM (Remove Duplicate File)

Keyboard Input Message

Access Level: 5

The RM command removes a file from disk or diskpack to resolve a DUP LIB situation. The new file remains intact.

Do not confuse RM (Remove Duplicate File), which is a keyboard input message, with the abbreviation for REMOVE (Delete Files), which is an MCP control instruction. The REMOVE command is a general command for removing *any* file, not just a duplicate file, from a disk or diskpack directory.)

A DUP LIB situation occurs when a program is trying to create a file but a file with that name already exists. The RM command removes the file that already exists.

- <mix number> - RM _____

Explanation

<mix number> This is the mix number of the task that is creating the file you want to save.

Details

The RM command removes the "old" file and leaves the "new" file, while the MR (Remove Duplicate File) command removes the "new" file and leaves the "old" file.

The RMOV option, if set, can resolve DUP LIB conditions on disk or diskpack by indicating that the old file is to be removed.

Example

The following is a scenario of messages displayed on the ODT in which the RM command is used. (These messages would be displayed on your system only if the OPEN, CLOS, BOJ, and EOJ system options were set.)

ODT Display	Comment
=> EXECUTE JOB1	The operator starts a program called JOB1.
BOJ JOB1=012 012586 9:15	JOB1 starts to execute and is assigned a mix number of 012.
DSK FILE FILEA OPEN OUT JOB1=012	JOB1 creates a file called FILEA. It is a disk file and has been opened for output.
DSK FILE FILEA CLOSED RELEASED JOB1=12	JOB1 closes the file FILEA.

ODT Display	Comment
=> EXECUTE JOB2	The operator starts a program called JOB2.
BOJ JOB2=018 103185 9:20	JOB2 starts to execute and is assigned a mix number of 018.
DSK FILE FILEA OPEN OUT JOB2=018	JOB2 creates a disk file that happens to be called FILEA.
**DUP LIB FILEA JOB2=018	JOB2 tries to close the file FILEA. The MCP detects that there is already a file called FILEA. JOB2 waits.
=> SPO 018 RM	The operator enters the RM command. This will remove the version of FILEA that has been created by JOB1, allowing the "new" version of FILEA to close.
DSK FILE FILEA CLOSED RELEASED JOB2=018	JOB2 is now able to close its file.
EOJ JOB2=018 9:30 IN 1:36 CHG 2:22	JOB2 continues executing until it finishes.

See also:

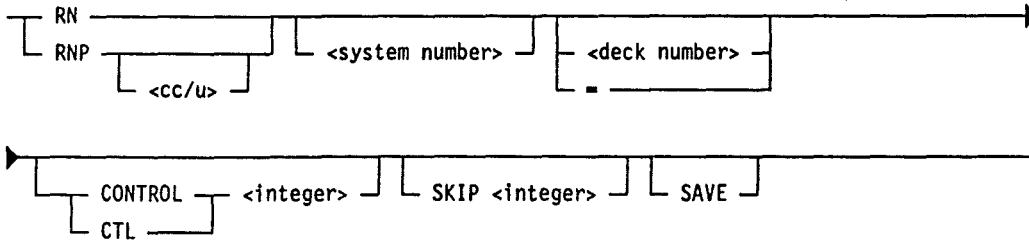
- The REMOVE (Delete Files) command on Page 2-315.
- The MR (Remove Duplicate File) command on Page 2-234.
- The USE RMOV record in the *V Series Systems Operations Guide Volume 1: Installation*.

RN, RNP (Activate Pseudo Reader)

Keyboard Input Message

Access Level: 1

The RN command processes the control records in a pseudo deck on disk or diskpack.



Explanation

<p>RN RNP</p>	<p>RN indicates that the pseudo decks to be activated are on disk; RNP, on diskpack.</p>
<p><cc/u></p>	<p>This can be used to designate that only decks from a diskpack with the specified channel number and unit number are to be activated.</p>
<p><system number></p>	<p>This selects pseudo decks created by a particular system in a multi-system shared configuration.</p> <p>If <system number> is not supplied, the default is the system from which you enter the RN command.</p>
<p><deck number></p>	<p>This is the number of an individual deck that is to be activated.</p> <p>Pseudo decks are designated as #snnnn. The <i>s</i> represents the system number and is not required unless the deck was created by a system other than the one from which you are entering the command. The <i>nnnn</i> is a 4-digit number with leading zeros assumed.</p> <p>Do not include the pound sign (#) in the command syntax.</p>
<p>=</p>	<p>This indicates that all decks are to be activated. For example, RN = activates all available readers on disk and on diskpack as well. RNP = activates the available readers only on diskpack.</p>

CONTROL CTL	These are synonymous and allow positioning to any control statement within the pseudo deck. If you omit the CONTROL clause, processing begins with the first card in the reader.
<integer>	<p>This specifies the card number of the control card where deck processing is to begin. This number can be obtained from the information supplied by the MCP when the PCRM option is set.</p> <p>The PCRM option, if set, causes the display of the location and contents of the first control card and each DATA/DATAB in a control card group as they are processed from pseudo decks.</p>
SKIP <integer>	This designates the number of data records to be skipped before the first data file in the pseudo reader is to be assigned.
SAVE	This specifies that the deck is to be retained when processing is completed; otherwise, the reader is purged when all statements have been read.

Note: A pseudo deck created by a CANDE "COPY TO CTLD" process can be activated only on the system on which it was created, and while the CANDE system is running.

Examples

RN =

This activates all pseudo card decks on disk.

RN 20002

This activates pseudo card deck number 20002, which is on disk.

RN 18 SAVE

This activates pseudo card deck number 18 and saves the deck after processing is finished.

RNP 2

This activates pseudo card deck number 2, which is on diskpack.

See also:

- The CD, CDP (List Inactive Pseudo Decks) command on Page 2-68.
- The DA, DAP (Deactivate Pseudo Reader) command on Page 2-103.
- The OL (Display Peripheral Status) command on Page 2-248.
- The RD, RDP (Remove Pseudo Decks) command on Page 2-313.
- The SD (Set Deck Limit) command on Page 2-342.
- The LIMIT READERS, USE PCRM, and USE APCR records in the *V Series Systems Operations Guide Volume 1: Installation*.

RO (Reset System Options)

Keyboard Input Message

Access Level: 8

The RO command resets (turns off) system options.

```
- RO <option name>
```

Explanation

<option name>

This specifies a system option that was set either through one of the USE records of the system configuration file or through the SO (Set System Options) command.

See Table 2-8 under SO (Set System Options) for a list of the available options.

Details

To see the status of a system option, you can use the TO (Type System Options) command.

Resetting the DEBUG, DCOM, DCP, DMS2, MICR, QWIK, SYST and TRAK options causes the MCP to release the memory space used by the designated extension module. These modules can be loaded at any time. However, certain modules, such as MICR, are not released until all users of the modules no longer require that module.

The Remote Job Entry (RJE) module is loaded at system initialization time because the MCP requires portions of it to handle Work Flow Language (WFL) programs. Resetting the RJE option causes the MCP to disable only the RJE handler portion of the RJE module.

Examples

```
RO DCOM
```

This resets the DCOM option that loads a data comm module into memory when a data comm device is activated. The MCP will release the memory space used by the DCOM module.

```
RO TPLB, OPEN, CLOS
```

This resets the TPLB option that displays a message each time a tape file becomes available; the OPEN option that displays a file open message each time a program opens a file; and the CLOS option that displays a file closed message each time a program closes a file.

See also:

- The SO (Set System Options) command on Page 2-352.
- The TO (Display System Options Status) command on Page 2-379.
- The USE records in the *V Series Systems Operations Guide Volume 1: Installation*.

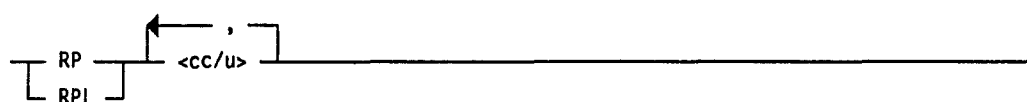
RP (Ready and Purge Magnetic Tape)

Keyboard Input Message

Access Level: 7

The RP command combines the RY (Ready Peripheral Device) and PG (Purge Magnetic Tape) commands to make a magnetic tape unit ready and to purge the tape. The RP command also has additional capabilities that allow you to assign the tape to a mix number when the tape is purged and/or to override a program's tape type request.

Purge and Ready a Tape



Explanation

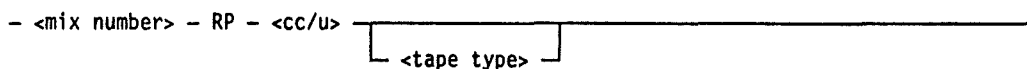
RP
RPL

This indicates that you want to ready and purge the specified tape unit. Use RPL to lock the tape unit after it is purged, preventing its inadvertent assignment to a program.

<cc/u>

This is the channel number and unit number of the magnetic tape unit to be readied and purged.

Purge, Ready, and Assign a Tape to a Mix Number



Explanation

<mix number>

This represents the mix number of a job. If the specified job is waiting for an output tape, the tape is purged and assigned to this job.

<tape type>

This overrides a tape type request of a program. You can specify the <tape type> as MT9, MPE, GCR or MTC.

Examples

RP 4/1, 4/2, 4/5

This readies and purges the magnetic tapes on channel number 4, unit numbers 1, 2, and 5.

RPL 4/3

This readies, purges, and locks magnetic tape unit number 3 on channel number 4.

113 RP 6/1 MPE

This readies and purges the magnetic tape unit on channel number 6, unit number 1 with a tape type of MPE and assigns it to the job with mix number 113.

See also:

- The AC (Purge and Place Tape Number in Label) command on Page 2-2.
- The PG (Purge Magnetic Tape) command on Page 2-282.
- The RY (Ready Peripheral Device) command on Page 2-338.
- The SN (Purge and Place Tape Number in Label) command on Page 2-350.

RQ (Remove Storage Queue Entries)

Keyboard Input Message

Access Level: 8

RQ purges the storage queue (STOQ) of all queue entries in a specific queue or all entries in all queues.

The contents of the storage queue can be determined through the WQ (Display STOQUE Count) command.

```
- RQ [ = _____ ]
      [ <queue name> ]
```

Explanation

= This removes all entries from all queues.

<queue name> This is the name associated with a group of STOQ entries.

Explanation of Storage Queue (STOQ)

A storage queue (STOQ or STOQUE) is a buffer in memory used to transfer information asynchronously from one program to another. This means that if a program sends a message to another program through STOQ and that program is not ready to accept the message, the message is stored in the storage queue until the other program is ready to retrieve it.

The receiving program need not be in the mix when the message is sent. The receiving program retrieves its messages, one at a time.

A storage queue is created when a program puts a message in it or requests a message from it. It continues to exist as long as there are messages in it, or as long as a program is awaiting a message from it.

Each storage queue has a name, assigned by the programs according to the rules of the programming language. The sending program and the receiving program must, in general, agree on this name.

Examples

```
RQ =
```

This removes all entries from the STOQ.

```
RQ JACK
```

This removes the group of STOQ entries associated with the queue named JACK.

See also:

- The WQ (Display STOQUE Count) command on Page 2-420.
- Complete details on storage queues are given in the various language manuals and in the *V Series MCP/VS Programming Reference Manual*.

RS (Remove Job from Schedule)

Keyboard Input Message

Access Level: 1

The RS command removes a scheduled job request from the MCP schedule list.

- <mix number> - RS

Explanation

<mix number>

This designates a job that is in the schedule but not yet in the mix.

If the AFTR system option is set, all programs to be executed after the indicated program and all successors to these jobs are also removed.

If AFTR is not set, successors to the job being removed with an RS are not disturbed.

Removing All Scheduled Jobs

To remove all scheduled jobs from the MCP schedule list, use the syntax shown in the following figure.

- RS=

Examples

12 RS

This removes job with schedule number 012 from the MCP schedule list.

RS =

This removes all scheduled jobs.

See also:

- The RA (Remove After-Linkage) command on Page 2-312.
- The WS (Display Jobs in Schedule) command on Page 2-423.
- The USE AFTR record in the *V Series Systems Operations Guide Volume 1: Installation*.

RUN (Initiate Execution of Program)

MCP Control Instruction

Access Level: 1

The RUN command initiates the execution of a program under timesharing.

- RUN - <program name> _____

Explanation

<program name> This must be the name of a program, up to six characters long, in the system Process Information Block (PIB); that is, the program must have been previously declared in a PROGRAM record in the CANDE initialization file.

Details

The program is subject to the standard restrictions applied to timesharing processes, except that it is not attached to any terminal.

Programs initiated with RUN are not placed in the schedule but enter the mix immediately. If timesharing is terminated, any programs running under timesharing are terminated.

Programs initiated through Work Flow Language (WFL), Remote Job Entry (RJE), and BNA network architecture cannot be run in the timesharing area. RUN is invalid with RJE and BNA and initiates a batch-area execution with WFL.

Example

```
RUN CPTEST
```

This executes the program named CPTEST under timesharing.

See also:

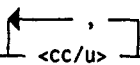
The *V Series CANDE Installation and Operations Guide* for information on timesharing.

RW (Rewind and Unload Tape)

Keyboard Input Message

Access Level: 7

The RW command unloads a tape residing on the specified tape drive. If the drive is declared SHARED, it will also be saved. The unload function works only for tape drives that have the automatic unload feature.

- RW  _____

Explanation

<cc/u>

This is the channel number and unit number of the magnetic tape unit.

Examples

RW 6/1, 6/2

This saves and unloads the tapes on channel 6, unit numbers 1 and 2.

See also:

The SV (Save a Peripheral Unit) command on Page 2-368.

The range for <begin address> is 0 through 999999; the range for <length> is 1 through 999999. Both <begin address> and <length> are always specified in decimal.

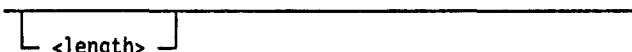
Example

RXD 3 5275

This returns the space on disk with ID 3 at address 5275.

RXP (Return Diskpack Space)

The RXP command returns diskpack space that was previously deleted by an XP (Remove Pack Sectors) command.

- RXP - <cc/u> - <begin address> 

Explanation

<cc/u>

This is the channel number and unit number of the diskpack unit.

<begin address>
<length>

The <begin address> must specify the beginning address of an area removed by an XP command or no space will be returned. If there is a match, the entire space of the removed area is returned independent of the optionally specified <length>.

The range for <begin address> is 1 through 16700000; the range for <length> is 1 through 16700000. Both <begin address> and <length> are always specified in decimal.

Example

RXP 1/9 2500

This returns the space at address 2500 on the diskpack on drive 1/9.

See also:

- The WX (Display Removed Space) command on Page 2-425.
- The XD (Remove Disk Segments) command on Page 2-436.
- The XM (Remove Memory from System) command on Page 2-437.
- The XP (Remove Pack Sectors) command on Page 2-438.

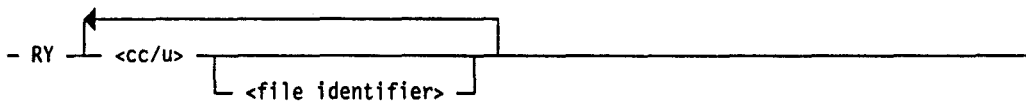
RY (Ready Peripheral Device)

Keyboard Input Message

Access Level: 7

The RY command clears the status of an unassigned peripheral unit and makes it available to the MCP.

The RY command resets all exception flags maintained by the MCP for the specified unit and makes locked or saved units available. The MCP then reads the device as applicable. For example, if the unit is a tape, it reads the tape label. If the unit is a train printer, it loads the translate table file. If the unit is a card reader, it reads the first card.



Explanation

<cc/u>

This is the channel number and unit number of the peripheral unit to be made available.

<file identifier>

This is for train printers and specifies the name of the translate table file to be loaded into the train printer buffer. File identifiers can be up to six characters long and cannot consist of all nulls, all blanks or all zeros.

You can use your own translate table files or use those bound to the MCP.

An RY command to a train printer always reloads the current (or specified) train buffer file.

An RY command to a train printer in use can specify only the train buffer file currently in use on the printer. This makes it impossible to accidentally change the train format during a print run.

Note: *When a shared tape unit is made ready on two systems at the same time, any job using the tape will incur a rewind error and be discontinued.*

Examples

RY 10/2

This clears the status of the unassigned peripheral on channel number 10, unit number 2.

RY 3/4, 3/5, 1/0

This clears the status of the unassigned peripherals on channel number 3, unit numbers 4 and 5, and the peripheral on channel number 1, unit number 0.

RY 16/0 PRN123

This clears the status of the unassigned train printer on channel number 16, unit number 0 and loads the translate table file called PRN123 into the train printer buffer.

See also:

- The RP (Ready and Purge Magnetic Tape) command on Page 2-329.
- The PG (Purge Magnetic Tape) command on Page 2-282.
- The SV (Save a Peripheral Unit) command on Page 2-368.

SB (Interrogate Database Activity)

Keyboard Input Message

Access Level: 8

The SB command allows you to obtain information about database activity. You can display information about the following:

- One particular active database
- All jobs currently accessing a particular active database
- All active databases on the system

Specifying Mix Number

```
SB _____  
└─<mix number>┘
```

Explanation

<mix number> This causes the status of the active database program associated with that database user program to be displayed.

SB This specified by itself without the <mix number> causes the system to display the names of all the currently active databases.

Specifying Database Name

```
SB _____  
└─<database name>┘
```

Explanation

<database name> This causes the status of the active database program and all its active database user programs to be displayed.

Examples

User Input	System Response
12 SB	P=7 M=7 DMS/DATAB1=032 WTG DMS FUNCTION
SB TESTDB	P=4 M=4 PROGD1=020 EXECUTING
	P=4 M=4 TESTV1=005 WTG DMS FUNCTION
SB	SB= DATAB1 TESTDB DBACC5

See also:

- The SP (Display/Set/Change Database Parameters) command on Page 2-356.
- *The V Series DMSII Operations Reference Manual Volume 2: Database Administration.*

SD (Set Deck Limit)

Keyboard Input Message

Access Level: 8

SD limits the number of decks that the MCP Automatic Pseudo Card Reader (APCR) function can have active at one time.

- SD - <integer>

Explanation

<integer>

This can be a value from 0 through 80 and indicates the maximum number of readers that APCR can have active simultaneously.

If an SD command is not supplied to the system, the default value of 80 is assumed.

Note that the specified value is not affected by a halt/load operation.

Details

The APCR option can be set through the USE APCR record of the system configuration file or with an SO (Set System Options) command. The APCR option directs the MCP to automatically place pseudo card decks into pseudo card readers as the decks become present on disk.

If the APCR option is not set, use the RN, RNP (Activate Pseudo Reader) command to manually accomplish this function.

You can use the DA, DAP (Deactivate Pseudo Reader) command to deactivate pseudo card readers.

Example

SD 60

This limits the number of decks the APCR function can have active at one time to 60.

See also:

- The DA, DAP (Deactivate Pseudo Reader) command on Page 2-103.
- The RN, RNP (Activate Pseudo Reader) command on Page 2-324.
- The SO (Set System Options) command on Page 2-352.
- The LIMIT READERS and USE APCR record in the *V Series Systems Operations Guide Volume 1: Installation*.

Explanation

BACKUP

This displays the following information:

- The current primary and secondary backup families where backup files are stored by default
- The next backup numbers to be assigned to printer and punch backup files
- The lowest backup numbers used by operating system intrinsics (PBDPRN, BFRF and so forth) when searching for printer and punch backup files
- The number of records in each block (PRNRPB) and each area (PRNRPA) of printer backup files

BLT

This displays an entry from a Block Lockout Table (BLT), or displays an entire BLT. A Block Lockout Table is a reference table showing which disk/diskpack blocks are locked by which programs.

<BLT number> This indicates a specific Block Lockout Table entry.

EXCHANGE <exchange number>

This displays all the channels on an exchange and some information about them. The <exchange number> is the number of the exchange (0 through 77) for which you want the system to display information.

FILE

SHOW FILE displays information about a diskpack or disk file.

<file> This is the name of the diskpack or disk file.

<family> This is the family name of a diskpack file.

IRS

This displays information about all active independent runners.

PROCESSOR

This displays the following information about a single processor system, or each processor in a multiprocessor system:

- System number
- Hostname
- Processor number
- Percentage of the CPU available
- State

SHARED	<p>This displays the following information about a non-shared system or each system in a multi-system shared configuration:</p> <ul style="list-style-type: none">• System number• Hostname• Amount of memory available• Percentage of CPU available• Number of jobs• State
SSP	<p>This displays information about an SSP or all SSPs on the system.</p> <p><SSP channel number> This indicates a specific SSP.</p>
SYSTEM	<p>This displays the following system and coldstart information:</p> <ul style="list-style-type: none">• System number• Hostname• Current MCP file• Current configuration file• Maximum tasks• Maximum IRs• Last coldstart• Last halt/load <p>If you do not specify a system number, the default is the current system.</p> <p>ALL This repeats the information listed above for all systems.</p> <p><n> This indicates the number of the system you want information about. For the SHOW SYSTEM command, system numbers start with 0 (zero).</p>
TASK	<p>This displays information about a task.</p> <p><task number> This is the mix number of the user task about which you want the system to display information. Task numbers can range from 2 through 775.</p>

SHOW

TASKS	This displays information about all user tasks.
TRAK	This displays the current TRAK specifications (determined by the most recently entered BT command). If no BT command has been entered, a message is displayed indicating that data is not being inserted into the TRAK buffer.
VCS	This displays information about internal data structures used by the V Series Communication System (VCS). This SHOW option is primarily used by Unisys for maintenance of the VCS system.

Examples

SHOW IRS

This displays information about all operating system tasks (independent runners).

SHOW SYSTEM 0

This displays system and coldstart information about system 0.

SHOW TASKS

This displays all user tasks.

See also:

- The ALTER (Change System Parameters) command on Page 2-21.
- The DBDR (Produce SSP Content Report) command on Page 2-110.
- The DBSR (Produce Block Lockout Table Report) command on Page 2-114.

SI (Set Remote Device Identification)

Keyboard Input Message

Access Level: 8

The SI command changes the identification of a remote device.

```
- SI [ <cc/u> _____ ] <new unit ID> _____
      [ <current unit ID> ]
```

Explanation

<cc/u>	This is the channel number and unit number of a data communications device.
<current unit ID>	This is the current name of the device. The name can be one to six characters long. The first character must be alphabetic, but the remaining characters can be alphabetic or numeric.
<new unit ID>	This is the name you want the device to have. The name can be one to six characters long. The first character must be alphabetic, but the remaining characters can be alphabetic or numeric.

Details

Note that a subsequent halt/load operation will reset the unit ID to its original UNIT record specification. See UNIT records in the *V Series Systems Operations Guide Volume 1: Installation* for more information.

Examples

```
SI 21/0 REMOT2
```

This changes unit ID of the remote device on channel number 21, unit number 0 to REMOT2.

```
SI REMOT1 REMOT2
```

This changes the unit ID of the remote device named REMOT1 to REMOT2.

See also:

- The DL (Delete Peripheral Devices) command on Page 2-135.
- The UNIT (Add Peripheral Devices) command on Page 2-385.
- UNIT records in the *V Series Systems Operations Guide Volume 1: Installation*.

SK (Skip Backup Records)

Keyboard Input Message

Access Level: 3

The SK command designates a number of records to be skipped during printing of a printer backup file (on tape or disk) or punching of a punch backup disk file.

You cannot use SK on a locked program.

- <mix number> - SK <integer>

Explanation

- | | |
|--------------|--|
| <mix number> | This identifies the PBDPRN or PCHOUT program that is to skip the number of records denoted by the <integer>. |
| - | This is a minus sign, which when it precedes <integer> indicates that the records should be skipped backwards in the file. Up to 999999 records can be skipped backward. |
| <integer> | This specifies the number of records to be skipped. Up to 999999 records can be skipped forward with each execution of this command. |

Note: The number of records in the file is not necessarily equivalent to the number of lines of listing it produces.

Examples

007 SK-350

This causes the PBDPRN or PCHOUT program associated with mix number 007 to skip back 350 records while printing or punching a backup file.

164 SK 1000

This causes the PBDPRN or PCHOUT program associated with mix number 164 to skip forward 1000 records while printing or punching a backup file.

See also:

- The PC, PCP (Create Punch Backup File) command on Page 2-275.
- The PRINT (Print Files) command on Page 2-291.
- The QT (Quit Program Operation) command on Page 2-308.

SM (Send System Message)

Keyboard Input Message

Access Level: 0

SM sends a message to a Message Control System (MCS) program or to controller programs of the V Series Communication System (VCS). The MCS programs can be for B 974, or B 874 data communications devices. They must be executing in the mix and have a queue attached. However, the queue cannot be full. The VCS controller programs include the Network Controller (NC) and the Application Controller (AC).

- <mix number> - SM - <text>

Explanation

<mix number> This is the mix number of the desired program. You can obtain the mix number with the AJ (Display Active Jobs) command. MCS programs retrieve text from their queues by issuing a READ MCS QUEUE function (function 81). VCS controller programs receive text through a "non-confirmed data indication" (type 22).

<text> This is the text of your message and can consist of any letters, numbers, and special characters except periods.

The maximum text length is 79 bytes.

Example

```
20 SM SYSTEM WILL BE COLDSTARTED AT 8 PM TONIGHT
```

This sends the message to the MCS with mix number 020.

See also:

- The SS (Send Message) command on Page 2-363.
- The *B 2000/B 3000/B 4000/V Series GEMCOS Implementation And Programming Reference Manual*.

SN (Purge and Place Tape Number in Label)

Keyboard Input Message

Access Level: 7

The SN command purges a tape and puts an external physical tape number into a magnetic tape label to permanently identify it.

The SN command also has additional capabilities that allow you to assign the tape to a mix number when the tape is purged and/or to override a program's tape type request.

Purge a Tape and Assign a Tape Number

Explanation

- SNL** This locks the tape after purging it. SNL and <mix number> cannot be used in the same statement.
- <cc/u>** This is the channel number and unit number of the appropriate tape drive. The tape must be ready and contain a write ring.
- <physical tape number>** This is placed into the label of the designated tape reel and can be any integer up to five digits in length. The <physical tape number> remains in the label throughout the life of the tape or until you perform another SN or AC operation on that tape.

Purge and Assign a Tape to a Mix Number

Explanation

- <mix number>** This causes the tape on the drive designated by <cc/u> to be purged and assigned to the job with this mix number.
- If the tape has GCR capability and no tape type is specified, GCR mode will be used.

<tape type> This overrides a user program's tape type request.
The <tape type> can be MT9, MPE, GCR or MTC.

Details

If you do not specify the <mix number> and <tape type> *and* you are using a GCR-capable drive, the tape will be purged and made available in GCR mode.

Examples

SNL 4/1 22

This purges the magnetic tape on channel number 4, unit number 1, places the number 22 in its label, and then locks the tape.

SN 46/1 123, 46/2 456

This purges the magnetic tape on channel number 46, unit number 1 and places the number 123 in its label; purges the magnetic tape on channel 46, unit number 2 and places the number 456 in its label.

29 SN 12/3 GCR 84127

This purges the GCR magnetic tape on channel 12, unit number 3, assigns it to mix number 029, and places the number 84127 in its label.

See also:

- The AC (Purge and Place Tape Number in Label) command on Page 2-2.
- The CN (Display Tape Number) command on Page 2-78.
- The LABEL1 (Change Tape Label Format) command on Page 2-212.
- The PG (Purge Magnetic Tape) command on Page 2-282.
- The RP (Ready and Purge Magnetic Tape) command on Page 2-329.
- The TM (Write Tape Mark) command on Page 2-377.

SO (Set System Options)

Keyboard Input Message

Access Level: 8

The SO command sets (turns on) certain system options.

You can also set system options through the USE records of the system configuration file. See the *V Series Systems Operations Guide Volume 1: Installation* for more information.

You can reset (turn off) system options with the RO (Reset System Options) command or display the status of a system option with the TO (Display System Options Status) command.

```
- SO <option name>
```

Explanation

<option name> This specifies one or more of the system options described in Table 2-8.

Table 2-8. System Options Available For Setting with SO

Option	Description
AFTR	Automatically removes scheduled jobs that are after-linked to an abnormally terminated job
APBD	Automatically prints and saves backup files after they are closed
APCR	Automatically places pseudo card decks into pseudo card readers as the decks become present on disk or diskpack
AUHL	Automatically initiates a halt/load in case of system failure
AURD	Automatically removes pseudo card readers from disk if the files have erroneous control instructions
BOJ	Displays Beginning-Of-Job (BOJ) messages each time a job enters the active mix
CHRG	Specifies that all programs must have a charge number in order to execute
CLOS	Displays a file-closed message each time a program closes a file
COPY	Displays a message each time SYSTEM/COPY compares or transfers a file
DATE	Requires a date to be entered in order to halt/load

Table 2-8. System Options Available For Setting with SO (cont.)

Option	Description
DEBUG	Loads the interactive DEBUG module to trace the operating system or a user program
DCOM	Loads a data comm module into memory when a data comm device is activated
DCP	Loads a Data Communications Processor (DCP) into memory
DMS2	Loads the DMS2 extension module into memory
EOJ	Displays an End-Of-Job (EOJ) message each time a program goes to an EOJ
GTDK	Forces program trace requests to backup disk
ISC	Allows programmatic use of the Inter-System Connect (ISC) for inter-system communications
LIB	Displays a message each time a file is removed or changed
MICR	Loads a reader-sorter module into memory
OPEN	Displays a file open message each time a program opens a file
PBD	Routes or directs printer backup files to disk
PBP	Routes or directs printer backup files to diskpack
PBPS	Directs all areas of printer and punch backup to a single diskpack
PBT	Directs printer backup files to a scratch tape or a tape already in use for backup on a ready drive
PCD	Indicates that punch card backup files are to be built on disk
PCH	Directs punch files to the card punch without creating backup files
PCP	Indicates that card backup files are to be routed to diskpack
PCRM	Displays the location and contents of the first control card and each DATA(B) in a control card group as they are processed from pseudo card decks
PRN	Directs printer files to the printer; no printer backup file is made
QWIK	Loads the QWIK module, which causes user program overlays to be stored and retrieved from a memory buffer
RJE	Enables the Remote Job Entry (RJE) handler to be executed when an RJE station becomes active
RMOV	Resolves duplicate file conditions on disk or pack by indicating that the old file is to be removed
RTSL	Indicates that short or long tape block reads that are not multiples of the logical record size are to be treated as I/O errors
SCHM	Causes a message to be displayed each time a program is scheduled for execution

Table 2-8. System Options Available For Setting with SO (cont.)

Option	Description
SNAP	Produces a SNAP picture or "snapshot" to capture the state of memory when certain processor errors occur
STGO	Indicates that a message is to be displayed whenever an executing program stopped (such as with the ST or STOP (Suspend Program Processing) command), or when the program is restarted (such as with the GO or START (Resume Stopped Program) command)
SYST	Loads a module into memory in order to run FLAME
SYUP	Executes the SYSUP program after a cold start or halt/load
TERM	Terminates programs that have irrecoverable errors, otherwise they will go into a DS/DP condition.
TIME	Requires that the time be entered before a cold start or a halt/load completes
TPLB	Causes a message to be displayed each time a tape file becomes available
TPMK	Saves all tapes containing non-rewound reels with a write ring in the event of a halt/load
TRAK	Loads a TRAK module into memory and allocates a buffer for tracking information
TRMD	Dumps jobs that terminate abnormally
WRKP	Directs work files to unrestricted system resource diskpacks
ZIPM	Displays the message "<job ID> ZIP <control information message>" each time a program passes an instruction to the MCP

Details

Setting the DEBUG, DCOM, DCP, DMS2, MICR, QWIK, SYST and TRAK options causes the MCP to load an extension module into memory. These modules can be loaded at any time.

The RJE extension module is always loaded during system initialization because the MCP requires portions of it to handle Work Flow Language (WFL) programs. However, when you use SO to set the RJE option, the MCP enables the RJE-handling portions of the module and ensures that the DCOM module, which is also needed for RJE operations, is loaded as well.

Examples

SO PRN

This sets the PRN option that causes print jobs to be printed so that no printer backup files need to be created.

SO TPLB, ZIPM, PBD

This sets the TPLB option to have a message be displayed each time a tape file becomes available; sets the ZIPM option to have the message "<job ID> ZIP <control information message>" displayed each time a program passes an instruction to the MCP; and sets the PBD option to route or direct printer backup files to disk.

See also:

- The RO (Reset System Options) command on Page 2-327.
- The TO (Display System Options Status) command on Page 2-379.
- The USE records in the *V Series Systems Operations Guide Volume 1: Installation*.

SP (Display/Set/Change Database Parameters)

Keyboard Input Message

Access Level: 8

Use the SP command to display, set, or change the value of certain database parameters while the Database Program (DBP) is executing.

The DBP is the program that performs all functions necessary for accessing and updating the database to which it is attached. All user application programs for a particular database interface with the DBP, and the DBP interleaves the processing of each program as much as possible.

For a complete description of the syntax and usage of this command see the *V Series DMSII Operations Reference Manual Volume 2: Database Administration*.

See also:

- The SB (Interrogate Database Activity) command on Page 2-340.
- The *V Series DMSII Operations Reference Manual Volume 2: Database Administration*.

SPO (Simulate SPO through Card Reader)

MCP Control Instruction

Access Level: 0

The SPO command provides capability for keyboard input messages and text to go to the MCP through a card reader as if they had been entered at the ODT. This is similar to the ZIP construct allowed in an executing program by the various programming languages.

Any response to the message is sent to the ODT.

- SPO - <text> _____|

Explanation

<text>

This can be any valid keyboard input message, with the following restrictions:

- If a ? SPO card is entered in a card deck (including CTLDCK), it is processed immediately and does not cause EOF action on the executing program.
- ? SPO HL is invalid.
- SPO is invalid in a pseudo reader.

Details

The most common usage of SPO and its counterpart, the CC (Enter Control Instructions) command, is to send control information from a user program to the MCP using the ZIP facility or SPOMESSAGE (SPOM) BCT. ZIP assumes control text, so to enter a keyboard input message, you must prefix it with SPO. SPOM assumes keyboard input, so control text must be prefixed with CC. You can use the SPO and CC commands to switch from one mode to the other:

```
ZIP 'SPO RY 6/2; CC PFM LIST TAPE1 A.'
```

Examples

```
? SPO LD
```

This sends an LD (Create Pseudo Card Deck) command to the MCP. LD schedules the LDCNTL utility, which reads card files and converts them into pseudo card files on disk or diskpack.

SPO

? SPO RY 6/0

This sends an RY (Ready Peripheral Device) command to the MCP to clear the status of the peripheral on channel number 6, unit number 0 and makes it available to the MCP.

CALL SYSTEM ZIP USING ''SPO SV 7/0, 16/0.''

This is a line of code from a COBOL74 user program that causes the operating system to execute the system command contained within the object program. In this case, the SPO command sends an SV (Save a Peripheral Unit) command to the MCP. This SV command saves the peripheral devices associated with channel number 7, unit number 0 and channel number 16, unit number 0.

See also:

The CC (Enter Control Instructions) command on Page 2-66.

SQ (Squash Disk)

Keyboard Input Message

Access Level: 8

The SQ command instructs the MCP to consolidate all disk files into a contiguous area starting at the lowest address of the disk on which they currently reside. In this way, all available disk space on each disk is made contiguous.

This command is not available for MCP/VS 3.1.

See also:

- The SQP (Squash Pack) command on Page 2-360.

SQP (Squash Pack)

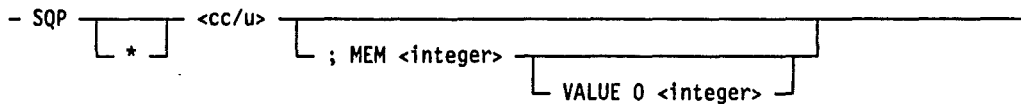
Keyboard Input Message

Access Level: 8

The SQP command invokes the SQUASH utility, a bound intrinsic program that can perform SQUASH and/or COMPLEMENT functions on a designated diskpack as follows:

- SQUASH consolidates all areas on a designated diskpack.
When an area is moved, the table links, directory, and file headers are appropriately updated.
- COMPLEMENT reconstructs the working available table from the information in the master available table and the directory after the squash phase is complete. It can also be invoked as a single function. See the VALUE specifications for more information.

Note: If the SQP command is used on a Version 2 diskpack family, the system must have exclusive access to the base pack of the family in order to access the file directory information. For more information on pack family versions, refer to "Pack Subsystems" in the V Series Systems Operations Guide Volume 3.



Explanation

***** This indicates that a recovery tape is to be created or loaded before the squash of the diskpack actually begins. This tape consists of the diskpack label, master available table, working available table, directory table, all file headers, and all file areas on the diskpack.

<cc/u> This designates the channel number and unit number location of the diskpack to be squashed or complemented.

MEM <integer> This requests additional memory to be allocated for execution of the SQUASH program. Adding memory to the minimum required by SQUASH speeds up the SQUASH function. It also slightly increases the speed of the tape dump and of the COMPLEMENT function. A value expressed by three or fewer characters is assumed to be kilodigits (for example, 80 = 80 kd); a value expressed by four to six characters is assumed to be digits (for example, 80000 = 80,000 digits).

VALUE 0

This allows you to request a diskpack squash or diskpack complement by setting SW5 of the SQUASH program prior to its execution. The possible entries for VALUE are as follows:

Value	Meaning
0	Perform squash and compare any recovery tape created or loaded.
1	Skip comparison of any recovery tape created or loaded.
2	Create and compare recovery tape but do not squash diskpack.
3	Only create recovery tape. Do not compare tape or squash diskpack.
C	Force diskpack complement even if the diskpack label indicates that pack recovery is required.
D	Load diskpack from recovery tape ignoring the current recovery state of the diskpack but validating the recovery tape header record against the diskpack label. Thus, the diskpack label cannot be smashed for this to work.
F	Only perform a diskpack complement.

Details

The diskpack must not be in use while a SQUASH operation is attempted. If any files are open on the designated diskpack when SQP is entered, the MCP will display the following message:

** KBD IGNORED: UNIT IN USE

When SQUASH is executed, it immediately saves the entire diskpack. The operator should not save the diskpack before entering SQP. No other programs can access the diskpack until SQUASH completes.

In the case of processor or program exceptions, the SQUASH program will be reinstated automatically at its own recovery routine. This ensures that a memory dump for recovery purposes is available if you discontinue the SQUASH program.

SQUASH should not be attempted on a diskpack where a PS= indicates errors. Only DISPKV can clear up such a situation. See the *V Series Systems Operations Guide Volume 3: System Utilities* for more information.

SQUASH is not permitted on a shared diskpack. If SQP is entered for a shared diskpack, the MCP will respond with the following message:

**INVALID UNIT.

Examples

SQP 18/2

This invokes the SQUASH utility for a diskpack with a channel number of 18 and a unit number 2.

SQP * 18/03; MEM 80000

This invokes the SQUASH utility for a diskpack with a channel number of 18 and unit number 3, allocates another 80000 digits of memory for the process, and requests that a recovery tape be created, the diskpack squashed, and the tape compared.

See also:

The DISPKV utility in *V Series Systems Operations Guide Volume 3: System Utilities*.

SS (Send Message)

Keyboard Input Message

Access Level: 0

The SS command directs messages from one ODT to another.

```
- SS [ ALL | <unit ID> ] <text>
```

Explanation

ALL This directs a message from an ODT to all other ODTs on the system.

<unit ID> This specifies the name of the destination ODT. The name should be one declared in the UNIT ODT or UNIT OCS record of the system configuration file and should be one to six characters long. The first character must be alphabetic, but the remaining characters can be alphabetic or numeric.

<text> This is the text of the message.

Example

```
SS ALL GO OFF LINE PLEASE
```

This sends the message "GO OFF LINE PLEASE" to all ODTs on that line.

See also:

- The SM (Display System Message) command on Page 2-349.
- The UNIT ODT and UNIT OCS records in the *V Series Systems Operations Guide Volume 1: Installation*.

ST (Suspend Program Processing)

Keyboard Input Message

Access Level: 3

The ST command stops the execution of a program and makes the memory it occupies the prime candidate for rollout to disk if another program is waiting for memory.

Rollout to disk is not done automatically. The program is removed from the ready-to-run list, and will be rolled out only if memory is required by a task.

The ST command has the same effect as that of the STOP (Suspend Program Processing) command.

The stopped program can be restarted with a GO (Restart a Stopped Program) or START (Restart a Stopped Program) command.

- <mix number> - ST _____|

Explanation

<mix number> This is the mix number of the program to be suspended.

Details

The current state of the program is saved for use when the program is restarted.

The MCP suspends execution of the designated program when all I/O operations in progress for that program are completed.

The execution of a GO (Restart a Stopped Program) command or START (Restart a Stopped Program) command causes the MCP to resume execution of the program.

If the program to be stopped is not currently executing, the ST command is ignored.

Example

32 ST

This stops the program associated with mix number 032.

See also:

- The GO (Restart A Stopped Program) command on Page 2-182.
- The START (Resume Stopped Program) command on Page 2-365.
- The STOP (Suspend Program Processing) command on Page 2-367.

START (Restart a Stopped Program)

MCP Control Instruction

Access Level: 3

The START command restarts a program that was stopped by the ST (Suspend Program Processing) or STOP (Suspend Program Processing) command.

The GO (Restart A Stopped Program) command has the same effect as the START command.

- START - <program name> _____|

Explanation

<program name> This is the name of the program to be restarted.

Details

The program restarts its execution at the point where it stopped. It starts with the same values it was using when it stopped.

If you issue START on a program that is not stopped, the system ignores the command.

If there is not enough memory for the program being restarted, the following rules apply:

- Memory is allocated by priority. If you raise the program's memory priority, it might restart.
- When sufficient memory becomes available, the program will restart.
- A program being restarted will start before a program of the same priority that is waiting in the schedule.
- Programs with lower memory priority are rolled out from memory to disk if this will make enough room for the program that is waiting to restart.
- If two or more programs with the same name have been stopped, the START command starts the one with the highest processor priority. If the stopped programs have the same priority, it is unpredictable which will be restarted.

Example

```
START PAYROL
```

This restarts a program named PAYROL.

START

See also:

- The GO (Restart a Stopped Program) command on Page 2-182.
- The ST (Suspend Program Processing) command on Page 2-364.
- The STOP (Suspend Program Processing) command on Page 2-367.

STOP (Suspend Program Processing)

MCP Control Instruction

Access Level: 3

The STOP command stops an executing program and makes the memory it occupies a prime candidate for rollout to disk if another program is waiting for memory.

Rollout to disk is not done automatically. The program is removed from the ready to run list, and will be rolled out only if memory is required by a task.

The STOP command has the same effect as that of the ST (Suspend Program Processing) command.

- STOP - <program name> _____

Explanation

<program name> This is the name of the program to be stopped.

Details

The current state of the program is saved for use when the program is restarted.

This command takes effect when all I/O operations for the program have completed. Peripheral devices in use by the program are not released to the system.

The execution of a GO (Restart a Stopped Program) or START (Restart a Stopped Program) command causes the MCP to resume execution of the program.

If the program to be stopped is not currently executing, the STOP command is ignored.

If two or more programs with the same name are active when you issue the STOP command, the one with the highest priority will stop. If priorities are equal, it is unpredictable which one will stop.

Example

```
STOP PAYROL
```

This stops a program named PAYROL.

See also:

- The GO (Restart a Stopped Program) command on Page 2-182.
- The START (Restart a Stopped Program) command on Page 2-365.
- The STOP (Suspend Program Processing) command on Page 2-367.

SV (Save a Peripheral Unit)

Keyboard Input Message

Access Level: 7

The SV command saves a peripheral unit, preventing the system from accessing it.

The unit can be made ready again by a halt/load or with an RY (Ready Peripheral Device) or UR (Inhibit or Uninhibit Unit) command.



Explanation

<cc/u> This indicates the channel number and unit number of the peripheral to be saved.

Details

If a program is using a unit when the SV command is entered, the unit is not saved until the program releases it.

You can save a unit to prevent other programs from accessing it. Ready or uninhibit the unit when you want it to be used.

The SV command may be used to transfer a unit from one system to another if the unit is physically connected to both systems. Save the unit on one system, and then ready it on the other.

Examples

SV 1/5

This saves the unit on channel number 1, unit number 5.

SV 1/8, 1/5, 4/0

This saves the unit on channel 1, unit 8, the unit on channel 1, unit 5, and the unit on channel 4, unit 0.

See also:

- The RY (Ready Peripheral Device) command on Page 2-338.
- The UR (Inhibit Or Uninhibit Unit) command on Page 2-399.

SW (Set Programmatic Switches)

Keyboard Input Message

Access Level: 1

The SW command sets the programmatic switches in an executing program. A programmatic switch is a feature of some programming languages such as COBOL.

```

- <mix number> - SW - <switch number> - <setting>

```

The diagram shows the command format: `- <mix number> - SW - <switch number> - <setting>`. Brackets are used to group the components: a bracket under `<mix number>`, a bracket under `SW`, a bracket under `<switch number>`, and a bracket under `<setting>`. An arrow points from the `<setting>` bracket back to the `SW` command, indicating that the setting applies to the switch.

Explanation

- | | |
|------------------------------------|--|
| <code><mix number></code> | This is the mix number of the executing program. |
| <code><switch number></code> | This is the number of the switch whose value you want to change. The switch number can be from 1 to 8. |
| = | This is optional and does not influence the command. |
| <code><setting></code> | This indicates the value of the switch setting and can be a number from 0 through 9. |

More than one switch can be set by a single SW command. If `<setting>` is longer than one digit, the first digit sets the switch designated by the `<switch number>`. The subsequent digits set the subsequent contiguous switches. For example, `012 SW 1 34` sets switch 1 to a value of 3 and also sets switch 2 to a value of 4.

Restrictions

You cannot use a mix number of 0 or 1, because that indicates the MCP.

Switches cannot be set on a program that is locked. See the LOCK (Lock Program) command.

Details

Switches 1 through 8 correspond to COBOL switches SW1 through SW8.

This command inserts integer values into the executing program code. The values for switch 8 and switches 1 through 7 are inserted into program base-relative addresses as shown in Table 2-9.

Table 2-9. Memory Locations of Programmatic Switches

Switch	Memory Location
8	000000
1	000001
2	000002
3	000003
4	000004
5	000005
6	000006
7	000007

The SW command cannot affect locations beyond base-relative address 7. If you need to insert values beyond base-relative address 7, use the IN (Insert Data into Program), INSERT (Insert Data into Program), or VALUE (Insert Value into a Program) commands. IN can insert values past location 7 while a job is executing; INSERT and VALUE insert values prior to the execution of a job.

Examples

2 SW 4 1

This affects the job with mix number 002, setting switch 4 to a value of 1.

14 SW 2 109

This affects the job with mix number 014, setting switch 2 to a value of 1, switch 3 to 0, and switch 4 to 9.

See also:

- The IN (Insert Data into Program) command on Page 2-195.
- The INSERT (Insert Data into Program) command on Page 2-197.
- The LOCK (Lock Program) command on Page 2-227.
- The VALUE (Insert Value into a Program) command on Page 2-403.
- The discussions of <parameter 1> under the following:
 - The COMPILE (Compile Program) command on Page 2-80.
 - The DEBUG (Attach Program To DEBUG Session) command on Page 2-121.
 - The EXECUTE (Execute Library Program) command on Page 2-162.
- Programmatic switches are described further in the language manuals.

TEST (Mark a Test Operation)

MCP Control Instruction

Access Level: 1

The TEST command is used with a job initiation command such as EXECUTE or COMPILE. It marks the job as a test operation in the system Run log (RLOG). The TEST command has no effect on the scheduling or execution of any program.

- TEST _____

Details

This command affects the Run log by setting the 4-bit in the supplementary execution code field (RL-SXE, which is digit 71) of the Job Schedule record (type 3/0), Short Job Schedule record (type 4/0) and Beginning-of-Job record (type 6/0) for the job in the RLOG.

If one program starts another program using the ZIP construct, the TEST command can be included in the ZIP syntax. In this way an entire system of programs can be correctly flagged in the Run log, as well as the one originally executed.

Examples

```
EXECUTE PROG; TEST
```

This starts the program called PROG, marking the job in the RLOG as a test operation.

See also:

- The RERUN (Log a Rerun Operation) command on Page 2-319.
- The system Run log in the *V Series System Software Logging Operations Reference Manual*.
- The ZIP construct in the various language manuals.

TI (Display Time Statistics)

Keyboard Input Message

Access Level: 0

The TI command displays on the ODT the amount of system time used by a particular job.

- <mix number> - TI _____|

Explanation

<mix number> This is the mix number of the job about which you want to display time statistics.

Example

26 TI

This displays system time statistics for the job with mix number 026. The following figure shows a sample response to the TI command.

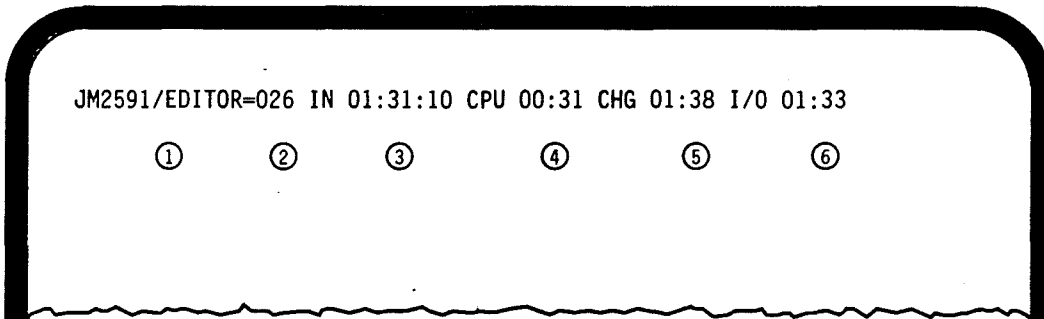


Figure 2-2. System Response to a TI Command

Each of the elements marked by circled numbers in Figure 2-2 is further described by the following.

Number	Description
①	This is the name of the job.
②	This is the mix number of the job.
③	This is the length of time the job has been in the mix.

Number	Description
④	This is the amount of direct processor time used by the job.
⑤	This is the amount of time that will be charged against the charge number associated with the job.
⑥	This is the amount of I/O wait time.

See also:

- The AJ (Display Active Jobs) command on Page 2-14.
- The DQ (Display ODT Queue) command on Page 2-145.
- The MX (Display Jobs in Mix) command on Page 2-236.
- The TIME (Discontinue Program after a Specified Amount of Time) command on Page 2-374.
- The WJ (Display Waiting Jobs) command on Page 2-411.

TIME (Discontinue Program After a Specified Amount of Time)

MCP Control Instruction

Access Level: 1

TIME discontinues a program (to EOJ) if it uses more than a given amount of processor time.

- TIME . <integer>

Explanation

= This is optional and does not influence the command.

<integer> This is the amount of processor time, in seconds, that the job is allowed to accumulate. It can be from one to five digits. If fewer than five digits are used, leading zeros are assumed.

Details

Processor time is the accumulated time the processor has spent working on the job. It is also called *direct time*.

You can use the TI (Display Time Statistics) command to find out the amount of system time used by a particular job.

Example

```
EXECUTE PAYROL; TIME 300
```

This starts the program called PAYROL. If the job uses more than 300 seconds of processor time, the job will be discontinued. Note that there is no limit to the amount of time the job can be in the mix; only the amount of processor time is monitored.

See also:

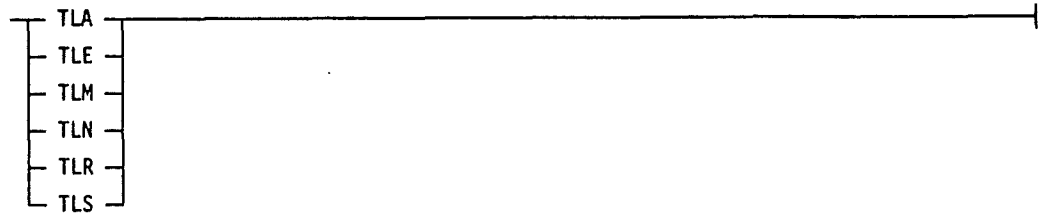
The TI (Display Time Statistics) command on Page 2-372.

TL (Transfer Log)

Keyboard Input Message

Access Level: 8

The TL command causes a log transfer. TL closes the current system log file and directs logging information into a new log file.



Explanation

TLA	This transfers all logs in use—the Run log, the Maintenance log, the ODT log, the Network log and (for V 500 Series systems) the ECM log.
TLE	This transfers the ECM log. The Environmental Control Module (ECM) log is created only on V 500 series systems.
TLM	This transfers the Maintenance log.
TLN	This transfers the Network log. The Network log is used by the V Series Communication System (VCS).
TLR	This transfers the Run log.
TLS	This transfers the ODT log.

Details

Log transfer happens automatically when a log file becomes full or at LOGXFERTIME. You can use the TL or LN (Transfer and Print Log) command to close the current log file so that you can analyze it, or to start a new log file at a particular time. This depends on the information you are trying to capture.

The LN (Transfer and Print Log) command transfers the log and starts the log analysis program. LN always starts the log analysis program; TL starts the log analysis program only if the AUTO option is set in the USE record associated with the log in the system configuration file.

The system configuration file can contain a USE record for each system log (USE RLOG, USE MLOG, USE NLOG, and USE SLOG). If a USE record for a particular system log is present, the log will be created.

If the AUTO option is set on the USE record for the log file, the appropriate log analysis and reporting program is started whenever that log is transferred, whether it is transferred automatically when it fills up or at LOGXFERTIME, or manually by the TL or LN command.

If the WRAP option is set on a the USE record for the log file, the log file is renamed each time it is transferred by incrementing a number in the file name from 0 to 9. For example, the name of the MLOG would change from MLOGs1 to MLOGs9 (where s is the system number). If this option is not set, the file names do not change, and the new log file will overwrite the another file at log transfer time. If this option is not used, the files will be renamed from MLOGs0 to MLOGs1.

Examples

TLR

This transfers the Run log.

TLA

This transfers all logs in use.

See also:

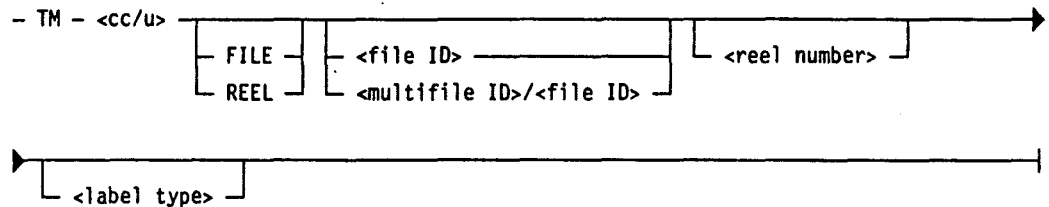
- The LN (Transfer and Print Log) command on Page 2-223.
- The USE LOG and CONTROL LOGXFERTIME records in the *V Series Systems Operations Guide Volume 1: Installation*.
- The *V Series System Software Logging Operations Reference Manual* for complete details on system logging.

TM (Write Tape Mark)

Keyboard Input Message

Access Level: 7

The TM command writes a tape mark and, optionally, an End-Of-File (EOF) or End-Of-Reel (EOR) label at the current point on a magnetic tape.



Explanation

<cc/u>	This is the channel number and unit number of the tape drive on which the tape is mounted. If TM <cc/u> is entered without any of the options, a tape mark without an ending label is written.						
FILE	This writes a tape mark and an EOF label. This is the default if the file identifier is entered						
REEL	This writes a tape mark and an EOR label.						
<file ID> <multifile ID> /<file ID>	These elements specify the name of the tape file. The name can be up to six characters (for example, MYFILE) or it can be a multifile ID (for example, MYTAPE/MYFILE). If either a file identifier or multifile ID/file ID is given, a tape mark and an EOF label are written.						
<reel number>	This is the number of the reel. The default is 1.						
<label type>	This specifies the kind of end label to write and allows the following values: <table> <tr> <td>B</td> <td>Unisys BSL standard, which is the default</td> </tr> <tr> <td>U</td> <td>USASI standard</td> </tr> <tr> <td>I</td> <td>A label standard for your installation (see the INSTALLATION LABEL1 record in <i>V Series Systems Operations Guide Volume 1: Installation</i>)</td> </tr> </table>	B	Unisys BSL standard, which is the default	U	USASI standard	I	A label standard for your installation (see the INSTALLATION LABEL1 record in <i>V Series Systems Operations Guide Volume 1: Installation</i>)
B	Unisys BSL standard, which is the default						
U	USASI standard						
I	A label standard for your installation (see the INSTALLATION LABEL1 record in <i>V Series Systems Operations Guide Volume 1: Installation</i>)						

Restriction

The tape drive must be in a saved state for this command to be used. Tape drives are automatically saved during halt/load if the TPMK option is set.

Details

The TM command can be used to add tape marks or labels to a tape that lacks such delimiters because of a system failure during file creation.

After a halt/load, if the TPMK option is set, all tape drives with reels that need to be rewound and have a write ring will have a saved status. You can use the TM command to write tape marks and labels, allowing these saved tape files to be properly closed.

Examples

TM 3/2

This writes a tape mark at the current point on the tape on channel number 3, unit number 2.

TM 10/3 REEL FILEA 003

This writes a tape mark, an EOF mark for FILEA, and an EOR label for reel number 003 on channel number 10, unit number 3. The label will be the default type.

See also:

- The AC (Purge and Place Tape Number in Label) command on Page 2-2.
- The CN (Display Tape Number) command on Page 2-78.
- The LABEL1 (Change Tape Label Format) command on Page 2-212.
- The SN (Purge and Place Tape Number in Label) command on Page 2-350.
- The SV (Save a Peripheral Unit) command on Page 2-368.
- The INSTALLATION LABEL1 and USE TPMK record in the *V Series Systems Operations Guide Volume 3: System Utilities*.

TO (Display System Options Status)

Keyboard Input Message

Access Level: 0

The TO command returns the status of the system options.

System options can be set either through the USE records in the system configuration file or through the SO (Set System Options) command. The RO (Reset System Options) command can be used to reset system options.



Explanation

- = This displays the settings of system options that can be set and reset.
- === This displays the settings of all system options.
- <option name> This displays the setting of a particular option.

Examples

TO =

This displays the setting of all settable or resettable system options such as AFTR=1 APBD=1 COPY=1 RJE=0 SYUP=0 TRAK=1.

TO OPEN

This displays the setting of the OPEN option. For example: OPEN=0.

TO MICR, ZIPM, SNAP

This displays the settings of the MICR, ZIPM, and SNAP options. For example: MICR=0 ZIPM=0 SNAP=1.

See also:

- The RO (Reset System Options) command on Page 2-327.
- The SO (Set System Options) command on Page 2-352.
- The USE records in the *V Series Systems Operations Guide Volume 1: Installation*.

TR (Time Reset)

Keyboard Input Message

Access Level: 8

The TR command changes the time of day used by the MCP.

- TR - <4-digit integer> _____|

Explanation

<4-digit integer> This is the new time given in 24-hour format (military time), without punctuation.

Restriction

No jobs can be in the mix when this command is entered. If there are jobs in the mix, the command is ignored.

Examples

TR 0800

This changes the time to 8:00 a.m.

TR 1535

This changes the time to 3:35 p.m.

See also:

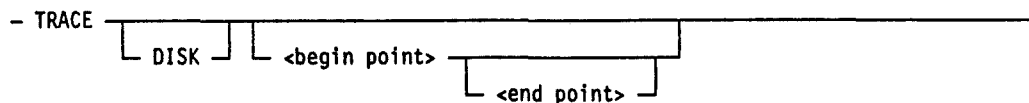
- The DR (Date Reset) command on Page 2-147.
- The WD (Display Date) command on Page 2-408.
- The WT (Display Time) command on Page 2-424.
- The USE DATE and TIME records in the *V Series Systems Operations Guide Volume 1: Installation*.

TRACE (Initiate Trace)

MCP Control Instruction

Access Level: 3

The TRACE command indicates that the program is to be traced when it enters the mix.



Explanation

DISK

This specifies that the trace is to go to backup disk.

DISK has the same effect as using the OU (Direct File to Device) command to send the file to disk or as setting the GTDK system option. The GTDK option can be set through the USE GTDK record in the system configuration file or with an SO (Set System Options) command.

If this option is omitted, the trace goes to the system default medium, determined by the setting of the GTDK system option.

<begin point>
<end point>

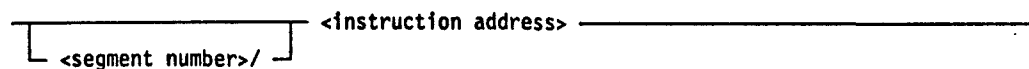
These allow the operator to specify beginning and ending limits for the tracing.

If the <begin point> is specified, the trace begins printing when the program reaches the designated point.

If the <end point> is also specified, the trace terminates at that point and does not resume until the <begin point> is again reached. The program is still being traced during this period, but the output is suppressed.

If you omit <begin point> and <end point>, tracing begins with the first instruction and ends with the last instruction.

The format for the <begin point> and <end point> specifications is shown in the following figure.



TRACE

<segment number>
<instruction address> This must be the exact segment number and address of an instruction. If either is 0, the corresponding limit is not enabled. A <segment number> of 0 means any segment, not just segment 0.

Details

The trace is listed on the first available line printer; if none is available, any printer in use by the designated program is used. If printers are not available, the program is suspended until a printer becomes available or until the operator directs the trace to backup disk through an OU (Direct File to Device) command. If the GTDK option is set, it goes to a disk backup file.

You can use the GT (Initiate Trace) command to trace a program that is executing.

The NT (Terminate Trace) command stops the tracing of a program.

Traces can also be started programmatically. The TRACE program branch communicate (BCT) requests the MCP to initiate or terminate a program trace. See the *V Series Program Interfaces Programming Reference Manual* and the various programming language reference manuals for more information.

The interactive DEBUG module has a number of sophisticated debugging features that can be used to trace user programs or the operating system itself. DEBUG (Attach Program to a DEBUG Session) and ID (Invoke DEBUG Session) can be used to start DEBUG sessions. GT and TRACE can be used only for user programs.

Example

```
EXECUTE PROG; TRACE
```

This starts the program PROG and begins tracing the program starting with the first instruction.

See also:

- The DEBUG (Attach Program to a DEBUG Session) command on Page 2-121.
- The GT (Initiate Trace) command on Page 2-183.
- The ID (Invoke DEBUG Session) command on Page 2-191.
- The NT (Terminate Trace) command on Page 2-243.
- The OU (Direct File to Device) command on Page 2-257.
- The SO (Set System Options) command on Page 2-352.
- The USE GTDK record in the *V Series Systems Operations Guide Volume 1: Installation*.
- The *V Series Program Interfaces Programming Reference Manual* and the various programming language manuals for information about TRACE.

UL (Assign Unlabeled File)

Keyboard Input Message

Access Level: 7

The UL command tells the MCP where to find an unlabeled file that is needed by a program.

```
- <mix number> - UL - <cc/u> _____
                        |_____|
                        <number of tape blocks to skip>
```

Explanation

<mix number>	This is the mix number of the program waiting for a file.
<cc/u>	This indicates the channel number and unit number of the device where the file is located.
<number of tape blocks to skip>	This applies to magnetic tapes. When the program opens the file, the tape is spaced forward the specified number of blocks, or until a tape mark is encountered, whichever comes first. Then the first data block is read. This allows magnetic tapes prepared on non-Unisys systems to be read.
	The number of blocks to skip must be less than 100.

Details

This command can be used when a program cannot find a file that it needs, causing a NO FILE message to appear on the ODT. You can direct the program to an unlabeled file by using the UL command and specifying a peripheral unit. The program expects the file on that peripheral unit to be the file it needs. It expects the file to be unlabeled. If the file is a tape with a foreign label not recognized by the system, the label may be bypassed with the skip count.

This command can be used to read magnetic tapes that lack standard tape labels. This includes tapes prepared on non-Unisys systems.

The UL command can be used with the following hardware types:

- Magnetic tape files
- Card files
- Reader/sorter files

For card files, the recording mode is assumed to be EBCDIC.

Card readers and reader/sorters can be either ready or not ready when the command is given.

Examples

23 UL 1/2

This tells the program with mix number 023 that the unlabeled file it needs is on channel number 1, unit number 2.

5 UL 0/5 2

This tells the program with mix number 005 that the unlabeled tape it needs is on channel number 0, unit number 5, and that the tape must be spaced forward two blocks (or to the first tape mark) before it is read.

See also:

- The FR (Designate Final Reel) command on Page 2-179.
- The IL (Assign Labeled File) command on Page 2-192.
- The OF (Optional File Indication) command on Page 2-245.

UNIT

PCH	This denotes an 80-column card punch.
RETRY <0-99>	This indicates the number of times the system should attempt to complete an I/O operation. The default setting is 10 retries. A value of 0 prevents any error recovery attempts. The number must be in the range 0 through 99.
SAVED	This saves the unit when it is added. The unit will remain saved after a cold start. The unit remains saved until made available with an RY (Ready Peripheral Device) command.
RESERVED	This reserves the unit when it is added. The unit will remain reserved after a cold start. This option can be used to declare a non-functioning unit to the system. You can use the UR (Inhibit or Uninhibit Unit) command to make the reserved unit available.

Example

UNIT 3/0 = PCH RETRY 15 SAVED RESERVED

This adds to the system an 80-column card punch, allows it 15 retry attempts, and gives the punch a saved and reserved status when it is added and after a cold start.

Adding a Data Communications Processor

To add a Data Communications Processor (DCP) to the system environment, use the following syntax.

- UNIT - <cc>/0 = DCP DCDLP
B874
B974
ICPV <DCP number 1-9>

Explanation

<cc>/0	This represents the channel number and a unit number for the device. Note that the unit number of the device must be 0.
=	This is optional and does not influence the command.
DCP	This denotes a Data Communications Processor.

DCDLP
B874
B974
ICPV

These are used for documentation purposes only, since the exact model of DCP is determined automatically.

<DCP number 1-9>

This indicates the logical DCP that is being declared. The number can range from 0 through 9, and this number is used to match each DCP specified in the NDL to a physical channel.

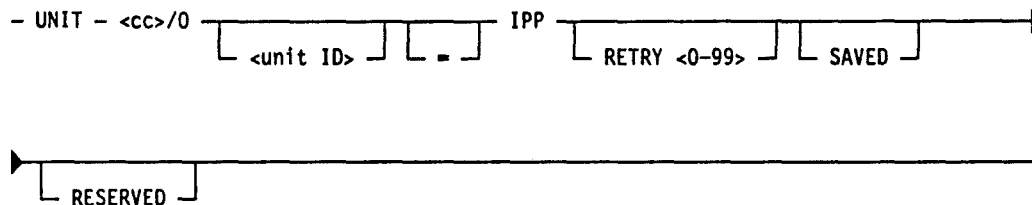
Example

UNIT 02/0 DCP B974 3

This adds a B974 DCP with a DCP number of 3 to the system on channel 2.

Adding an Image Page Printer (IPP)

To add an IPP to the system environment, use the following syntax.



Explanation

<cc>/0

This represents the channel number and a unit number for the device. Note that the unit number of the device must be 0.

<unit ID>

This denotes the identifier of the unit. The IPP can have up to a 6-character name. It is recommended that the unit have the same name as its default printer control file (PCF).

=

This is optional and does not influence the command.

IPP

This denotes an Image Page Printer.

RETRY <0-99>

This indicates the number of times the system should attempt to complete an I/O operation. The default setting is 10 retries. A value of 0 prevents any error recovery attempts. The number must be in the range from 0 through 99.

SAVED

This saves the unit when it is added. The unit will

UNIT

remain saved after a cold start. The unit remains saved until made available with an RY (Ready Peripheral Device) command.

RESERVED

This reserves the unit when it is added. The unit will remain reserved after a cold start. This can be used to declare a non-functioning unit to the system. You can use the UR (Inhibit or Uninhibit Unit) command to make the reserved unit available.

Example

```
UNIT 6/0 DOCPCF = IPP SAVED
```

This adds to the system an IPP called DOCPCF on channel 6. When the IPP is added and after a cold start, the device will be saved.

Adding an Inter-System Connect (ISC)

To add an ISC to the system environment, use the following syntax.

```
- UNIT - <cc>/0 [ = ] <ISC name> - ISC [ SAVED ] [ RESERVED ]
```

Explanation

<cc>/0

This represents the channel number and a unit number for the device. Note that the unit number of the device must be 0.

=

This is optional and does not influence the command.

ISC

This denotes an Inter-System Connect (ISC).

<ISC name>

This is a 1- to 6-character name. In BNA, it will be the first six characters of the IO_STATION_GROUP name and, therefore, the external name of the ISC file through which the station group is communicating.

SAVED

This saves the unit when it is added. The unit will remain saved after a cold start. The unit remains saved until made available with an RY (Ready Peripheral Device) command.

RESERVED

This reserves the unit when it is added. The unit will remain reserved after a cold start. This option can be used to declare a non-functioning unit to the system. You can use the UR (Inhibit or Uninhibit Unit) command to make the reserved unit available.

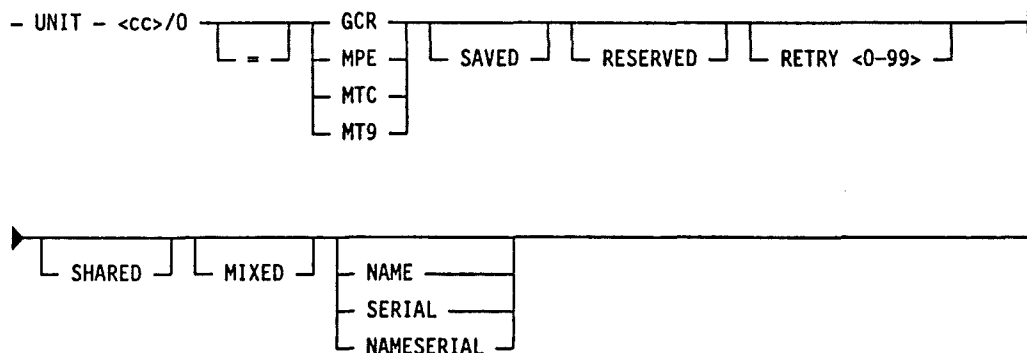
Example

UNIT 1/0 ISC GEORGE

This adds an ISC called GEORGE to the system on channel 1.

Adding MPE, GCR, MTC and MT9 Tape Drives

To add GCR, MPE, MTC, and MT9 type tape drives to the system, use the following syntax.

**Explanation**

- <cc>/0 This represents the channel number and a unit number for the device. Note that the unit number of the device must be 0.
- = This is optional and does not influence the command.
- GCR This denotes a tape drive that supports Group Coded Recording (GCR 6250 bpi) type tape.

Note: *Drives that support more than one recording density must be declared as GCR. To select a mode other than GCR, use syntax within a user program or override the drive's default mode using system commands. For example, to write a PE mode tape on a GCR drive that can operate in PE mode, use the tape type override feature of the AC (Purge and Place Tape Number in Label), PG (Purge Magnetic Tape), RP (Ready and Purge Magnetic Tape) or SN (Purge and Place Tape Number in Label) command.*

- MPE This denotes a tape drive that reads and writes Magnetic Phase-Encoded (MPE 1600 bpi) type tape.
- MTC This denotes a tape drive that reads and writes MTC (38000 bpi) type tape cartridges.

UNIT

MT9	This denotes a tape drive that reads and writes MT9 (800 bpi) type tape.
SAVED	This saves the unit when it is added. The unit will remain saved after a cold start. The unit remains saved until made available with a RY (Ready Peripheral Device) command.
RESERVED	This reserves the unit when it is added. The unit will remain reserved after a cold start. This can be used to declare a non-functioning unit to the system. You can use the UR (Inhibit or Uninhibit Unit) command to make the reserved unit available.
RETRY <0-99>	This indicates the number of times the system should attempt to complete an I/O operation. The default setting is 10 retries. A value of 0 prevents any error recovery attempts. The number must be in the range 0 through 99.
SHARED	<p>This saves the drive each time a program closes a tape file with a LOCK or RELEASE.</p> <p>This option automatically saves a drive after a halt/load.</p> <p>SHARED is used when a drive is physically connected to more than one system in a multi-system shared configuration. The drive should never be readied by more than one system at any given time as access errors will occur.</p>
MIXED	This indicates that the device supports the MT9 and MPE tape densities. MIXED does not apply to GCR or MTC types.
NAME SERIAL NAMESERIAL	These options apply to MTC tape drives only. NAME indicates that you want the display panel of the unit to display the tape name. SERIAL indicates that you want the display panel of the unit to display the serial number of the tape. NAMESERIAL indicates that you want the display panel of the unit to alternate the display between the tape name and the serial number.

Example

UNIT 04/0 MPE RESERVED RETRY 20

This adds to the system an MPE tape drive on channel 4. When the device is added and after a cold start, the device will be reserved. Twenty retry attempts to complete an I/O are allowed.

Adding a Non-Status Device (NST)

To add an NST to the system environment, use the following syntax. NSTs should be used for diagnostic purposes only.

- UNIT - <cc/u> - NST _____|

Explanation

<cc/u> This indicates the channel number and unit number of the device.

NST This denotes a Non-Status device.

Example

UNIT 6/3 NST

This adds an NST to the system on channel number 6, unit number 3.

Adding an ODT or OCS

To add an Operator Display Terminal (ODT) or Operator Control Station (OCS) to the system environment, use the following syntax.

- UNIT [<cc>/0] [<cc>/4] <unit ID> [_] [ODT] [OCS] LEVEL <number> [HLSPO] _____|

_____|

[<UNILINE firmware file name>] [AD <AD specs>]

Explanation

<cc>/0
<cc>/4 This indicates the channel number and unit number of the terminal. The unit number must be 0 for an OCS or 4 for an ODT.

<unit ID> This is a 1- to 6-character name. The first character must be alphabetic.

LEVEL <number> This limits the level of MCP commands that can be input from the terminal. See "Access Level" in Section 1 for more information.

HLSPO This indicates that the terminal is the HLSPO and will display halt/load and system failure messages.

<UNILINE firmware file name>	indicates the name of the file that contains the UNILINE firmware if the terminal is connected to a UNILINE DLP.
AD <AD specs>	See the AD (Specify ODT/OCS Screen Parameters) command for information about the AD specifications that control the automatic display of system status information on an ODT.

Example

```
UNIT 4/4 A107 = ODT LEVEL 7
```

This adds an ODT to the system. The ODT must have unit number 4 and will be on channel 4, have a unit ID of A107, and have an access level of 7. The default AD specifications will be used.

Adding a Buffered Printer or Train Printer

To add a buffered or train printer to the system environment, use the following syntax.

```
- UNIT - <cc>/0<unit ID> [ - ] [ PRN ] [ TPR ] [ SAVED ] [ RESERVED ]
```

Explanation

<cc>/0	This represents the channel number and a unit number for the device. Note that the unit number of the device must be 0.
<unit ID>	This denotes the train image file to be loaded. For buffered printers, use the name PRN256 for the 2000 LPM printer, 1050-1250 printer, and 600 LPM printer. For train printers, use the name of the default train printer image file (for example, (FUL96). The files can be copied and renamed to provide for unique names for the printers. See the LOADER utility in the <i>V Series Systems Operations Guide Volume 3: System Utilities</i> for more information.
=	This is optional and does not influence the command.
PRN	This denotes a buffered printer
TRP	This denotes a train printer.

UNIT

- SAVED** This saves the unit when it is added. The unit will remain saved after a cold start. The unit remains saved until made available with an RY (Ready Peripheral Device) command.
- RESERVED** This reserves the unit when it is added. The unit will remain reserved after a cold start. This option can be used to declare a non-functioning unit to the system. You can use the UR (Inhibit or Disinhibit Unit) command to make the reserved unit available.

Example

```
UNIT 5/0 PRN256 PRN SAVED
```

This adds a buffered printer to the system environment. The 2000 LPM printer will be on channel 5, has an ID of PRN256, and will have a saved status when it is added and after a cold start.

Adding Data Communications Devices (RJE, TC5, VDD, TWX)

To add RJE, TC5, VDD, or TWX data communications devices to the system environment, use the following syntax.

```
- UNIT - <cc>/0 - <unit ID> [ - ] [ RJE ] [ TC5 ] [ VDD ] [ TWX ] [ DIAL ] [ ASYN ]
```

Explanation

<cc>/0

This represents the channel number and a unit number for the device. Note that the unit number of the device must be 0.

<unit ID>

This is a 1- to 6-character name for RJE, TC5, VDD, and TWX devices. The first character must be alphabetic, and the remaining characters alphanumeric. Unlike most other devices, data communications lines have no label or other physical identifier that associates a line with the device. To allow programmatic assignment by name, each line is assigned a unit ID. The unit ID doubles as its adapter ID.

=

This is optional and does not influence the command.

RJE

This denotes a Remote Job Entry device.

TC5	This denotes a TC 500 device.
VDD	This denotes a B 2X/B3X or ET Series workstation or a B 9352 or TD 800 Series terminal.
TWX	This denotes a teletypewriter through a UNILINE DLP.
DIAL	This specifies that the device is linked to a switched line.
ASYN	This is used only with RJE devices and declares the line as asynchronous.

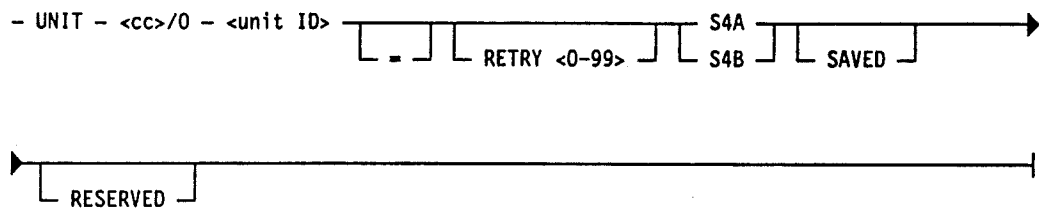
Example

```
UNIT 2/0 JGM33 RJE ASYN
```

This adds to the system an RJE device with a unit ID of JGM33 on channel number 2. The device will use asynchronous communication.

Adding Check Reader Sorters

To add S4A or S4B check reader sorter devices to the system environment, use the following syntax.

**Explanation**

<cc>/0	This represents the channel number and a unit number for the device. Note that the unit number of the device must be 0.
<unit ID>	This denotes the identifier of the unit. The unit ID for S4A and S4B devices is a 1- to 6-character name. The first character must be alphabetic, the remaining characters alphanumeric.
=	This is optional and does not influence the command.
S4A	This denotes a 9137 reader sorter.

UNIT

S4B	This denotes a 9138 reader sorter.
RETRY <0-99>	This indicates the number of times the system attempts to complete an I/O operation. The default setting is 10 retries. A value of 0 prevents any error recovery attempts. The number must be in the range 0 through 99.
SAVED	This saves the unit when it is added. The unit will remain saved after a cold start. The unit remains saved until made available with an RY (Ready Peripheral Device) command.
RESERVED	This reserves the unit when it is added. The unit will remain reserved after a cold start. This option can be used to declare a non-functioning unit to the system. You can use the UR (Inhibit or Disinhibit Unit) command to make the reserved unit available.

Example

UNIT 4/0 S4A002 S4A RETRY 12 SAVED

This adds to the system a S4A check reader sorter with a unit ID of S4A002 on channel number 4. This unit is allowed 12 retry attempts and will have a saved status when it is added and after cold start.

See also:

- The DISK (Add a Disk) command on Page 2-128.
- The DL (Delete Peripheral Devices) command on Page 2-135.
- The PACK (Add a Pack) command on Page 2-262.
- The RY (Ready Peripheral Device) command on Page 2-338.
- The UR (Inhibit or Uninhibit Unit) commands. command on Page 2-399.
- The DISK, PACK, and UNIT records in the system configuration file in the *V Series Systems Operations Guide Volume 1: Installation*.

UNLOAD (Move Library Files between Media, Remove Old Files)

MCP Control Instruction

Access Level: 5

The UNLOAD command moves files. When the files have been moved to the new location, they are deleted from the old location.

Unisys recommends that you use the MOVE command and SYSTEM/COPY instead of the UNLOAD command. The UNLOAD command will become unavailable at a future release.

The UNLOAD command starts a program called LOADMP, if disk files are to be moved to tape, or a program called PACKUP, if files are to be moved to or from diskpack. The user must supply LOADMP and PACKUP because Unisys does not supply them. If the appropriate program is not present on disk, the UNLOAD command is ignored.

See Appendix B for UNLOAD syntax if you have user-coded LOADMP and/or PACKUP utilities.

See also:

- The COPY (Invoke File Transfer Utility) command on Page 2-86.
- Appendix B.

UP (Unlock Program)

Keyboard Input Message

Access Level: 1

The UP command unlocks a locked program, thus allowing certain ODT commands to be directed to that program.

- <mix number> - UP _____|

Explanation

<mix number> This is the mix number of the locked program to be unlocked.

Details

This command reverses the effect of the LOCK or LP (Lock Program) commands. After the program is unlocked, it can be discontinued or otherwise manipulated.

Example

23 UP

This unlocks the locked program associated with mix number 023.

See also:

- The LOCK (Lock Program) command on Page 2-227.
- The LP (Lock Program) command on Page 2-228.

UR (Inhibit or Uninhibit Unit)

Keyboard Input Message

Access Level: 8

The UR command can either temporarily remove a peripheral device from the system or restore it to the system. UR can also interrogate whether or not a peripheral device is inhibited.



Explanation

UR (without + or -)	This displays whether or not the device is currently inhibited.
UR +	This inhibits the unit.
UR -	This uninhibits the unit.
<cc/u>	This is the channel number and unit number of the device.

Details

Inhibited status, caused by a UR + command or the RESERVED option on DISK, PACK, and UNIT records or commands, can be removed only by a UR - command.

When a device is inhibited, it is ignored by the MCP and cannot be assigned to a program. However, it can still be reached by direct I/O.

Inhibited status is retained across halt/loads.

You should use the DL (Delete Peripheral Devices) command to permanently remove a device from the system.

You can interrogate a device that has been permanently removed from the system by using the UNIT (Add Peripheral Devices) command.

Examples

UR + 4/3

This inhibits the device on channel number 4, unit number 3.

UR - 3/0

This uninhibits (makes available) the device on channel number 3, unit number 0.

UR + 2/0, - 23/1

This inhibits the device on channel number 2, unit number 0 and uninhibits the device on channel number 23, unit number 1.

See also:

- The DL (Delete Peripheral Devices) command on Page 2-135.
- The UNIT (Add Peripheral Devices) command on Page 2-385.

USER (Assign Access Code)

MCP Control Instruction

Access Level: 0

The USER command assigns an access code to a single job or input command. An access code is a usercode/password/charge number combination.

You enter the USER command before a command that starts a program, such as COMPILE (Compile Program) or EXECUTE (Execute Library Program). The program then executes under the attributes of that access code. You can also precede file maintenance commands or any other command with the USER command.

```
- USER [ <usercode> ] [ / ] [ <password> ] [ / ] [ <charge number> ]
```

Explanation

- | | |
|------------------------------------|--|
| <code><usercode></code> | This refers to a code that identifies a user to the system. |
| <code><password></code> | This refers a group of characters by which the system determines the legality of a user. |
| <code><charge number></code> | This refers to a number used to assign charges for system use. |

Note: The `<usercode>`, `<password>`, and `<charge number>` are all optional. Slashes are required only if leading parameters are omitted (for example, `USER //63023`).

Details

The device from which the USER command is entered does not have to be logged in.

The security attributes of the access code become attached to the next program or function started from this device. These attributes override any that might have been active from previous BEGINUSER (Beginning of Job Stream) or LI (Log In) commands. When the input command or job affected by the USER command has completed processing, the security attributes revert to what they were before the USER command was entered.

If the CHRГ option is set, a charge number must be given for each program initiation (certain MCP utilities excepted).

USER

The charge number can be overridden by the CHARGE command. CHARGE associates a program with a charge number but does not affect the security attributes of the user or the program. (However, the charge number along with the user code and password must form a valid access code.)

Note that the BEGINUSER (Beginning of Job Stream) and LI (Log In) commands affect security attributes for all programs executed from the logged-in device, whereas the USER command affects security attributes for only one program.

Example

```
USER USR2/PASS2/999999; EXECUTE RUST
```

This starts the program RUST with the security attributes of user code USR2, password PASS2, and charge number 999999.

See also:

- The BEGINUSER (Beginning of Job Stream) command on Page 2-45.
- The CHARGE (Insert Charge Number) command on Page 2-74.
- The ENDUSER (Mark End of Job Stream) command on Page 2-158.
- The LI (Log In) command on Page 2-221.
- The LO (Log Out) command on Page 2-225.
- The SECURITY and USE CHRG records in the *V Series Systems Operations Guide Volume 1: Installation*.
- Details on system security are given in the *V Series System Security Installation and Operations Guide*.

VALUE (Insert Value into a Program)

MCP Control Instruction

Access Level: 1

The VALUE command inserts six digits into the code file of a program prior to its execution.

- VALUE - <starting address> _ _ <integer>

Explanation

<starting address>

This is the program base-relative address at which to start inserting data. The address can range from 00000 through 00058.

Addresses 0 through 7 (memory locations 000000-000007) can represent programmatic switches as shown in the following table.

Switch	Memory Location
8	000000
1	000001
2	000002
3	000003
4	000004
5	000005
6	000006
7	000007

=

This is optional and does not influence the command.

<integer>

This can contain the digits 0 through 9 and the undigits A through F. Six digits are always inserted into the program. If <integer> is less than six digits, leading zeros are assumed.

Details

You can use the IN (Insert Data into Program) and INSERT (Insert Data into Program) commands to enter information into a program. INSERT is used at program initiation time, while IN is used on a program that is already executing.

VALUE

The SW (Set Programmatic Switches) command sets programmatic switches, which is equivalent to inserting values at addresses 0 through 7. SW is used on a program that is already executing.

Note: The VALUE command cannot be associated with a COMPILE (Compile Program) or EXECUTE (Execute Library Program) command if you use the <parameter list> option. The MCP continues to allow slash parameters. However, when they are used, a warning will be displayed on the ODT and written to the ODT log stating that slash parameters will be deimplemented at a future MCP/VS release.

The first parameter of the <parameter list> causes eight digits to be inserted, while the VALUE command inserts only six digits. This may necessitate the use of trailing zeros when VALUE commands that begin at address 0 or 1 are replaced by <parameter list>. For example: EX ABC VALUE 0 = 1234 is equivalent to EX ABC (123400); while EX ABC VA 2 = 5678 is equivalent to EX ABC (5678).

Example

```
COMPILE PROGA COBOL9; VALUE 0 = 100000
```

This compiles PROGA with the COBOL9 compiler, inserts a value of 1 at location 0 of the COBOL9 compiler code, and inserts zeros at locations 2, 3, 4, 5, and 6.

See also:

- The COMPILE (Compile Program) command on Page 2-80.
- The EXECUTE (Execute Library Program) command on Page 2-162.
- The IN (Insert Data into Program) command on Page 2-195.
- The INSERT (Insert Data into Program) command on Page 2-197.
- The SW (Set Programmatic Switches) command on Page 2-369.

WB (Display Bound Programs)

Keyboard Input Message

Access Level: 0

The WB command returns a list of the utility programs whose code is bound to (intrinsic to) the MCP.

- WB -

Sample response

The following two lines show a sample response to the WB command:

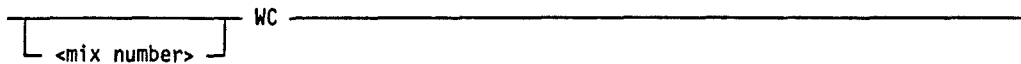
```
WB = DMPDUT RLGOUT SLGOUT MLGOUT LDCNTL DSKOUT DMSDLR PRN256  
WB = COPY SQUASH DISPKV PBDPRN FUL96 FUL72 SORT: MERG: NEWLAB
```

WC (Display Available Memory)

Keyboard Input Message

Access Level: 0

The WC command returns the amount of contiguous memory, in kilodigits (kd), that is currently available.



Explanation

<mix number> If a mix number is supplied, the response is the amount of memory being used by that program, exclusive of disk file headers and device-alternate blocks.

Sample Response

```
MEMORY: 28463 77 50 30 30 30 15 9 9 8
        5 5 5 5 4 2 2 2 1 1
        1
TOTAL AVAILABLE: 28754 KD
```

In this response, there are 21 areas of memory available. One is 28463 kilodigits long and the others are 77, 50, 30, 30, 30, 15, 9, 9, 8, 5, 5, 5, 5, 4, 2, 2, 2, 1, 1, kilodigits long. Each area is a piece of contiguous memory, but the areas are not contiguous to each other.

Details

Sometimes the MCP cannot arrange memory so that all available memory is contiguous. This can be caused by memory that has been removed from the system by an XM (Remove Memory from System) command, among other causes.

If the response to WC indicates two or more areas of available memory, these are not necessarily the maximum amounts of memory that could be made available in each area. The MCP could increase the available space in one area and decrease the space in the other by moving programs from one area to the other. Thus, the decision to attempt the execution of a job must not be based solely on the response to a WC command.

Examples

WC

This causes a display of the amount of memory currently available.

5 WC

This causes a display of the amount of memory being used by the job with mix number 005.

See also:

The XM (Remove Memory from System) command on Page 2-437.

WD (Display Date)

Keyboard Input Message

Access Level: 0

The WD command returns either the current date or the day of the week of a given date.

```

- WD _____
  |
  | + <integer> |
  | - <integer> |
  | <mm/dd/yy>  |
  | <yyddd>     |
  |_____
  
```

Explanation

WD	This displays the current date, day of the week, and Julian date.
+ <integer>	This causes the number of days specified by <integer> to be added to the current date. The day of the week and the Julian date are calculated.
- <integer>	This causes the <integer> number of days to be subtracted from the current date. The day of the week and the Julian date are calculated.
<mm/dd/yy>	This causes a date to be supplied in month/day/year format - for example, 01/23/87 for January 23, 1987. The day of the week and the Julian date are calculated. The date can range from 01/01/74 through 12/31/98.
WD <yyddd>	This causes a julian date to be supplied, with two digits for the year and three digits for the day. For example, 85185 is the 185th day of 1985. The date and day of the week are calculated. The date can range from 74001 through 98365.

Sample Response

The following figure shows a sample response to a WD command.

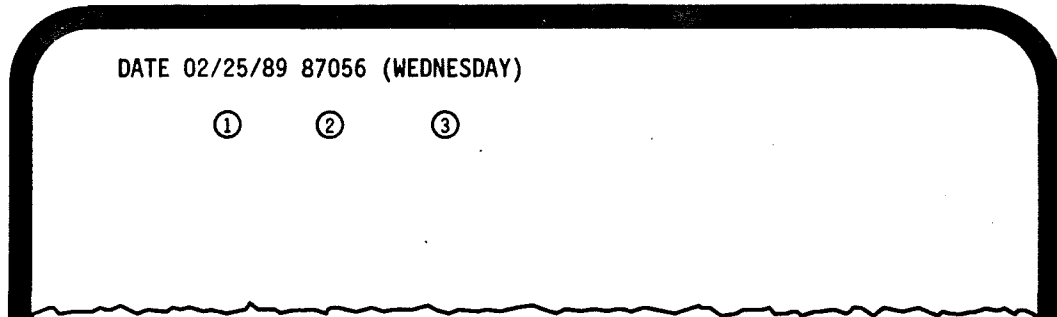


Figure 2-3 . System Response to the WD Command

Each of the elements marked by circled numbers in Figure 2-3 is further described by the following.

Number	Description
①	This is the current date.
②	This is the current date expressed as a Julian date.
③	This is the day of the week.

Details

The current date is based on the date entered by the operator at system initialization time or through the DR (Date Reset) command.

The space between WD and the minus sign is required, but not between WD and the plus sign (+).

Examples

WD + 344

This displays the date 344 days in the future.

WD - 2

This displays the date of the day before yesterday.

WD 75360

This displays the date and day of the week of the 360th day in 1975.

See also:

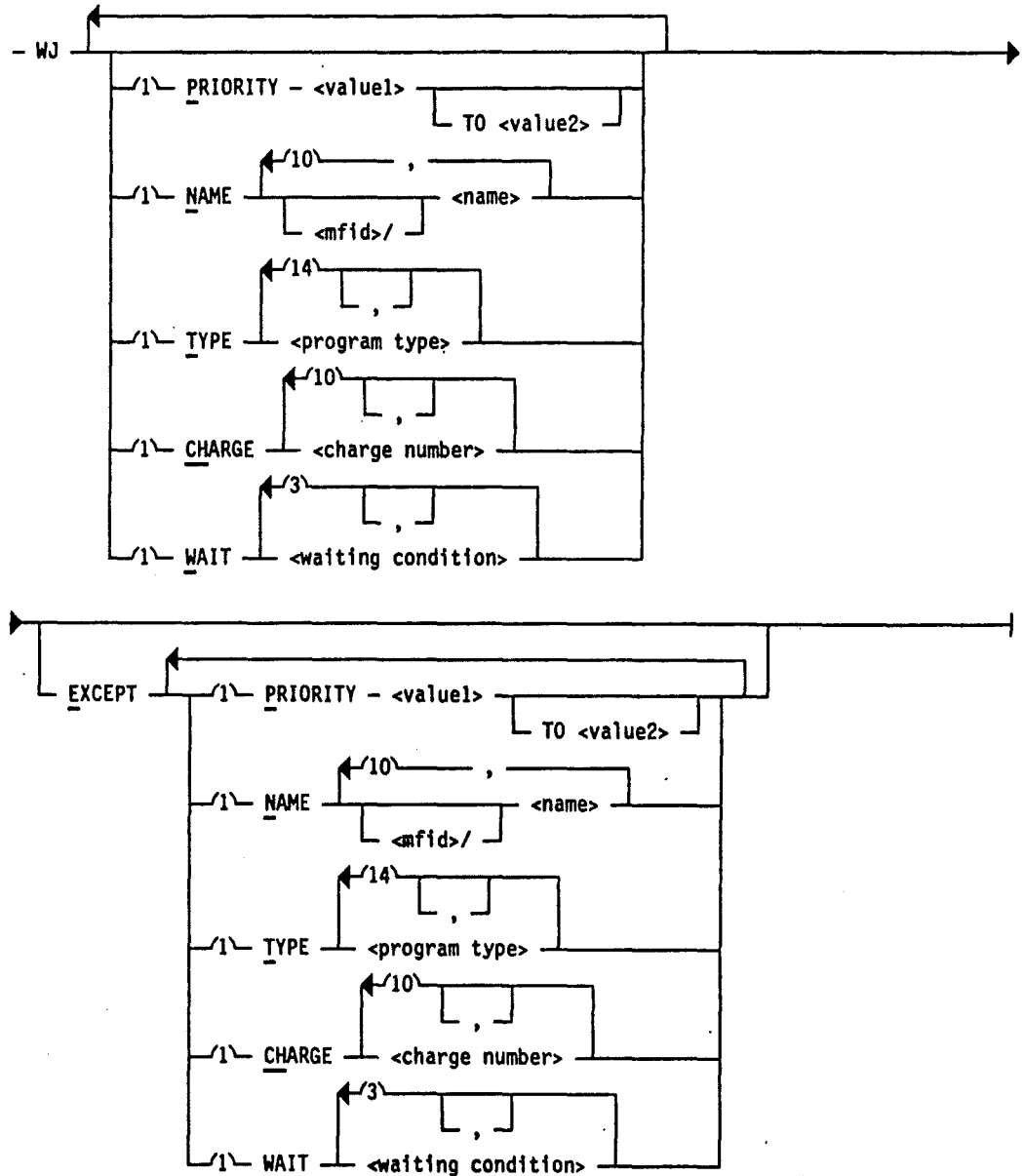
- The DR (Date Reset) command on Page 2-147.
- The TR (Time Reset) command on Page 2-380.
- The WT (Display Time) command on Page 2-424.
- The DATE and TIME records in the *V Series Systems Operations Guide Volume 1: Installation*.

WJ (Display Waiting Jobs)

Keyboard Input Message

Access Level: 0

WJ lists only those jobs that are currently waiting. You can list a subset of the waiting jobs by using syntax options of the command.



Explanation

PRIORITY <value1> TO <value2> Selects jobs based on processor priority. You have the option of selecting a single priority or a range of priorities. The values of <value1> and the optional <value2> must be integers from 1 through 9, and <value1> must be less than or equal to <value2>.

NAME <multiprogram name> / <program name> Selects jobs based on the program name and (optionally) on the multiprogram name. You can enter a maximum of 10 name selections, separated by commas.

You can also select jobs based on part of a name by using the equal sign as a masking character. Use an equal sign for any unknown characters in the name. For example, the following command displays all programs in the mix beginning with the three characters DMP, but does not display the program DMPDOUT.

WJ NAME DMP= EXCEPT NAME DMPDOUT

TYPE <program type> Selects jobs based on the type of program. You can enter a maximum of 14 type selections. The following list shows the available program types:

<u>BNAHNDL</u>	Program is a BNA handler.
<u>COMPILER</u>	Program generates object code.
<u>COMPISHAR</u>	Program that generates object code, executed as a timesharing task.
<u>COPY</u>	Program is SYSTEM/COPY intrinsic.
<u>DMPALL</u>	Program is DMPALL intrinsic.
<u>DMSPROG</u>	Program is a DMSII control program (for example, DBP).
<u>DSKOUT</u>	Program is DSKOUT intrinsic or diskpack SQUASH intrinsic.
<u>EXECUTE</u>	Program is executing.
<u>HANDLERS</u>	Program is any type of handler (BNA, WFL, time sharing).
<u>LOADMPP</u>	Program is LOADMPP intrinsic.
<u>MCS</u>	Program is Message Control System (MCS).
<u>PACKUP</u>	Program is PACKUP intrinsic.
<u>SHAREHNDL</u>	Program is a timesharing handler program (for example, CANDE).
<u>SORT</u>	Program is SORT: or SORT. intrinsic.

	<u>TIMESHARE</u>	Program is executed as time sharing task.
	<u>WFLHNDL</u>	Program is a Work Flow Language (WFL) handler.
	<u>UTILITY</u>	Program is any type of intrinsic (DMPALL, LOADMP, PACKUP, DSKOUT, SYSTEM/COPY).
CHARGE <charge number>		<p>Selects jobs based on the charge number the job is executed under. You can enter as many as 10 charge number selections and 10 charge number exceptions.</p> <p>You can also select jobs based on part of a charge number by using the equal sign as a masking character. Use an equal sign for any unknown characters in the charge number. For example, the following command displays jobs executed under a charge number beginning with the characters 66, but does not display jobs executed under charge numbers 660001 or 669999.</p> <p>WJ CH 66= EXCEPT CH 660001, 669999</p>
WAIT <waiting condition>		<p>Selects jobs that are waiting for a specific condition to be satisfied (for example, waiting hardware, waiting memory and so forth). You can enter as many as 3 waiting conditions and 3 waiting condition exceptions.</p> <p>The available types of waiting conditions are as follows:</p>
	<u>DEVICE</u>	Waiting for hardware device to become available
	<u>EVENT</u>	Waiting for a software event to be signaled (for example, STOQUE, CRCR, and so forth)
	<u>OPERATOR</u>	Waiting for operator to perform an action
	<u>SPACE</u>	Waiting for memory, disk space or diskpack space
EXCEPT		All text following the keyword EXCEPT is treated as exceptions to the selections entered previously. You must group all of the exceptions must be grouped together after the keyword EXCEPT; you cannot enter a selection and an exception followed by another selection.

Details

The WJ command corresponds to the WM option of the <table> element of the AD command. WJ and WM display the jobs waiting in the mix, including the processor and memory priorities and memory required for each job.

Some reasons a job can be waiting are as follows:

- There is not enough memory available.
- It is waiting for a peripheral device to become available.
- It is waiting to receive a message from another program.
- It is waiting for the operator to give it a message with the AX (ACCEPT) command.
- It has received a no-file or duplicate-file message.

Sample Display

The following figure shows a response to the WJ command.

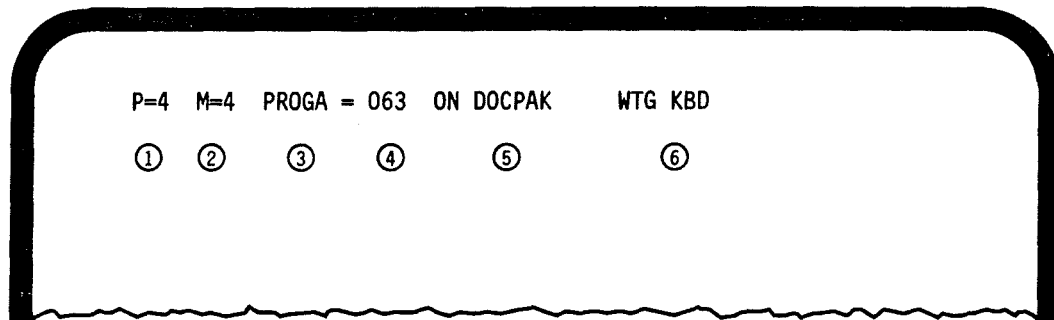


Figure 2-4. Sample Response to the WJ Command

Each of the circled numbers in Figure 2-4 corresponds to an entry in the following table.

Number	Description
①	This is the processor priority of the job.
②	This is the memory priority of the job.
③	This is the name of the job.
④	This is the mix number of the job.
⑤	This is the location of the program code file.
⑥	This is the current status of the job.

If no jobs are waiting, the system displays the message NULL MIX.

See also:

- The AD (Specify ODT/OCS Screen Parameters) command on Page 2-4.
- The AJ (Display Active Jobs) command on Page 2-14.
- The MX (Display Jobs in Mix) command on Page 2-236.
- The WS (Display Jobs in Schedule) command on Page 2-423.

WM (Display MCP Version)

Keyboard Input Message

Access Level: 0

The WM command displays the name, release level, patch level, and version date of the MCP currently running; a list of system options that are set; the system number; the hostname; and the amount (in kilodigits) of main memory and MCP QWIK memory.

- WM _____

Sample Response

The following figure shows a response to the WM command.

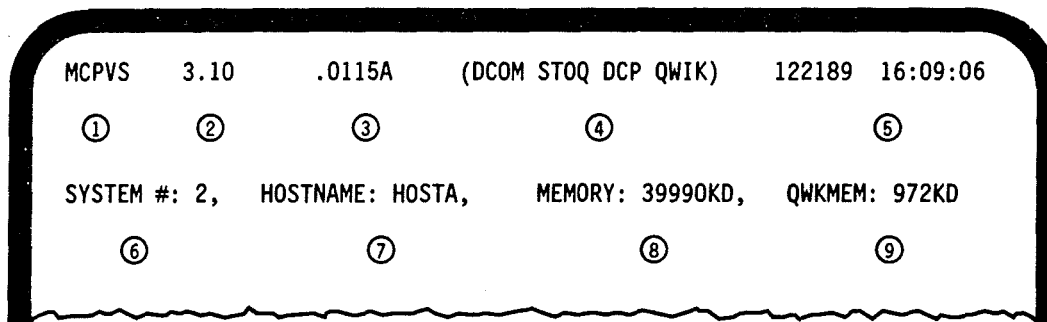


Figure 2-5. Sample Response to the WM Command

Each of the circled numbers in Figure 2-5 corresponds to an entry in the following table.

Number	Description
①	This is the name of the current MCP.
②	This is the MCP release level.
③	This is the MCP patch level.
④	These are the system options that are set. System options are set by USE records in the system configuration file or by the SO (Set System Options) command. They can be reset by the RO (Reset System Options) command.
⑤	This is the version date and time of the MCP. The date is in <i>mmdyy</i> format.

Number	Description
⑥	This is system number of the system on which the command was entered.
⑦	This is the hostname of the processor. The hostname is set by the HOSTNAME record in the system configuration file or by the HN (Identify or List Host Name) command. The hostname identifies this processor in communications, system logs and networking environments.
⑧	This is the amount of main memory available to the MCP.
⑨	This is the amount of memory assigned as MCP QWIK memory. MCP QWIK memory is memory available to store the most recently used MCP overlays, so that they will not have to be accessed from disk if they are needed again. This feature increases performance and is set by the LIMIT QWKMEM record in the system configuration file.

See also:

- The HN (Identify or Change Hostname) command on Page 2-189
- The RO (Reset System Options) command on Page 2-327.
- The SO (Set System Options) command on Page 2-352.
- The HOSTNAME, LIMIT QWKMEM, and USE records in the *V Series Systems Operations Guide Volume 1: Installation*.

WO (Display Active User QWIK Operation Counts)

Keyboard Input Message

Access Level: 0

The WO command displays the statistics maintained for user QWIK memory for the current five-minute period.

- WO _____|

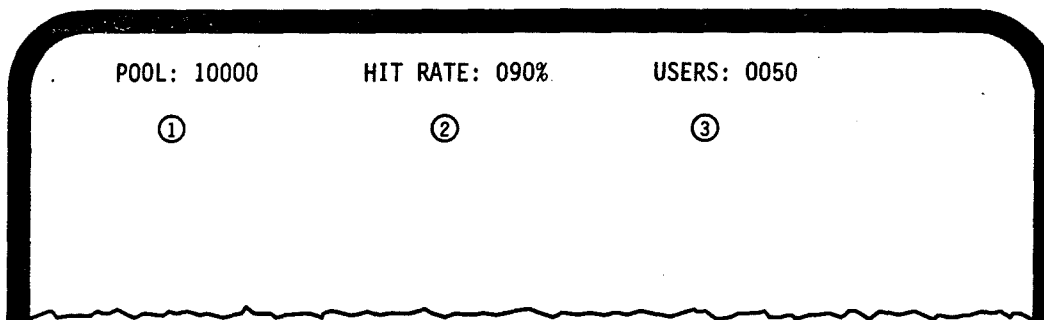
Explanation of User QWIK Memory

The user QWIK pool is memory dedicated to storing the most recently used user program overlays, so that they will not have to be accessed from disk if they are needed again. This feature usually increases performance. You can set this feature through the USE QWIK record in the system configuration file or by using the SO (Set System Options) command. You can be reset it by using the RO (Reset System Options) command.

There is a difference between user QWIK memory and MCP QWIK memory. MCP QWIK memory is memory dedicated to storage of the most recently used MCP overlays and is set by the LIMIT QWKMEM record in the system configuration file.

Sample Response

The following figure shows a response to the WO command.



Each of the circled numbers in Figure 2-6 corresponds to an entry in the following table.

Number	Description
①	This is the size of the QWIK pool set by the LIMIT QWIKPOOL record in the system configuration file or the QWIKPOOL (Set Program Overlay Buffer Pool) command.

Number	Description
②	The hit rate is the percentage of times that an overlay was called and found in the QWIK pool, and therefore did not have to be retrieved from disk.
③	The number of QWIK pool users is the number of tasks currently retrieving overlays from the QWIK pool.

Details

Statistics, tabulated by QWIK for each 5-minute interval, are placed in the Run log and can be interpreted by RLGOUT or the Time Analysis and Billing System II (TABSII). Among these statistics are the following:

- The number of times an overlay was found in the pool
- The number of times an overlay was called
- The percentage of overlay calls in which the overlay was found in the QWIKMEM pool

More detailed statistics on user QWIK memory are available through the Run log report program and the TABSII program product. See the *V Series System Software Logging Operations Reference Manual* and the *B 2000/B 3000/B 4000/V Series TABS II Installation and Operations Guide*.

See also:

- The QWIKPOOL (Set Program Overlay Buffer Pool) command on Page 2-310.
- The SO (Set System Options) command on Page 2-352.
- The RO (Reset System Options) command on Page 2-327.
- The TO (Type System Options) command on Page 2-379.
- The USE QWIK and LIMIT QWIKPOOL records in the system configuration file in the *V Series Systems Operations Guide Volume 1: Installation*.
- The *B 2000/B 3000/B 4000/V Series TABS II Installation and Operations Guide*.

Sample Responses

QUEUEA 5

There are five messages waiting in a storage queue named QUEUEA.

QUEUEB 0 (WTG)

There are no messages in the storage queue named QUEUEB, but a program has issued a request for a message from this queue. The program is waiting for the message.

See also:

- The RQ (Remove Storage Queue Entries) command on Page 2-331.
- The USE STOQ record in the *V Series Systems Operations Guide Volume 1: Installation*.
- Complete details on storage queues are given in the various language manuals and in the *V Series MCP/VS Programming Reference Manual*.

WRU (Display ODT Access Code)

Keyboard Input Message

Access Level: 0

The WRU command displays the usercode and charge number under which the ODT is logged in.

- WRU _____|

Sample Response

YOU ARE:

UC:9688, CG:4150, UN:JBlunt, ST:C132S3, LSN:221

WS (Display Jobs in Schedule)

Keyboard Input Message

Access Level: 0

The WS command displays a list of jobs that are scheduled to execute but are not yet executing.

- WS _____

Details

If no jobs are in the schedule, the message "NULL SCHED" appears on the ODT.

Jobs can be removed from the schedule by the RS (Remove Job from Schedule) command.

See also:

- The AFTER (Hold Program) command on Page 2-12.
- The RA (Remove After-Linkage) command on Page 2-312.
- The RS (Remove Job from Schedule) command on Page 2-333.
- The WJ (Display Waiting Jobs) command on Page 2-411.
- The WY (Display Job Status) command on Page 2-427.

WT (Display Time)

Keyboard Input Message

Access Level: 0

The WT command displays the time of day. The time is given in 24-hour format (military time). The time of day used by the system is set at system initialization. You can change the system time by using the TR (Time Reset) command.

- WT _____|

Sample Response

TIME 13:46:26

This indicates that the time is 1:46 p.m. and 26 seconds.

See also:

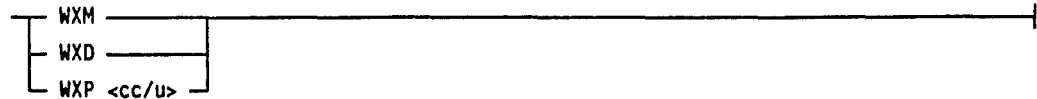
- The DR (Date Reset) command on Page 2-147.
- The TR (Time Reset) command on Page 2-380.
- The WD (Display Date) command on Page 2-408.
- The DATE and TIME records in the *V Series Systems Operations Guide Volume 1: Installation*.

WX (Display Removed Space)

Keyboard Input Message

Access Level: 8

The WX command lists all areas of memory, disk, or diskpack that have been removed from the system by the operator.



Explanation

WXM	This displays deleted areas of memory.
WXD	This displays deleted areas of disk.
WXP <cc/u>	This displays deleted areas of diskpack. The <cc/u> is the channel number and unit number of the diskpack drive.

Details

Areas of memory, disk, or diskpack can be removed from the system if they become faulty. You can do this by using the XM (Remove Memory from System), XD (Remove Disk Segments), or XP (Remove Pack Sectors) command.

Areas that have been removed with the XM, XD, or XP command can be restored to the system with the RXM, RXD, and RXP forms of the RX (Return Deleted Space) command.

Examples

WXM

This displays the areas of memory that have been deleted.

WXD

This displays the disk areas that have been deleted.

WXP 8/3

This displays the areas deleted from the diskpack mounted on the drive that is on channel number 8, unit number 3.

See also:

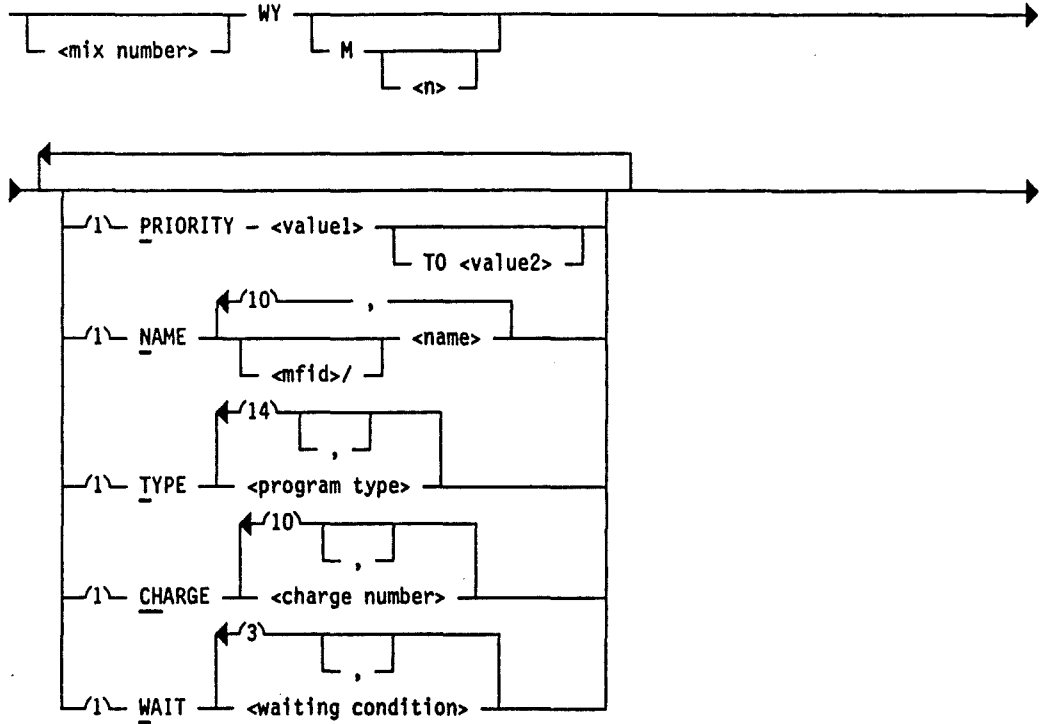
- The RX (Return Deleted Space) command on Page 2-336.
- The XD (Remove Disk Segments) command on Page 2-436.
- The XM (Remove Memory from System) command on Page 2-437.
- The XP (Remove Pack Sectors) command on Page 2-438.

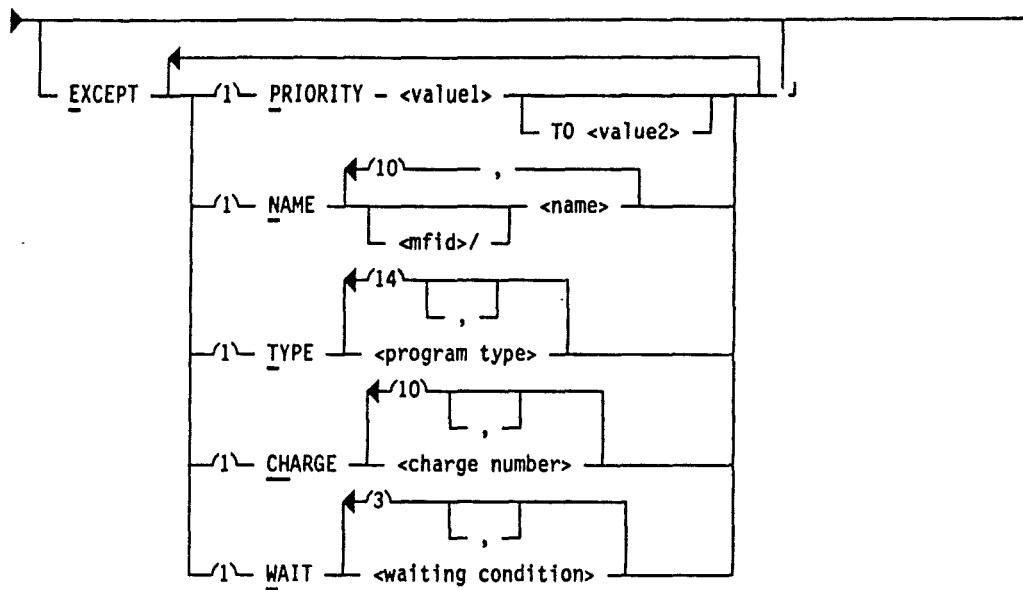
WY (Display Job Status)

Keyboard Input Message

Access Level: 0

The WY command displays the status of a job and the messages associated with that job. WY also displays the status of all jobs in the mix. You can display a subset of jobs in the mix by using syntax options of the command.





Explanation

<mix number>

This is the mix number of a job on the system. The operating system displays the status of the job associated with the mix number. The value of <mix number> cannot be 0 or 1.

Because this option selects one specific job, the job selection options (PRIORITY, NAME, and so forth) are ignored.

M

This causes the MCP to display both previous ACCEPT and DISPLAY messages. For example, <mix number> WY M displays the following:

- The status of the job
- Messages sent to the job through the AX (Response To An ACCEPT) command
- Messages sent from the job for display on the ODT (for example, by the DISPLAY verb in COBOL)

<n>

This is the number of messages to display and can be a number from 1 to 10. The default is 2.

PRIORITY <value1> TO <value2>

Selects jobs based on processor priority. You have the option of selecting a single priority or a range of priorities. The value of <value1> and the optional <value2> must be integers from 1 through 9, and the value of <value1> must be less than or equal to <value2>. The abbreviation P is accepted for PRIORITY.

NAME
<mfid> / <name>

Selects jobs based on the program name and (optionally) on the multiprogram name, or multifile id (mfid). You can enter a maximum of 10 name selections, separated by commas. The abbreviation N is accepted for NAME.

You can also select jobs based on part of a name by masking the name using the equal sign as a masking character. Use an equal sign for any unknown characters in the name. For example, the following command displays all programs in the mix beginning with the three characters DMP, but does not display the program DMPOUT.

WY NAME DMP= EXCEPT NAME DMPOUT

TYPE <program type>

Selects jobs based on the type of program. The abbreviation T is accepted for TYPE. You can enter a maximum of 14 type selections. The available program types are as follows:

<u>B</u> NAHNDL	Program is a BNA handler.
<u>C</u> OMPILER	Program generates object code.
<u>C</u> OMPSHAR	Program that generates object code, executed as a timesharing task.
<u>C</u> OPY	Program is SYSTEM/COPY intrinsic.
<u>D</u> MPELL	Program is DMPALL intrinsic.
<u>D</u> MSPROG	Program is a DMSII control program (for example, DBP).
<u>D</u> SKOUT	Program is DSKOUT intrinsic or diskpack SQUASH intrinsic.
<u>E</u> XECUTE	Program is executing.
<u>H</u> ANDLERS	Program is any type of handler (BNA, WFL, time sharing).
<u>L</u> OADMP	Program is LOADMP intrinsic.
<u>M</u> C S	Program is Message Control System (MCS).
<u>P</u> ACKUP	Program is PACKUP intrinsic.
<u>S</u> HAREHNDL	Program is a timesharing handler program (for example, CANDE).
<u>S</u> ORT	Program is SORT: or SORT. intrinsic.
<u>T</u> IMESHARE	Program is executed as time sharing task.
<u>W</u> FLHNDL	Program is a Work Flow Language (WFL) handler.

UTILITY Program is any type of intrinsic (DMPALL, LOADMP, PACKUP, DSKOUT, SYSTEM/COPY).

CHARGE <charge number>

Selects jobs based on the charge number the job is executed under. You can enter as many as 10 charge number selections and 10 charge number exceptions. The abbreviation CH is accepted for CHARGE.

You can also select jobs based on part of a charge number by using an equal sign as a masking character. Use an equal sign for any unknown characters in the charge number. For example, the following command displays jobs executed under a charge number beginning with the characters 66, but does not display jobs executed under charge numbers 660001 or 669999:

WY CH 66= EXCEPT CH 660001, 669999

WAIT <waiting condition>

Selects jobs that are waiting for a specific condition to be satisfied (for example, waiting for hardware, waiting for memory and so forth). You can enter as many as three waiting conditions and three waiting condition exceptions. The abbreviation W is accepted for WAIT.

The available types of waiting conditions are as follows:

<u>DEVICE</u>	Waiting for hardware device to become available
<u>EVENT</u>	Waiting for a software event to be signaled (for example, STOQUE, CRCR, and so forth)
<u>OPERATOR</u>	Waiting for operator to perform an action
<u>SPACE</u>	Waiting for memory, disk space or diskpack space

EXCEPT

All text following the keyword EXCEPT is treated as exceptions to the selections entered previously. You must group all of the exceptions must be grouped together after the keyword EXCEPT; you cannot enter a selection and an exception followed by another selection.

Details

System responses to the WY command can include the following types of job status:

- EXECUTING - The program is being processed.
- WAITING - The job is waiting for something to happen before it can continue processing. It can be waiting for a message from another program, for a message from the operator, for a message from a data communication device, for a peripheral device to become available, or for additional space in memory.
- TERMINATING - The MCP is going through the routines for ending this job.

Examples

WY

This displays the status of all jobs in the mix.

2 WY

This displays the status of the job associated with mix number 002.

WY PR 1-6 CH 661~~---~~ EXCEPT CH 661000

This displays the status of all jobs with processor priorities from 1 through 6 that are executing under a charge number beginning with 661, except for any jobs executed under a charge number of 661000.

22 WY M 10

This displays the status of the job associated with mix number 022, the last 10 messages that have been sent to that job through the AX command, and/or the last 10 messages that job has displayed on the ODT.

See also:

- The AJ (Display Active Jobs) command on Page 2-14.
- The WJ (Display Waiting Jobs) command on Page 2-411.
- The WS (Display Jobs in Schedule). command on Page 2-423.

XA (Add/Remove a Channel from an Exchange)

Keyboard Input Message

Access Level: 8

The XA command can be used to add or remove a channel from an exchange.

- XA - <exchange number> + <channel number>

The diagram shows the command format: '- XA - <exchange number> + <channel number>'. Brackets are used to group the exchange number and the channel number. An arrow points from the plus sign to the channel number, and another arrow points from the minus sign to the exchange number.

Explanation

- <exchange number> This indicates the exchange to which you are adding or from which you are deleting a channel.
- If you enter an exchange number that has not been previously declared in an EXCHANGE record in the system configuration file, the system creates an exchange with the exchange number as the primary channel, and the channel you were adding as the single alternate channel.
- + <channel number> This indicates the channel you are adding. This channel cannot be a member of another exchange, and must be of a hardware type identical to the primary channel of the exchange.
- <channel number> This indicates the channel you are deleting. This channel cannot be a member of another exchange, and must be of a hardware type identical to the primary channel of the exchange.

Examples

XA 11 + 04

This adds channel 04 to exchange 11.

```
DLP 24 DISK HSTLUB
DLP 25 DISK HSTLUB
DLP 50 DISK HSTLUB
XA 24 + 25, + 50
```

This adds DLPs with channel numbers 24, 25, and 50 to the system environment. Channel number 24 becomes the primary channel and exchange. Channels 25 and 50 are added as alternate channels.

See also:

- The DD (Delete DLP) command on Page 2-120.
- The DLP (Add DLPs) command on Page 2-137.
- The SHOW (Display System Information) command on Page 2-343.
- The EXCHANGE record in the *V Series Systems Operations Guide Volume 1: Installation*.

XC (Inhibit/Uninhibit or Display Status of a Channel)

Keyboard Input Message

Access Level: 8

The XC command inhibits, uninhibits, or displays the status of a channel.



Explanation

XC <channel number> This displays the status of the channel. The status can be one of the following

INHIBITED The channel is inhibited, meaning that it is unavailable to the system.

NOT INHIB The channel is uninhibited, meaning that it is available to the system.

XC + <channel number> This inhibits the specified channel, making it unavailable to the system.

XC - <channel number> This uninhibits the specified channel, making it available to the system.

Details

The XC command is generally used to make a channel available to the system after an unrecoverable channel error has made it unavailable.

Inhibited or uninhibited status is retained across halt/loads.

Restrictions

The XC command cannot be used to inhibit the HLSPO or the primary disk channel.

Examples

XC +5

This inhibits channel number 5, making it unavailable to the system.

XC -4

This uninhibits channel number 4, making it available to the system.

XC 30

This displays the availability status of channel number 30.

See also:

- The XA (Add/Remove a Channel from an Exchange) command on Page 2-432.

XD (Remove Disk Segments)

Keyboard Input Message

Access Level: 8

The XD command removes disk segments from the system.

```
- XD - <ID number> - <beginning segment address> _____  
                                     |_____|  
                                     <length in segments>
```

Explanation

- <ID number> This is the ID number of the disk unit from which segments are to be removed.
- <begin segment address> This is the segment address of the first segment to be removed.
- <length in segments> This is the number of segments to be removed. The default is 1,000 segments.

Restriction

A disk area that is assigned to a file cannot be removed by the XD command. Use the KS (Analyze Disk Space) command to ascertain if a disk area is assigned to a file.

Details

The XD command removes the specified segments from the Disk Available table of the MCP, making them unavailable.

You can return disk segments to the system by using the RXD (Return Deleted Space) command.

Example

```
XD 10 10000 500
```

This deletes segments 10000 through 10499, a total of 500 segments, from disk unit 10.

See also:

- The KS (Analyze Disk Space) command on Page 2-207.
- The RXD (Return Deleted Space) command on Page 2-336.
- The WX (Display Removed Space) command on Page 2-425.
- The XM (Remove Memory from System) command on Page 2-437.
- The XP (Remove Pack Sectors) command on Page 2-438.

XM (Remove Memory from System)

Keyboard Input Message

Access Level: 8

The XM command removes a specified area of memory from the system.

- XM - <beginning address> _____
 └─ <length in kd> ─┘

Explanation

<beginning address> This is the absolute address in kilodigits (kd) of the first location to be removed. If <length> is not specified, 1 kd of memory is removed.

<length in kd> This is the number of memory locations in kilodigits (kd) to be removed. The default is 1 kd.

Restrictions

The memory cannot be in use.

Details

A halt/load subsequent to the XM command does not return the space to the system unless the memory map has been destroyed.

Memory is returned to the system by a cold start or an RXM command.

Example

XM 310 5

This removes 5 kd of memory, starting at absolute address 310000.

See also:

- The RX (Return Deleted Space) command on Page 2-336.
- The WC (Display Available Memory) command on Page 2-406.
- The WX (Display Removed Space) command on Page 2-425.

XP (Remove Pack Sectors)

Keyboard Input Message

Access Level: 8

The XP command removes diskpack sectors from the system.

```
- XP - <cc/u> - <beginning sector address> _____
                                     |_____
                                     |<length in sectors>|
```

Explanation

<cc/u>	This is the channel number and unit number of the diskpack drive from which sectors are to be removed.
<beginning sector address>	This is the location in decimal of the first sector to be removed. The limits for <beginning sector address> are 1 through 16700000 or subject to drive limitations. If <length in sectors> is not specified, 1000 sectors are removed.
<length in sectors>	This is the number of sectors in decimal to be removed. The limits for <length in sectors> are 1 through 16700000 or subject to drive limitations. The default is 1000 sectors.

Restrictions

The diskpack must not be in use.

An area that is assigned to a file cannot be removed by the XP command.

Details

The XP command removes the diskpack sectors from the master available table of the diskpack, making these sectors unavailable for assignment. The PS (Analyze Pack Space) command can be used to ascertain unavailable sectors.

You can return diskpack sectors to the system by using an RXP (Return Deleted Space) command.

Examples

```
XP 3/12 2000 1500
```

This deletes sectors 2000 through 3499, a total of 1500 sectors, from the diskpack that is mounted on the drive that is declared as unit 12 on channel 3.

See also:

- The PS (Analyze Pack Space) command on Page 2-303.
- The RX (Return Deleted Space) command on Page 2-336.
- The WX (Display Removed Space) command on Page 2-425.

()

()

()

Appendix A

Understanding Railroad Diagrams

Introduction

Railroad diagrams show you the rules for putting words and symbols together into commands and statements that the computer can understand. These diagrams consist of a series of paths that show the allowable structure, constants, and variables for a command or a statement. Paths, which are represented by horizontal and vertical lines, show the order in which the command or statement is constructed. Many railroad diagrams have a number of different paths.

Example



If you follow this railroad diagram from left to right, you will discover three acceptable commands. These commands are

REMOVE

REMOVE SOURCE

REMOVE OBJECT

Some of the actual railroad diagrams you will encounter might be more complex. Regardless of the level of complexity, all railroad diagrams are visual representations of commands and statements. Railroad diagrams are intended to:

- Show the mandatory items.
- Show the user-selected items.
- Present the order in which the items must appear.
- Show the number of times an item can be repeated.
- Show the necessary punctuation.

This appendix describes the elements of the diagrams and provides examples.

Constants and Variables

A constant is an item that cannot be altered. You must enter the constant as it appears in the diagram. If a constant is partially underlined, you can abbreviate the constant by entering only the underlined letters. In addition to the underlined letters, any of the remaining letters can be entered. If no part of the constant is underlined, the constant cannot be abbreviated. Constants can be recognized by the fact that they are never enclosed in angle brackets (< >) and are in uppercase letters.

A variable is an item that represents data. You can replace the variable with data that meets the requirements of the particular command or statement. When you replace a variable with data, you must follow the rules defined for the particular command or statement. All variables in railroad diagrams are enclosed in angle brackets (< >).

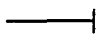
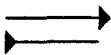
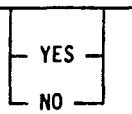
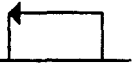
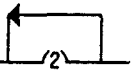
In the following example, BEGIN and END are constants and <statement list> is a variable. The constant BEGIN can be abbreviated because it is partially underlined. Valid abbreviations for BEGIN are BE, BEG, and BEGI.

— BEGIN —<statement list>— END —————|

Constraints

Constraints are used in a railroad diagram to control progression through the diagram. Constraints consist of symbols and unique railroad diagram line paths.

The following figure illustrates the constraints used in railroad diagrams.

Symbol	Explanation
	Vertical bar. Indicates that the command or statement can be followed by another command or statement.
	Right arrow. Indicates that the diagram occupies more than one line.
—<required>—	Required item. Indicates the constants, variables, and punctuation that must be entered in a command or statement.
	User-selected items. You select the item or items to include.
	Loop. Indicates that an item or group of items can be repeated.
	Bridge. Indicates the maximum number of times an item can be repeated. If an asterisk (*) immediately precedes the number in a bridge, then the item associated with the bridge <i>must</i> be specified at least once.

The following paragraphs describe the constraints.

Vertical Bar

The vertical bar symbol (|) represents the end of a railroad diagram and indicates the command or statement can be followed by another command or statement.

— SECONDWORD — (—<arithmetic expression>—) —————|

Right Arrow

The right arrow symbol (▶) is used when the railroad diagram is too long to fit on one line and must continue on the next. A right arrow appears at the end of the first line and another right arrow appears at the beginning of the next line.

— SCALERIGHT — (—<arithmetic expression>— , —————▶
▶<arithmetic expression>—) —————|

Required Items

A required item can be either a constant, a variable, or punctuation. A required item appears as a single entry, by itself or with other items, on a horizontal line. Required items can also exist on horizontal lines within alternate paths or nested (lower-level) diagrams. If the path you are following contains a required item, you must enter the item in the command or statement.

In the following example, the word EVENT is a required constant and <identifier> is a required variable:

— EVENT —<identifier>—————|

User-Selected Items

User-selected items appear one below the other in a vertical list. You can choose any one of the items from the list. If the list also contains an empty path (solid line), none of the choices are required. A user-selected item can be either a constant, a variable, or punctuation. In the following railroad diagram, either the plus sign (+) or minus sign (-) can be entered before the required variable <arithmetic expression>, or the symbols can be disregarded because the diagram also contains an empty path.

+
-

 <arithmetic expression>—————|

Understanding Railroad Diagrams

Loop

A loop represents an item or group of items that you can repeat. A loop can span all or part of a railroad diagram. It always consists of at least two horizontal lines, one below the other, connected on both sides by vertical lines. The top line is a right-to-left path that contains information about repeating the loop.

Some loops include a return character. A return character is a character (often a comma or semicolon) required before each repetition of a loop. If there is no return character, the items must be separated by one or more blank spaces.



Bridge

Sometimes a loop also includes a bridge, which is used to show the maximum number of times an item can be repeated. The bridge can precede the contents of the loop, or it can precede the return character (if any) on the upper line of the loop.

The bridge determines the number of times you can cross that point in the diagram. The bridge is an integer enclosed in sloping lines (`/` `\`). Not all loops have bridges. Those that do not can be repeated any number of times until all valid entries have been used.

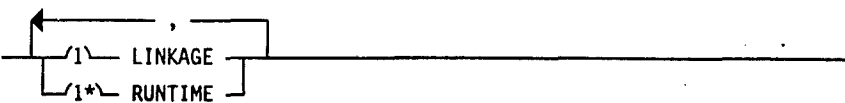
An asterisk immediately following the bridge denotes that the item associated with the bridge must be used at least once.



or



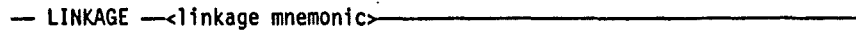
Required Item



In the first bridge example, you can enter LINKAGE or RUNTIME no more than two times. In the second bridge example, you can enter LINKAGE or RUNTIME no more than three times. In the third bridge example, you can enter both LINKAGE and RUNTIME no more than once, and you must enter RUNTIME at least once.

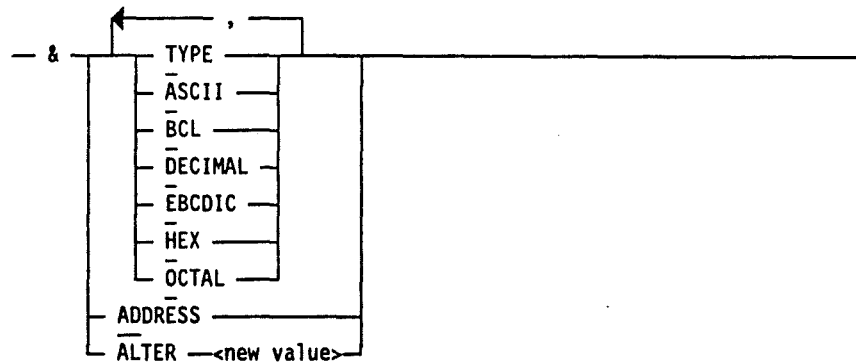
Following the Paths of a Railroad Diagram

The paths of a railroad diagram lead you through the command or statement from beginning to end. Some railroad diagrams have only one path, while others have several alternate paths. The following railroad diagram indicates there is only one path, which requires the constant LINKAGE and the variable <linkage mnemonic>:



Alternate paths provide choices in the construction of commands and statements. Alternate paths are provided by loops, user selected items, or a combination of both. More complex railroad diagrams can consist of many alternate paths, or nested (lower-level) diagrams, that show a further level of detail.

For example, the following railroad diagram consists of a top path and two alternate paths. The top path includes an ampersand (&) and the constants (that are user-selected items) in the vertical list. These constants are within a loop that can be repeated any number of times until all options have been selected. The first alternate path requires the ampersand (&) and the required constant ADDRESS. The second alternate path requires the ampersand (&) followed by the required constant ALTER and the required variable <new value>.



Railroad Diagram Examples

The following examples show five railroad diagrams and possible command and statement constructions based on the paths of these diagrams.

Example 1

<lock statement>

— LOCK — (— <file identifier> —) —————|

Sample Input

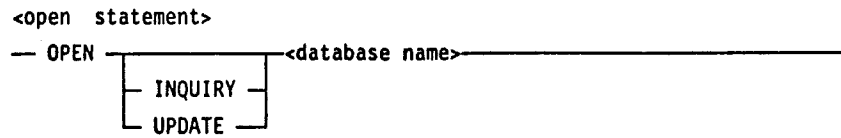
LOCK (F1)

LOCK (FILE4)

Explanation

LOCK is a constant and cannot be altered. Because no part of the word is underlined, the entire word must be entered. The parentheses are required punctuation and F1 and FILE4 are examples of values for the variable <file identifier>.

Example 2



Sample Input

OPEN DATABASE1

OPEN INQUIRY DATABASE1

OPEN UPDATE DATABASE1

Explanation

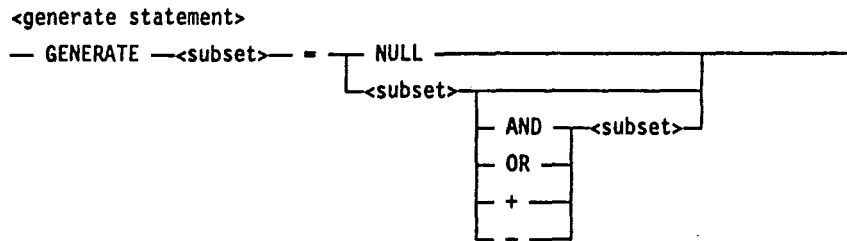
The first sample input shows the constant OPEN followed by the variable DATABASE1, which is a database name. The railroad diagram shows two user-selected items, INQUIRY and UPDATE. However, because there is an empty path (solid line), these entries are not required.

The second sample input shows the constant OPEN followed by the user-selected constant INQUIRY and the variable DATABASE1.

The third sample input shows the constant OPEN followed by the user-selected constant UPDATE and the variable DATABASE1.

Understanding Railroad Diagrams

Example 3



Sample Input

GENERATE Z = NULL

GENERATE Z = X

GENERATE Z = X AND B

GENERATE Z = X + B

Explanation

The first sample input shows the **GENERATE** constant followed by the variable **Z**, an equal sign, and the user-selected constant **NULL**.

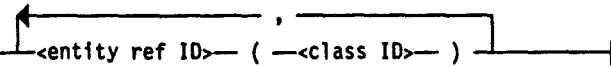
The second sample input shows the **GENERATE** constant followed by the variable **Z**, an equal sign, and the user-selected variable **X**.

The third sample input shows the **GENERATE** constant followed by the variable **Z**, an equal sign, the user-selected variable **X**, the **AND** command (from the list of user-selected items in the nested path), and a third variable, **B**.

The fourth sample input shows the **GENERATE** constant followed by the variable **Z**, an equal sign, the user-selected variable **X**, the plus sign (from the list of user-selected items in the nested path), and a third variable, **B**.

Example 4

<entity reference declaration>

— ENTITY REFERENCE — 

Sample Input

ENTITY REFERENCE ADVISOR1 (INSTRUCTOR)

ENTITY REFERENCE ADVISOR1 (INSTRUCTOR), ADVISOR2, (ASST_INSTRUCTOR)

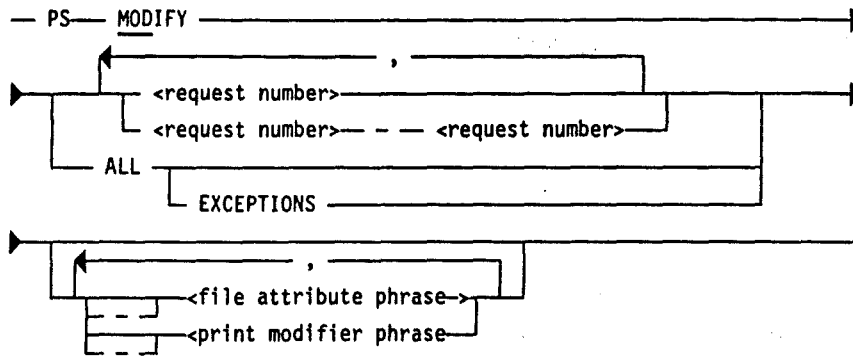
Explanation

The first sample input shows the required item ENTITY REFERENCE followed by the variable ADVISOR1 and the variable INSTRUCTOR. The parentheses are required.

The second sample input illustrates the use of a loop by showing the same input as in the first sample followed by a comma, the variable ADVISOR2, and the variable ASST_INSTRUCTOR. The parentheses are required.

Understanding Railroad Diagrams

Example 5



Sample Input

```

PS MODIFY 11159

PS MODIFY 11159,11160,11163

PS MODIFY 11159-11161 DESTINATION = "LP7"

PS MOD ALL EXCEPTIONS

```

Explanation

The first sample input shows the constants PS and MODIFY followed by the variable 11159, which is a <request number>.

The second sample input illustrates the use of a loop by showing the same input as in the first sample followed by a comma, the variable 11160, another comma, and the final variable 11163.

The third sample input shows the constants PS and MODIFY followed by the user-selected variables 11159-11161, which are <request number>s, and the user-selected variable DESTINATION = "LP7 ", which is a <file attribute phrase>.

The fourth sample input shows the constants PS and MODIFY followed by the user-selected constant ALL, followed by the user-selected constant EXCEPTIONS. Note that in this sample input, the constant MODIFY has been abbreviated.

Appendix B

Commands for User-Coded Utilities

Unisys-supplied LOADMP, PACKUP, PBDOUT, and PBTOUT utilities are not available under MCP/VS 2.0 and subsequent releases. LOADMP and PACKUP have been replaced by the SYSTEM/COPY utility, which is initiated by the COPY, COMPARE, COPYADD, MERGE, and MOVE commands. PBDOUT and PBTOUT have been replaced by the PBDPRN utility, which is initiated by the PRINT (Print Files) command.

When you enter commands to invoke LOADMP, PACKUP, PBDOUT, and PBTOUT, the MCP passes the command strings to user-written programs with the same names. If the appropriate program is not present on disk, the command is ignored and an error message displayed.

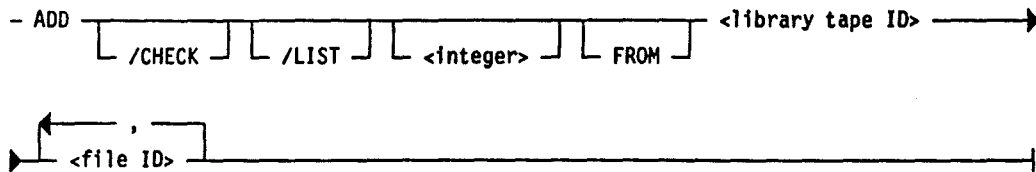
This appendix contains the syntax of those commands that initiate the LOADMP, PACKUP, PBDOUT, and PBTOUT utilities. These commands will become unavailable at a later release.

ADD (Copy Files)

The ADD command invokes LOADMP to transfer files from a library tape to disk, or invokes PACKUP to transfer files to or from diskpack.

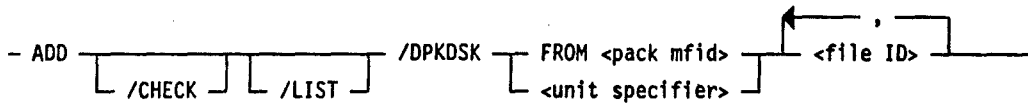
Copy Tape to Disk

To copy from a library tape to disk, use the following format:



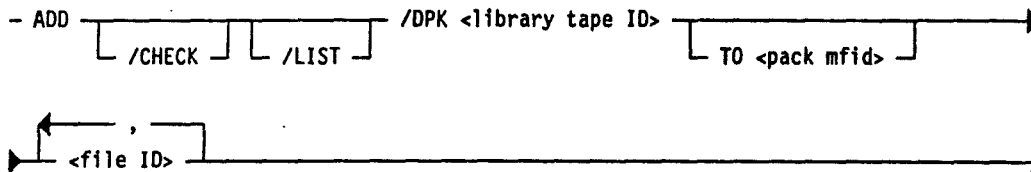
Copy Diskpack to Disk

To copy from a diskpack to a disk, use the following format:



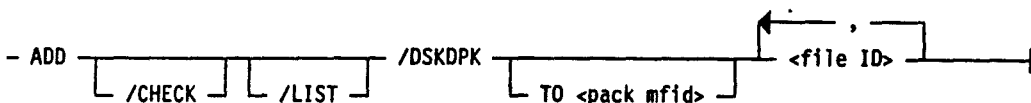
Copy Diskpack Library Tape to Diskpack

To copy from a diskpack library tape to a diskpack, use the following format:



Copy Disk to Diskpack

To copy from a disk to a diskpack, use the following format:



Copy Diskpack to Diskpack

To copy from a diskpack to another diskpack, use the following format:

- ADD /CHECK /LIST /DPKDPK FROM <pack mfid> <unit specifier> TO <pack mfid>
<file ID>

Copy Disk Library Tape to Diskpack

To copy from a disk library tape to a diskpack, use the following format:

- ADD /CHECK /LIST /MTPDPK <library tape ID> TO <pack mfid>
<file ID>

Copy Diskpack Library Tape to Disk

To copy a diskpack library tape to a disk, use the following format:

- ADD /CHECK /LIST /MTPDSK <library tape ID> <file ID>

CHECK (Verify)

The CHECK command verifies the validity of a library tape. CHECK initiates LOADMP if files are to be checked on disk, or PACKUP if files are to be checked on diskpack.

The syntax for CHECK is:

- CHECK _____ <library tape ID> _____
 [/COMPARE] [/LIST] [/DPK]

DUMP (Copy to Library Tape)

The DUMP command copies disk or diskpack files without removing the files. DUMP initiates LOADMP if the files are to be copied from disk, or PACKUP if files are to be copied from diskpack.

Dump Disk to Library Tape

To copy from disk to library tape, use the following format:

```
- DUMP [ /CHECK ] [ /COMPARE ] [ /LIST ] [ TO ] <library tape ID> [ , <file ID> ]
```

Dump Disk to Diskpack

To copy from disk to diskpack, use the following format:

```
- DUMP [ /CHECK ] [ /LIST ] /DSKDPK [ TO <pack ID> ] [ , <file ID> ]
```

Dump Diskpack to Library Tape

To copy from diskpack to library tape, use the following format:

```
- DUMP [ /CHECK ] [ /LIST ] /DPK - <library tape ID> [ FROM <pack mfid> ] [ <unit specifier> ] [ , <file ID> ]
```

Dump Diskpack to Disk

To copy from diskpack to disk, use the following format:

```
- DUMP [ /CHECK ] [ /LIST ] /DPKDSK [ FROM <pack mfid> ] [ <unit specifier> ] [ , <file ID> ]
```

Commands for User-Coded Utilities

Dump Diskpack to Diskpack

To copy a file from a diskpack to another diskpack, use the following format:

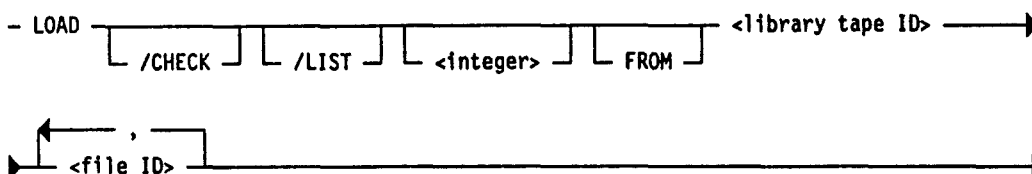
- DUMP [/CHECK] [/LIST] /DPKOPK [FROM <pack mfid>] [<unit specifier> - TO <pack mfid>]
[<file ID>]

LOAD

The LOAD command copies files, replacing any existing file with any file on the tape having the same file name.

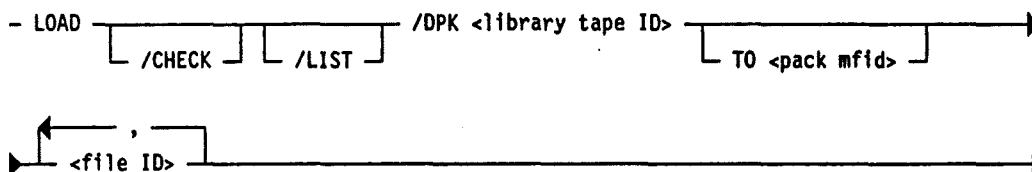
Copy Library Tape to Disk

To copy a file from library tape to disk, use the following format:



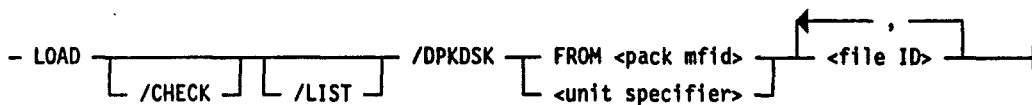
Copy Diskpack Library Tape to Diskpack

To copy a file from a diskpack library tape to a diskpack, use the following format:



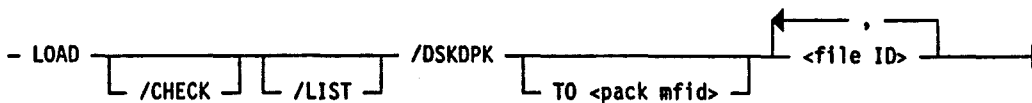
Copy Diskpack to Disk

To copy a file from diskpack to disk, use the following format:



Copy Disk to Diskpack

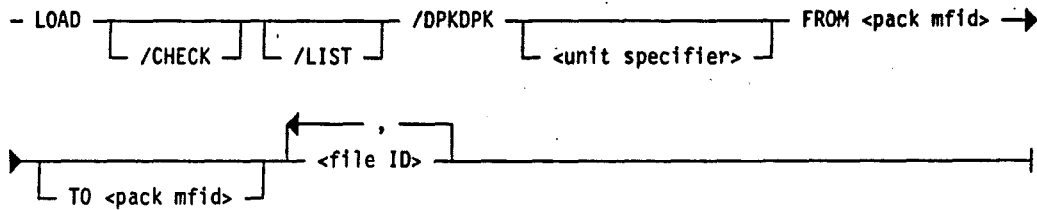
To copy a file from disk to diskpack, use the following format:



Commands for User-Coded Utilities

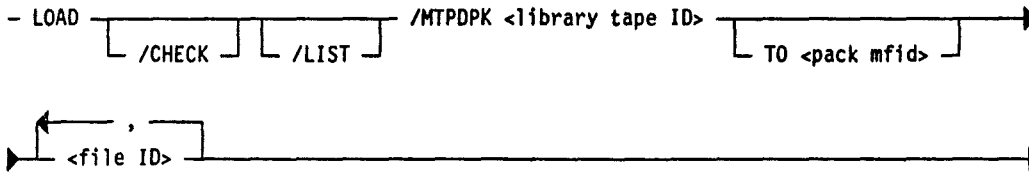
Copy Diskpack to Diskpack

To copy a file from diskpack to diskpack, use the following format:



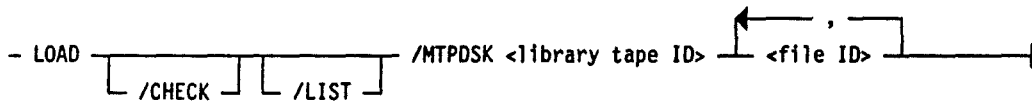
Copy Disk Library Tape to Diskpack

To copy a file from disk library tape to diskpack, use the following format:



Copy Diskpack Library Tape to Disk

To copy a file from diskpack library tape to disk, use the following format:

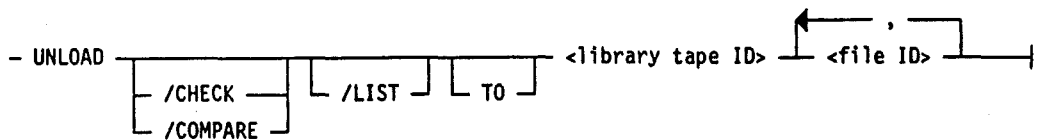


UNLOAD (Move Files)

The UNLOAD command copies files and removes the file identifiers after completing the operation.

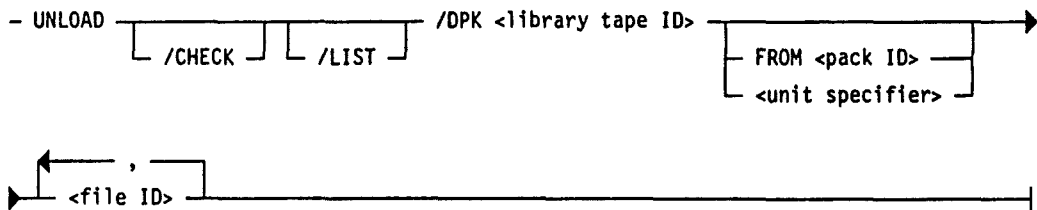
UNLOAD Disk or Diskpack to Tape

To unload from disk or diskpack to tape, use the following format:



UNLOAD Diskpack to Tape

To unload from diskpack to tape, use the following format:



UNLOAD Disk to Diskpack

Use the following format to unload from disk to diskpack:



UNLOAD Diskpack to Disk

Use the following format to unload diskpack to disk:



Commands for User-Coded Utilities

UNLOAD Diskpack to Diskpack

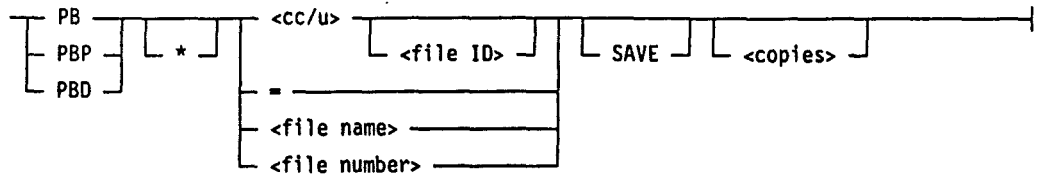
Use the following format to unload diskpack to diskpack:

- UNLOAD [/CHECK] [/LIST] /DPKDPK [FROM <pack mfid>] [<unit specifier>] TO <pack ID> →
▶ [<file ID>]

PBDOUT and PBTOUT (Print Backup Files)

The PB (Print Backup File) command initiates printing of printer backup files. If the printer backup file is on tape, PBTOUT is scheduled. If the file is on disk or diskpack, PBDOUT is scheduled.

The format for the PB command is:



()

()

()

Glossary

#

As the first character in a file name, identifies a pseudo card deck (control deck).

\$

As the first character in a file name, identifies a dump file.

*

As the first character in a file name, identifies a punch backup file.

@

As the first character in a file name, identifies a printer backup file.

=

See masking character.

5N disk

A head-per-track storage medium that has a capacity of 5.5 megabytes.

A

access code

An identification code consisting of a usercode, a password, and a charge number that a user enters when logging on, or when entering a job initiation command. Normally used only on systems that have System Security. *See also* user code, password, charge number.

access level

A security level established for each remote terminal and/or access code. In CANDE, the access level for each station is selected in the USERHO deck. Each system command can be entered only from a terminal with a particular access level (or with a higher access level).

active mix

Those jobs in the mix that are not waiting.

ACU

See automatic calling unit.

adapter ID

A name used to identify a data communication line connected to the system with a Uniline data link processor (DLP).

after-linkage – available table

after-linkage

A condition in which a specified task is scheduled to start only after another specified task has been completed.

alphanumeric

Any character in the system's character set.

APCR

Refers to the automatic pseudo card reader option.

area

The smallest amount of disk space occupied by a file. The size of an area is defined by the user.

ASCII (American Standard Code for Information Interchange)

A coded character set of 128 control and graphic characters in which each character is represented by a 7-bit code (8 bits plus parity). ASCII is defined by American National Standard X3.4-1967.

asynchronous

Transmission at an arbitrary rate. A character, or block of characters, is surrounded by start and stop bits from which a receiver derives the necessary timing.

asynchronous process

A procedure initiated by a program that runs independently of the initiating program.

audit

In DMSII, to keep track of each transaction in order to ease recovery from a failure.

AUHL option

See automatic halt/load option.

automatic calling unit (ACU)

A device that permits a modem or terminal to dial calls automatically, and thereby establish a link over a communications network.

automatic halt/load (AUHL) option

An option that causes an automatic halt/load when a system failure occurs, unless the failure occurs during a halt/load.

available table

A table maintained by the MCP that keeps track of the space available on each disk or disk pack in the system.

B

backup

(1) A copy of a file that can be used if the original becomes corrupted. (2) A description of a file temporarily stored on disk or disk pack that is output to a unit-record device (printer, card punch).

back up

To create a backup file.

bad spots

Unusable sectors on a disk or disk pack.

base area

The first area of a disk pack file. The base areas for all files of a disk pack family are on the master disk pack.

base disk pack

See master disk pack.

base relative address

An address within a program that is an offset from the beginning of the program rather than an absolute memory address.

BCL

A Unisys common character set in which each character is represented by a 6-bit code.

BCT

See branch communicate.

BDLC data link control

A bit-oriented data link control protocol that is compatible with the high-level data link control (HDLC) protocol defined by the International Standards Organization. BDLC stations support two-way simultaneous operation over dedicated, full-duplex lines, and two-way alternate operation over switched, half-duplex lines.

beginning of job (BOJ)

The start of job processing.

beginning of tape (BOT)

A location (indicated by a reflective marker) that indicates the beginning of the usable portion of the tape. The reflective marker is placed 15 to 25 feet from the physical beginning of the tape. Contrast with end of tape (EOT).

binary configuration

Binary addressing of sectors on disk or disk pack.

bisync

Unisys Binary Synchronous Communication protocol, a character-oriented data link control protocol similar to the IBM® BSC protocol. Bisync is designed for two-way alternate operation over either point-to-point or multipoint links. Contrast with BDLC.

bit

Binary digit. A bit is the most basic unit of computer information. A bit can have one of two values: 0 (off or false) or 1 (on or true).

bits per inch (bpi)

(1) The information-packing density of magnetic tape. (2) The density at which a magnetic tape unit can read and write.

blackout feature

A security feature that inhibits the display of an access code as it is entered during the log-in procedure.

block

(1) Physically-adjacent records that are transferred to or from a device as a group. (2) In data communications, a group of characters or bits sent as a unit.

blocking factor

The number of records per block.

block lockout table (BLT)

A table that keeps track of which file blocks are locked by which programs. The BLT resides in memory and is updated by the MCP every time a program locks or unlocks a file block. The BLT LIMIT record sets the maximum number of entries this table can have.

BLOCKSIZE

A file attribute that gives the length of a block.

BLT

See block lockout table.

BLT contention

A condition that occurs when one program wants to access a file block that is locked by another program running on the same processor. *See also* contention.

BNA

An overall set of design principles (that is, an architecture) under which Unisys systems can be joined into data communications networks. Each Unisys system that can be incorporated into a network has specific hardware and/or software conforming to this architecture.

BOF

beginning of file.

BOJ

See beginning of job.

BOT

See beginning of tape.

bpi

See bits per inch.

BPL programming language

An ALGOL-like language for V Series.

branch communicate (BCT)

A V Series machine instruction that causes control to be transferred from a user program to the MCP.

branch reinstate virtual (BRV)

An instruction used to reinstate a user task. It is executed only by the MCP's dispatching routine in the MCP kernel.

BRV

See branch reinstate virtual.

buffer

An area in which data is stored temporarily for transfer from one storage medium to another.

byte

Eight consecutive bits.

C

CANDE

See Command and Edit system.

CANDE Editor files

Files of the Editor program, which is a part of the Command and Edit (CANDE) system.

card

(1) A punched card used with a card reader to input the system configuration files. (2) Synonym for record when describing configuration records and control commands. *See also* card deck, card reader.

card deck

A set of punched card. This term usually refers to a number of cards comprising the data input to a program.

cardless system

A system that uses flexible disk operations instead of card reader operations for system initialization.

card reader

An input device that reads data from punched cards. *See also* card, card deck.

CCITT – concurrent file access

CCITT

See International Telegraph and Telephone Consultative Committee.

cc/uu

Represents the channel and unit address of a peripheral device. The *cc* is a 1- or 2-digit channel designation, and the *uu* is a 1- or 2-digit unit designation. The slash is required; spaces are not permitted. Tape drive unit numbers generally begin with 1; all other device designations usually begin with 0.

central processor unit (CPU)

The hardware unit that executes instructions.

channel

(1) In data communication, a communication path used to transmit signals between two or more points. (2) The data link processor (DLP) in the processor cabinet or standalone I/O cabinet that handles communication along a path. *See also* data link processor.

chargeable time

An amount calculated from direct time, input/output (I/O) time, and the current job's share of system overhead.

charge number

Charges for system use are assigned to this number. The charge number is assigned by the system administrator. *See also* access code.

clobber

A condition when a file is overwritten, destroyed, or otherwise made unuseable.

code file

A file produced by a compiler when a program is compiled. The file contains instructions in machine code. A code file is also referred to as an object code file.

cold-start

The process by which the MCP is loaded to system disk. A cold-start reinitializes all data structures on a disk, causing any information about existing files on that disk to be lost.

Command and Edit (CANDE) system

CANDE is a timesharing system that provides generalized file preparation and updating capabilities and task control in an interactive, terminal-oriented environment.

compile-and-go operation

A compile operation that does not produce a permanent code file.

concurrent file access

The ability of two or more programs to simultaneously read and update the same file.

CONFIG#

The name of the system configuration file. The pound sign (#) represents the system number.

CONFIG

See Configuration File utility (CONFIG).

Configuration File utility (CONFIG)

CONFIG is an interactive utility that allows you to create and modify a system configuration file, which is used to cold-start a system.

console DLP

A data link processor (DLP) used by the system to access the ODT, flexible disk drive, DLP test bus, and remote links.

contention

(1) A situation that occurs when multiple tasks compete for exclusive use of a resource, such as an I/O device or a record in a file. (2) In data communications, a situation in which multiple users vie for the right to use a transmission channel. (3) A situation that occurs when two or more tasks attempt to lock the same block in a file.

contention mode

In data communications, a line condition in which no station is designated as a master station, and each station on the line must monitor signals on the line and wait for an idle condition before bidding for master status.

contention overtime

The period of time a program can contend for a file block. *See also* stalemate.

contiguous items

In COBOL, items that are described by consecutive entries in the DATA DIVISION and that have a definite hierarchical relationship to each other.

continuation area

An area in a disk pack file that is allocated on a disk pack other than the disk pack on which the base area was originally located.

control command

(1) A Command and Edit (CANDE) input message that begins with a specific character, usually a question mark, and is used to control or interrogate the CANDE operating environment. A control command can be entered from any attached CANDE station. (2) An MCP control instruction.

control deck

A disk or disk pack file that contains 96 bytes per record, and six or eight records per block; a control deck generally contains the information needed to compile or execute a program.

control instruction

See MCP control instruction.

controller – data circuit terminating equipment (DCE)

controller

Provides the interface between the data link processor (DLP) in the host and the mass storage units (such as disk, disk pack, and tape units).

control point

In DMSII, a time at which no user program is in a transaction state, and all buffers are written to disk pack.

control store

The random-access memory (RAM) that contains the firmware.

core-to-core transfer (CRCR)

A method of transferring information between executing programs. Each programming language has its own syntax for the CRCR feature. CRCR transfer is synchronous; one program offers information in a core-to-core transfer and waits until another program retrieves it. The first program cannot continue processing until the information is retrieved.

CPU

See central processor unit.

CRCR

See core-to-core transfer.

CSE

Customer Service Engineer.

CTLDCCK

Control deck. A reserved file identifier. CTLDCCK is the name of the physical card deck used by the LDCNTL utility to build a pseudo card deck.

cylinder

All the tracks on all the platters of a disk or disk pack that are at the same radius.

D

DASDL

See Data and Structure Definition Language.

Data and Structure Definition Language (DASDL)

A language used by a database administrator to describe the physical and logical characteristics of a data base, and the criteria to be used to ensure the integrity and security of the data contained within it.

data circuit terminating equipment (DCE)

A device connected to data terminal equipment (DTE) on one end and to an X.25 network on the other end. The DCE serves as the entry point to the X.25 network, similar to a modem.

data comm

Synonym for data communication.

data communication

The bidirectional transfer of data between one or more devices and a central computer.

data communication device

(1) A device that routes information between the computer and peripheral devices, such as data communication processors, DLPs, and Uniline controllers.
(2) A terminal.

data communication processor (DCP)

A front-end processor that controls a data communication network, reducing the workload of the central processor.

data link processor (DLP)

A processor that serves as the system interface to a peripheral device, a peripheral controller, or data communication lines.

data terminal equipment (DTE)

(1) In X.25, the functional unit of a data station that establishes, maintains, and releases a connection and provides code and signal conversion between the data station and the transmission line. (2) User equipment, such as a host computer, front-end processor, concentrator, cluster controller, intelligent terminal, or any network entry access point. (3) An end-user device that connects to the data circuit terminating equipment (DCE), which either generates or receives data carried by the network.

database

An integrated, centralized system of data files that several applications can access and update concurrently. *See also* Database Management System II.

Database Management System II (DMSII)

DMSII is a network type database system with hierarchical features, and with features that can mimic attributes of a relational database.

database program (DBP)

A part of the DMSII that performs all functions necessary for accessing and updating the database to which it is attached. *See also* production DBP.

DBP

See database program.

DCE

See data circuit terminating equipment.

DCOM

(1) The data communication system option. (2) The name of the module that carries out data communication functions.

DCP – disk file name

DCP

See data communication processor.

DCP code file

A file containing microcode loaded into a data communications processor (DCP).

deck number

(1) The last four digits of a pseudo card reader file identifier. (2) The full 5-digit portion of the file identifier used to refer to a deck created on a specified system of a multiple system configuration.

default

The value automatically given to a variable when no other value has been assigned.

device

A hardware device, such as a disk drive, that writes data to or reads data from a medium. *See also* medium.

device dependent port (DDP)

The physical connection of the controller to the drives and the host system.

DFHDR

Disk file header.

diagnostics

A test used to detect malfunctions and specify their causes.

digit

(1) An integer from 0 to 9, inclusive. (2) A grouping of four consecutive bits, representing decimal 0 to 9 and hexadecimal A to F.

directory

A table of contents listing the files contained on a device. The device is usually a disk, a disk pack, or a tape.

direct processor time

See processor time.

direct time

See processor time.

disk

A mass storage unit consisting of 100 bytes per sector. *Contrast with* disk pack.

disk file

A file stored on 100-byte-per-sector disk media. The characters per record and records per block (blocking factor) can be set by the programmer.

disk file name

See file identifier.

disk pack

A mass storage unit consisting of 180 bytes per sector. *Contrast with disk.*

disk pack configuration

An arrangement of disk pack units. A naming convention is used when discussing disk packs. Such terms as 2×8 (two-by-eight) or 4×16 (four-by-sixteen) are used to refer to controllers and exchanges and the number of disk pack units on a subsystem. The first number refers to the number of I/O paths (DLPs) coming from the host system to the controller. The second number refers to the number of spindles (not drives) linked to that controller. The maximum number of spindles on a single subsystem is 16. The maximum number of paths to a single subsystem is dependent on the model.

disk pack controller

A standalone unit that controls data flow between the DLP in the host system and the individual disk pack spindles.

disk pack directory

See directory.

disk pack drive

A cabinet containing one or two disk pack spindles. *See also* disk pack spindle.

disk pack family

Two or more disk packs that have the same name. *Also see* family name.

disk pack file

A file stored on a disk pack.

disk pack firmware

Special firmware required by the disk pack controller.

disk pack initialization

A process that makes disk packs usable. Disk packs are initialized by the DISPKV or PTD utilities.

disk pack label

Data written on the disk pack at initialization to identify the disk pack. Label fields include the serial number, system access code, disk pack name, system interchange code, and the ID of the owner.

disk pack spindle

The hardware device on which a disk pack is mounted.

disk pack unit

A disk pack spindle.

DLP

See data link processor.

DMPMEM

A program used to create a raw memory dump on magnetic tape. *See* dump file.

DMSII – end-of-text character (ETX)

DMSII

See Database Management System II.

drive

The hardware device on which a disk or tape is mounted.

DTE

See data terminal equipment.

DTM

Data transfer module.

dual host

A disk controller feature that allows two DLPs to access a device (but not simultaneously).

dual port

A device that links a disk or disk pack unit to two separate controllers, providing path redundancy.

dump file

A file containing the contents of memory, often created for analysis at the time of a failure.

E

EBCDIC

See Extended Binary Coded Decimal Interchange Code.

ECM log

A system log file maintained on Unisys V 500 systems. The ECM log contains information about temperature, voltage, and other conditions on the system. The data in the ECM log originates in the Environmental Control Module (ECM). The ECM log is maintained on the hard disk of the system console and is transferred to the host system by system commands.

end of file (EOF)

The physical limit of a data file.

end of job (EOJ)

(1) A message that a job has finished executing. (2) In X.25, a control code that tells the receiver that a job is completed.

end of tape (EOT)

A reflective strip of tape placed on the inside edge of the backing side of the magnetic tape that indicates to the system that the tape is near the end of the reel. The reflective tape is placed no less than 25 feet from the physical end of the tape.

end-of-text character (ETX)

A character used to signal the end of input (EBCDIC "03").

EOF

See end of file.

EOJ

See end of job.

EOT

See end of tape.

EPROM

erasureable programmable read-only memory.

ETX character

See end-of-text character.

exchange

A device that enables multiple DLPs to be routed to multiple units (usually disk, disk pack, or tape).

execution time

The time required for a task to run to completion. Execution time is also referred to as run time, or in COBOL, as object time.

Extended Binary Coded Decimal Interchange Code (EBCDIC)

A coded character set in which each character is represented by an 8-bit code.

extension module

A part of the operating system that carries out a specialized function and that can be called into memory by the user when this function is needed.

F

family name

A name that identifies a set of mass storage volumes (especially disk pack). *See also volume, disk pack family.*

fault tolerance

The ability of a system to continue processing despite minor errors or failures.

FIB

See file information block.

file block contention

A situation where one program attempts to lock a block that is already locked by another program. The former program is suspended and must wait until the block is unlocked. The suspended program is said to be contending.

file header

A collection of file attributes on disk or disk pack that contains the physical characteristics of a file, such as the name of the file and the number of records per block.

file identifier – GCR density

file identifier

The filename. The name contained in the directory for a device. A file identifier is six characters long (with trailing blanks provided), but cannot be made up of all nulls, all blanks, or all zeros.

file information block (FIB)

A block of data generated by a compiler that describes a file to the MCP. The FIB contains flags, counters, addresses, and sizes used by the MCP during all phases of file processing.

file name

(1) A synonym for file identifier. (2) In Report Program Generator (RPG), a name or word that designates a set of data items. (3) In COBOL, a user-defined word that names a file described in a file description entry or in a sort-merge file description entry within the FILE SECTION of the DATA DIVISION. *See also* file identifier.

file number

(1) The last 4 digits of the identifier of a printer or punch backup file, or of a memory dump. (2) The full 5-digit portion of the file identifier used to reference a file created on a system in a multi-system shared configuration.

firmware

Microcode stored in a hardware device; used to control its operation.

FLAME

Functional Logical Analysis of Machine Efficiency. A program that reports on V Series performance.

forms mode

A mode of operation for a terminal in which the screen is divided into data fields (where user input is allowed) and display fields (where input is not allowed).

frame

(1) In bit-oriented protocols, the sequence of contiguous bits bracketed by and including opening and closing flag sequences (01111110). (2) A group of bits sent over a communications channel; generally a logical transmission unit sent between data-link-layer entities that contains its own control information for addressing and error checking.

front end processor (FEP)

Refers to a device that controls peripherals or that handles data communications. *See also* controller, data communication processor, data link processor.

full duplex

Two-way, simultaneous transmission.

G

GCR density

See group-coded recording (GCR) density.

GEMCOS

See Generalized Message Control System.

Generalized Message Control System (GEMCOS)

A message control system that can run on Unisys V Series systems to provide generalized file preparation and updating capabilities and task control from an ODT.

Gregorian date

The standard method of expressing the date (month/day/year or day/month/year). *Contrast with* Julian date.

group-coded recording (GCR) density

6250 bpi. GCR tapes can be written and read only by a tape drive that is GCR-capable.

group ID

A file identifier that can have terminating or embedded equal signs as masking characters to indicate a group of files. *See also* masking character.

guard file

A disk or disk pack file that contains information on the security access rights of another disk or disk pack file.

H**half-duplex**

Two-way transmission that occurs in only one direction at a time.

halt/load

A system-initialization procedure that temporarily halts the system and loads the MCP from disk or disk pack to main memory. A halt/load is used to start the machine and to recover from failures. During the halt/load process, the tables and lists used by the operating system are built in memory.

halt/load supervisory printer operator display terminal (HLSPO)

An ODT that displays halt/load messages and fail codes in the event of system failure. The HLSPO can be declared in the UNIT ODT record of the system configuration file, and serves as the terminal for system initialization and the fault handler.

HC2-DLP

See intersystem control.

header link

The pointer to a file header from the directory entry for the file.

hit rate

The percentage of times that a user program overlay was called and found in the user QWIK pool (that is, did not have to be retrieved from disk or disk pack).

H/L – industry compatible minidisk (ICMD)

H/L

See halt/load.

HLSPD

See halt/load supervisory printer operator display terminal.

host

The central mainframe computer in a data communications network. The host is the computer system containing the message control system (MCS) to which a given station is connected.

host attributes

Those attributes that are not common across a network but are applied on a host-by-host basis.

host components

Those components that are not common across a network, but rather, are applied on a host-by-host basis.

hostname

The logical name of a system. The hostname identifies the processor in the Run Log, the ODT Log, the Maintenance Log, and in a BNA network.

host transfer data link processor (HT-DLP)

A data link processor that connects the host to the controller for disk and disk pack drives.

I

ICMD

See industry compatible minidisk.

ID

(1) An identifier, usually a name or number, of a file or storage device. (2) The unit designation that specifies the logical number of the disk pack or disk device. *See also* identifier.

identifier

In COBOL, a data-name followed by the syntactically correct combination of qualifiers, subscripts, and indexes necessary to make unique reference to a data item.

ILPS

See Intelligent Laser Printing System.

independent runner (IR)

An independent runner is a special task initiated by the operating system to perform services for user tasks or for the operating system itself.

industry compatible minidisk (ICMD)

A flexible disk that conforms to industry standards. A floppy disk.

initialize, verify, and relocate (IVR)

A complete disk pack initialization process in which the disk pack controller performs the following: writes sector addresses and gaps in all sectors on the disk pack (initialization); reads and checks all of the addressable sectors on the disk pack for address errors and protection code errors (verification); for any errors found, writes the error sector's address into a spare sector address field, and writes the predefined data pattern into the sector's data area (relocation).

input/output (I/O)

An operation in which the system reads data from or writes data to a file on a peripheral device such as a disk.

input/output processor (IOP)

An asynchronous processor that controls the data flow between main memory and peripherals.

Intelligent Laser Printing System (ILPS)

A software system that controls an image (laser) printer.

interface

(1) A communication path between two units. (2) A set of conventions for passing data and/or information. (3) In Pascal, the name for a list of program elements that a module proposes to share with other modules or programs. Each interface is defined in the module in which its elements are declared and invoked in the program unit that will share the elements.

interlaced file storage

A disk or disk pack format in which the physical sectors on the storage medium are arranged in nonsequential order.

International Telegraph and Telephone Consultative Committee (CCITT)

A committee of the International Telecommunications Union (ITU) that has generated such international standards as Recommendation X.25 for packet-switching networks and Recommendation X.3 for packet assembler-disassemblers.

intersystem connect (ISC)

A system that provides local communication between Unisys processors. ISC provides a high-speed data transfer rate between systems through a DLP connection.

intrinsic

A utility program bound to the MCP. An intrinsic is usually initiated by the MCP, system commands, or user programs. In many cases, the user can provide a program of the same name that is invoked instead of the MCP-intrinsic program.

IOAT

I/O assignment table.

I/O base module

A frame or that houses DLPs. The I/O base module contains the logic used to provide priority-resolution among requesting DLPs.

IOMC – job initiation command

IOMC

I/O memory concentrator.

I/O path

(1) The code in the MCP executed when an I/O is requested. (2) The route I/O operations takes to or from a peripheral device through a channel, a DLP, or any controllers or exchanges that lead to and from the peripheral.

IOP

See I/O processor.

I/O subsystem

(1) The portion of the MCP and other system facilities that are primarily concerned with physical and logical I/O operations. (2) The hardware and software that manages all transfers of information between the MCP and peripheral devices.

I/O wait time

The accumulated time spent waiting for I/O operations to finish.

IOP

See I/O processor.

IPP

Image page printer. A laser printer.

IR

See independent runner.

ISC

See intersystem connect.

IVR

See initialize, verify, and relocate.

J

job

A program initiated by an EXECUTE, COMPILE, or other initiation command. A job can be in the job mix or the schedule, and can be executing, waiting, or stopped. Each job has its own task number. *Synonymous with task.*

job control command

System commands such as FILE, MEMORY, or PRIORITY that specify the execution attribute of a job.

job initiation command

A system command that starts the execution of a job or causes a job to enter the schedule for execution, such as COMPILE, DEBUG, or EXECUTE.

job mix

Synonymous with mix.

job schedule

See schedule.

job status

The status of a job in the mix. The status of a job at a given time could be EXECUTING, WAITING, TERMINATING, or SCHEDULED.

job transfer

A BNA function that allows a job to be transferred across a network, from one system to another.

Julian date

The method for expressing the date in which the days in a year are numbered consecutively. For example, January 1 is Julian date 001, January 2 is Julian date 002, and so forth. *Contrast with* Gregorian date.

K

kb

See kilobyte.

kd

See kilodigit.

kernel

The dispatcher and interrupt-handler of the MCP.

keyboard input message

One of the two types of MCP system commands that allow an operator to interface to the MCP. The other type of command is called an MCP control instruction. Keyboard input messages have a wide range of functions, such as inquiry and system maintenance. *See also* MCP control instruction.

keyword

A reserved word that must be present when the syntax in which the word appears is used.

kilobyte (kb)

One thousand bytes (2000 digits).

kilodigit (kd)

One thousand digits (500 bytes).

label – logical record size

L

label

The beginning or ending record of a mass storage volume or unit record device file.

LAK

See look-alike disk pack.

LDCNTL

A system utility that builds pseudo card decks from physical card decks.

library tape

A tape created by the SYSTEM/COPY program with format and labels suitable for storing files.

LINC

Logic Information Network Compiler. A fourth generation compiler for a language that defines complete data processing systems.

link

A physical connection between two hosts, or between a host and a communications device.

lock

(1) A shared file I/O operation that locks a block, preventing other programs that are using shared I/O operations from updating it. The block remains locked until it is unlocked. (2) A program locked by the LOCK (Lock Program) command or the LP (Lock Program) command when the program is executed. A locked program cannot be discontinued. This prevents unauthorized or inadvertent interference. A program is unlocked by the UP (Unlock Program) command. (3) In DMSII, the LOCK or MODIFY statement in the host language. The LOCK or MODIFY statement reads a record from a data set and locks the record to prevent concurrent modification by another user.

log transfer

The process of closing one log file and opening a new log file.

logical channel

See virtual circuit.

logical channel address

In data communications, either the logical channel number (LCN), the data terminal equipment (DTE) address, or the Network Definition Language (NDL) station used to refer to the logical channel.

logical record size

The amount of data accessed in the execution of one read or write statement in a program.

logical subsystem

A group of one or more 100-byte disk devices from one or more physical subsystems.

look-alike disk pack

A disk pack with a format of 100 bytes per sector.

M

magnetic ink character recognition (MICR)

A check encoding system that automates check handling. Checks are imprinted with magnetic ink characters of a specific typeface and dimension.

Maintenance Log (MLOG)

A file maintained by the MCP that contains a record of hardware and activity failures.

maintenance panel

A "soft" panel consisting of screens displayed on the ODT. The panel is used by Unisys support personnel to load, display, and otherwise manipulate machine state, hardware registers, memory, index registers, and so forth.

maintenance processor (MP)

A device that provides operator and maintenance interfaces to a V Series system. The maintenance processor allows local or remote field engineering personnel to run various levels of diagnostic programs.

masked file name

A string of characters used to select or list one or more files. An equal sign (=) is used as a masking (wildcard) character. For example, the specification A= searches for all files beginning with A.

masking character (=)

An equal sign used as a masking character. The equal sign stands for any character. For example, the specification FIL=2 is relevant to all of the following: FIL02, FIL12, FILA2, FILB2.

MAST

See memory area status table.

master available table (MAT)

A special disk or disk pack file created at initialization time that contains a list of all usable space on the disk or disk pack.

master clear

An electronic signal used to initialize a DLP and associated hardware. Contrast with base clear.

Master Control Program (MCP) – megabyte (MB)

Master Control Program (MCP)

The V Series operating system. The MCP is the heart of the V Series system. It is designed to make optimum use of all system resources to maximize productivity and processing efficiency.

master disk pack

The disk pack which contains the base areas for all files of a disk pack family.

MAT

See master available table.

MB

See megabyte.

MCP

See Master Control Program.

MCP control instruction

Along with keyboard input messages, one of the two types of system commands that allows you to interface with the MCP. The MCP control instructions generally affect only a program in the mix, in that they either initiate or modify a job.

MCP QWIK memory

Memory reserved for MCP overlays. The most recently used MCP overlays are kept in QWIK memory so that they do not have to be accessed from disk when they are needed.

MCP tables

A set of arrays, maintained by the MCP, that assists in controlling job execution and allocating system resources.

MCP/VS

MCP/V Series.

MCS

See message control system.

MDC

memory data card.

medium

A piece of hardware, such as a disk, capable of storing data. A device writes data to or reads data from a medium. *See* device.

Medium Systems Network Definition Language (MSNDL)

A Unisys program that generates MCS or data communication handler programs that run on the host and handle data communication functions.

megabyte (MB)

One million bytes.

memory area status table

Data maintained by the manager of the MCP to record the state of allocated memory.

memory dump

A copy of the contents of memory, sometimes referred to as a memory image. The system produces a memory dump when a problem occurs so that the problem can be analyzed.

menu

A formatted screen that presents the user with a set of selections.

message control system (MCS)

An MCS is a program that controls the local data communications network consisting of the host, the terminals, and possibly a network controller such as a DCP. The purpose of the MCS is to protect user programs from having to perform data communication tasks. The MCS routes each message between a specific terminal and a specific user program. More than one MCS can be on a host, performing different functions. GEMCOS is an example of a V Series MCS.

MICR

See magnetic ink character recognition (MICR).

microcode

See firmware.

microminidisk drive (MMDD)

A drive that is used to read and write to 3.5 inch disks.

minidisk

See industry compatible minidisk.

minidisk image file

A disk file containing the exact data as resident on the flexible disk.

mix

The jobs or tasks that are currently executing. A list maintained by the MCP of all jobs that have begun execution. *Contrast with* schedule.

MIX table

A table used by the MCP to maintain information about all programs in the mix or schedule. The type of information includes wait indicators (that is, reasons a program cannot continue), various time calculations (such as stopped and average RUN/WAIT time and time waiting I/O), and program status.

mix time

The length of time a job is in the mix.

MIXED

An option on UNIT NRZ and UNIT MPE indicating that the two unit declarations actually refer to the same physical drive, accessed in a mutually exclusive manner.

MFID – nonreturn-to-zero (NRZ)

MFID

See multifile identifier.

MLOG

See Maintenance Log.

MMDD

See microminidisk drive.

mnemonic

An easy-to-remember abbreviation, usually of a computer command or type of device.

MP

See maintenance processor.

MPE

See phase-encoded magnetic tape.

MT9

A 9-track ASCII-encoded magnetic tape. *See* nonreturn-to-zero.

MTC

A hardware mnemonic representing magnetic cartridge tape devices.

multifile identifier

Synonym for familyname.

N

NCP

See network communications port.

network communications port (NCP)

A logical entry point into a host system. NCPs are part of the V Series Communication System (VCS).

NDL

Network Definition Language.

Network Log (NLOG)

A system log file maintained on V Series systems that use the V Series Communication System (VCS). The network log contains information about errors and security violations that occur within the VCS data communication network.

NLOG

See Network Log.

nonreturn-to-zero (NRZ)

A tape recording mode; the density is 800 bpi.

nonstatus device (NST)

A device or unit that can be handled only by specialized programs, not by the MCP.

not ready

A hardware state recognized by the MCP. A not ready device cannot be used for I/O operations.

NRZ

See nonreturn-to-zero.

NST

See nonstatus device.

numeric

Comprised of the digits 0 through 9.

O

object code file

See code file.

OCS

See operator control station.

ODT

See operator display terminal.

ODT Log (SLOG)

A file maintained by the MCP that contains all the messages that are displayed on the ODT. *Synonymous with SPO Log (SLOG).*

operator control station (OCS)

This type of terminal is supported by a Uniline DLP. *Contrast with operator display terminal.*

operator display terminal (ODT)

The video terminal through which the operator enters system commands and receives system messages. *Contrast with operator control station.*

overlay

Logical divisions of a program that occupy the same memory area at different periods during the execution.

P

pack

See disk pack.

packet – PDN

packet

In data communication, a unit of data routed from one packet switching exchange to another by transmission line. A packet consists of data surrounded by packet-level headers and trailers. A message can be broken down into packets at one side of the network and reassembled into the original message at the other end.

packet assembler/disassembler (PAD)

Software that allows a teletypewriter-like terminal to send and receive data on a packet-switching network such as an X.25 network.

packet switching exchange (PSE)

In data communications, the switching nodes in a public data network.

packet switching network (PSN)

A data communication network that transmits data by dividing messages into smaller units (packets), transmitting the packets separately to their destinations, and reassembling the message at the receiving end.

parallel transmission

A technique used to send an entire character at one time by using separate channels or lines, or separate frequencies on one channel. *Contrast with serial transmission.*

parameter

A quantity or item that can be given a different value each time a process is repeated. A variable.

password

Normally used only on systems that have system security. A group of characters by which the system determines the legality of a user when a user logs on to the system or enters a job initiation command. The password is assigned by the system administrator.

patch file

A file containing the replacement, insertion, or deletion records that are used to update an existing source file.

path

See I/O path.

PCF

pseudo card reader file.

PCR

pseudo card reader.

PDN

See public data network.

phase-encoded magnetic tape (MPE)

A method of encoding information on a tape at 1600 bytes per inch (bpi). Some tape drives can read and write tapes only in phase-encoded mode.

physical subsystem

The disk drives, disk pack drives, or combination of disk and disk pack drives that are connected to the system through the same DLPs. This arrangement is also referred to as a disk subsystem or a disk pack subsystem.

physical tape number

A serial number that identifies a particular magnetic tape. The number is part of the tape label.

pool size

The amount of memory available for user QWIK memory.

port

A point of data entry into a computer, network, or other electronic device.

printer backup file

A printer file stored on disk, disk pack, or tape for later use.

priority

(1) A characteristic associated with a job that determines its precedence in the use of system resources. The MCP pulls jobs from the schedule into the mix according to priority. (2) In X.25, the sequence in which various entries and tasks are processed by the analysis of action codes and other priority real-time systems.

privileged mode

A program execution mode in which the processor allows the execution of privileged instructions. Usually reserved for the operating system.

processor name

A name given to a processor by the HOSTNAME record in the system configuration file.

processor number

A number given to each processor in a multiprocessor system. This number is used to make certain file names (for example, printer backup files) unique in a multi-system environment.

processor time

The accumulated time a processor has spent working on a job. *Synonym for* direct processor time, direct time.

production DBP

A database program that binds a COBOL program into the DBP, allowing the DBP to access internal file information and record areas without involving the MCP. Other host language programs can access the database through the production DBP.

program name – RAM

program name

The name of the object code file. A program name can have up to six characters; no punctuation is allowed. The first character must be a letter of the alphabet.

programmatic switch

A feature in some languages, such as COBOL, that allows commands to be given to a program during execution.

PSE

See packet switching exchange.

pseudo card deck

See control deck.

PSN

See packet switching network.

public data network (PDN)

A packet-switched network defined as a common carrier by government regulation.

punch backup file

A file containing the same information as a punch file, but held on disk or disk pack. *See* punch file.

punch file

A file of the type required for output on a card punch.

Q

queue

A list of tasks awaiting processing.

QWIK disk

A feature that allows part of system memory to be used like a disk, greatly speeding up access time.

QWIK memory

Memory that is reserved for rapid access to overlays.

QWIK pool

Memory available for user QWIK memory.

R

railroad syntax diagram

A graphic representation of the syntax of a command, utility, or program.

RAM

See random access memory.

random access memory (RAM)

A type of memory that allows the reading and writing of memory without regard to the location of the preceding read or write operation.

R/D

See result descriptor.

ready

A state of a peripheral device in which it can act on read and write commands.

record

(1) A group of logically related items within a file that are treated as a unit. (2) The minimum amount of data read from or written to a file on one execution of a read or write statement in a program.

release

(1) A method of closing a file. (2) A term used to indicate that a file is no longer being used by the program. (3) A particular version of a software package.

release level

A number assigned to a particular version of the MCP.

remote job entry (RJE)

A software package that allows jobs, data, and control commands to be sent from a remote terminal to a central system, and that allows output of data from the central system to be sent to remote peripherals.

restricted disk pack

A disk pack restricted in access to programs having a multifile ID identical to the label name of the disk pack.

restricted master disk pack

A master (base) disk pack that contains only files having a multifile ID identical to the label name of the disk pack.

result descriptor (R/D)

A coded description of the results of an I/O operation.

RJE

See remote job entry.

RLOG

See Run Log.

roll out

An action that occurs if insufficient memory is available for a particular job. In this case, the code file for the job can be moved to disk. The code file is stored on disk and the space it was using in memory is relinquished. Jobs with lower memory priorities are rolled out first. When memory becomes available again, the code file is moved into memory.

Run Log (RLOG) – shared disk pack

Run Log (RLOG)

A file maintained by the MCP that contains a record of all job and file activity on the system.

S

saved

A state in which a peripheral device is reserved for use by a particular program. No other program can use the saved device.

schedule

A list of jobs that are waiting, but have not yet started executing.

SCSI

See small computer systems interface.

sector

A sector is the minimum addressable area on a disk pack or disk. A disk sector is 100 bytes; a disk pack sector is 180 bytes. Sectors are also referred to as segments.

sector address

The logical address of a sector on a given disk or disk pack.

segment

See sector.

SENSITIVEDATA

A file security attribute that indicates that a file contains confidential or sensitive data. When the file is removed, its name is removed from the directory, and the data area will be overwritten with random data. Normally, the file name is removed from the directory, but the data area is not changed, thus allowing the possibility of file recovery.

sequential organization

The logical file structure in which a record is identified by a predecessor-to-successor relationship established when the record is placed in the file.

serial transmission

A technique of sending data one bit at a time, one after the other. *Contrast with* parallel transmission.

shared disk pack

A disk pack shared between multiple systems. The PACK record in the system initialization file can declare a shared disk pack. A shared disk pack must be declared with the same logical ID number and the same shared system processor (SSP) name for each processor.

shared file I/O operations

Multiple systems that access the same files. Shared (concurrent) file access is regulated by a set of shared file I/O operations. By using these I/O operations, programs can lock, read, seek, and write file blocks in a way that prevents data corruption by other programs that are using the same I/O operations.

shared system - disk and disk pack

Two to four V Series systems linked together in such a way that they share all disk devices and one or more disk pack devices.

shared system - disk only

Two to four V Series systems that share disk but do not share disk pack devices.

shared system - disk pack

Two to four V-Series systems linked together in such a way that they can share disk packs but they do not share disk devices. Each system runs its own copy of the same MCP.

shared system initialization records

Records to be included in the system initialization file for shared systems. These initialization records, such as the CONNECT, DLP, and SSP records, are used at cold-start time. They enable the MCP and other programs to perform concurrent file access and to share peripherals.

shared system processor (SSP)

A DLP that links up to four V Series systems so that they can share files on common disk and disk pack units. The systems can run common software and use conventional file structures (with the exception of DMSII files).

signed numeric

A numeric field that contains an algebraic sign (or a default sign).

small computer systems interface (SCSI)

A communications protocol used to communicate with peripheral devices, including magnetic tape devices and disk drives.

SMC

system maintenance control.

SN

See signed numeric.

snap

See snapshot.

snapshot

(1) The creation of a file that describes machine state at the time of a failure. (2) The file thus created.

soft device

Any device that requires a special code (firmware) to be loaded before the device can become operational.

SPO – STOQ

SPO

See operator display terminal.

SPOM

See SPOMESSAGE BCT.

SPOMESSAGE BCT

A programmatic branch communicate that passes keyboard input messages to the MCP and requests that the response be returned to the requestor. Contrast with ZIP and ZIPSPO.

SSP

See shared system processor.

SSP contention

A condition that occurs when a program running on one V Series system wants to access a file block that is locked by a program running on another V Series system.

stacked deck

A condition that occurs when the system configuration files for a shared system are loaded from the same card reader, and the card deck that is put into the card reader contains the system configuration files for more than one system.

stalemate

A situation in which two or more programs cannot continue processing because they have locked and are contending for mutually exclusive records.

start bit

Indicates the beginning of a character, or block of characters, in an asynchronous transmission. *See asynchronous.*

start/stop transmission

See asynchronous.

station

(1) The combination of functional units constituting the data terminal equipment (DTE), data circuit terminating equipment (DCE), and their common interface. (2) In the interactive datacomm configurator (IDC), a data structure that describes the attributes of a physical terminal. (3) The logical representation for definition (1).

statistics

In DMSII, a feature that keeps track of database use.

stop bit

Indicates the end of a character, or block of characters, in an asynchronous transmission. *See asynchronous.*

STOQ

See storage queue.

STOQUE

See storage queue.

storage queue

A storage queue, abbreviated to STOQ or STOQUE, is a buffer in memory used to transfer information from one program to another. It provides asynchronous inter-program communication. This means that if a program sends a message to another program not ready to accept the message, the message waits in the storage queue until it is accepted. Contrast with core-to-core.

string

A connected sequence or group of characters.

switch

See programmatic switch.

T

task

See job.

TBI

test bus interface.

U

user code

The user code is a number that identifies a user to the system when a user logs on to the system or enters a job initiation command.

user QWIK memory

Memory dedicated to storing the most recently used user program overlays so that they will not have to be accessed from disk if they are needed again. *See also* MCP QWIK memory.

utility

A standard program that helps computer operation, such as initializing floppies, or converting files from one medium to another.

V

V Series Communication System (VCS)

Data communication software that connects a V Series system with data communication terminals. VCS software resides and executes on a V Series system. *See also* network communications port, Network Log.

version date

The date a program was compiled.

virtual circuit – ZIPSPO

virtual circuit

A path between two devices in a packet switching network.

virtual static imaging device (VSID)

A device that produces high-speed, high-quality printouts (for example, a laser printer).

volume

A named medium that the MCP can recognize regardless of the device on which the medium resides.

VSID

See virtual static imaging device.

W

WFL

See Work Flow Language.

Work Flow Language (WFL)

The Unisys language used for constructing jobs that compile and run programs. WFL includes variables, expressions, and flow-of-control statements that offer the programmer a range of task control capabilities.

working available table

A file maintained by the MCP that records space available on disk or disk pack. There is one table for all disks. Each disk pack maintains its own table.

writeahead

In DMSII, a feature that causes buffers to be written to disk pack ahead of the point where the expected write is necessary.

X

X.25

An international, standard packet-switching network interface, designed to allow different computers to communicate.

Z

ZIP

A facility that allows user programs to send system commands. The commands that are sent with the ZIP facility are handled as if they had been entered at the operator display terminal (ODT). The ZIP facility is invoked by constructs in the programming languages.

ZIPSPO

A facility that gives a user program the ability to pass control information to the MCP and to determine if any errors occurred during the processing. If there is an error, a response will be displayed and returned to the program.

Index

A

- abnormal failures
 - use of DMPMEM for, III: 8-1
- ABort command, I: 4-2
- AC keyword
 - with DISPKV program, III: 3-7
- AC (Purge And Place Tape Number In Label), II: 2-2
- access code
 - description of, III: 3-7
 - list of, III: 3-7
 - USER (Assign Access Code), II: 2-401
 - WRU (Display ODT Access Code), II: 2-422
- access level restrictions, II: 1-12
- access to system for emergency shutdown, I: 2-7
- action commands, I: 4-2
- action messages
 - definition, IV: 1-2
 - output messages, IV: D-1
- ACTIVE, I: 4-9
- active mix, I: 5-71, II: 2-8
- AD keyword
 - with DISPKV program, III: 3-10
- AD specification, I: 5-68
 - commands, II: 2-4
- AD (Specify ODT/OCS Screen Parameters), II: 2-4
- ADD command, I: 4-2
- ADD (Copy Files from Library Tape to Disk or Diskpack), II: 2-11
- address
 - description of, III: 3-10
- address breakpoints
 - Debug Facility
 - actions for setting, III: 28-19
- Address field, I: 4-59
- AFTER (Hold Program), II: 2-12
- after-linkage, II: 2-312
- AFTR
 - option, I: 4-47, II: 2-352
- AFTR (cont.)
 - record, I: 5-57
- algebraic operators
 - list of, III: 5-8
- ALLOCATE (Create Disk Space And Directory Entries), II: 2-17
- ALLOCATE record, I: 5-2
- Allocate screen, I: 4-54
- ALTER (Change System Parameters), II: 2-21
 - BACKUP, II: 2-22
 - EXCHANGE, II: 2-24
 - FAMILY, II: 2-25
 - FILE, II: 2-28
 - PACK, II: 2-29
 - PROC, II: 2-31
 - SYSTEM, II: 2-33
 - used during system load, IV: 2-2
 - TRAKBUFFER, II: 2-34
 - VCS, II: 2-35
- alternate channel, I: A-6
 - use of, III: 3-6
- alternative program
 - to initialize
 - disk pack types, III: 3-31
- AM table, I: 5-70
- APBD
 - option, I: 4-47, II: 2-352
 - record, I: 5-57
- APCR
 - option, I: 4-47, II: 2-352
 - option
 - record, I: 5-57
 - SD (Set Deck Limit), II: 2-342
 - with LDCNTL program, III: 10-6
- area, III: 31-8
 - file, I: 5-3
- AREA field, I: 4-59
- ASCII files
 - automatic translation of, III: 5-2
- AT (Initiate BNA Job Transfer), II: 2-36
- AT (Pass Message to Remote Host), II: 2-38

Index

AUHL
option, I: 4-47, II: 2-352
record, I: 5-58
AURD
option, I: 4-47, II: 2-352
record, I: 5-58
AUTO KX, II: 2-211
option, I: 5-48
AUTO option, I: 4-47
auto process, I: 4-53
Automatic System Recovery Facility
(SYSUP)
description of, III: 25-1
initiating, III: 25-1
available table, III: 31-12
used to omit bad sectors, III: 19-1
AX command
END required, III: 6-3
starting DMPANL with, III: 6-2
with DISPKV program, III: 3-4
with DMPALL program, III: 5-1
with MDCOPV program, III: 14-2
with PBDPRN program, III: 17-7
with PKCOPY program, III: 19-2
with UNLODV program, III: 26-4
AX (Pre-Respond to an
ACCEPT), II: 2-39
AX (Respond to an ACCEPT), II: 2-41

B
backup files, I: 4-41, 5-6
ALTER BACKUP, II: 2-22
BF, BFP (List Backup Files on Disk or
Pack), II: 2-47
BFRF (Search, List, Remove Backup
Files), II: 2-49
CD,CDP (List Inactive Pseudo
Decks), II: 2-68
CV, CVP (Convert Punch Backup File
To Pseudo Card
Deck), II: 2-100
PC, PCP (Punch Backup
File), II: 2-275
RF, RFP (Remove Backup File From
Disk or Pack), II: 2-320
SHOW BACKUP, II: 2-344
SK (Skip Backup Records), II: 2-348
Badmem screen, I: 4-62
base pack, III: 31-13, 31-19

BCL (Mark Beginning Of Card Data
File), II: 2-43
BEGINUSER (Mark Beginning of Job
Stream), II: 2-45
BF, BFP (List Backup Files on Disk or
Pack), II: 2-47
BFRF intrinsic, II: 2-55
BFRF (Search, List, Remove Backup
Files), II: 2-49
BINARY (Read Binary Card
Deck), II: 2-57
block, III: 31-8
block lockout table (BLT)
entries, I: 4-51
message when full, IV: D-1
SHOW BLT, II: 2-344
blocking factor, I: 4-44, III: 31-8
BLT
option, I: 4-51
record, I: 5-29
SHOW command, II: 2-344
BNA
AT (Initiate BNA Job
Transfer), II: 2-36
AT (Pass Message to Remote
Host), II: 2-38
CA (List Port File
Candidates), II: 2-65
ENDAT (End Job Transfer), II: 2-157
entering commands through, II: 1-7
HN (Hostname), II: 2-189
NW (Networking Command
Prefix), II: 2-244
option, I: 5-9
time-out period, I: 4-51, 5-26
BNAHDL program, I: 4-51, 5-26
BNAODT program, I: 4-51, 5-26
BOJ
option, I: 4-47, II: 2-352
record, I: 5-58
Breakpoint Menu
Debug Facility
description of, III: 28-17
list of actions, III: 28-17
B 874 DCP, I: A-3
host load for, III: 11-1
B 874 Link Errors
chart of, III: 27-3
B 9352, I: A-3
B 974 DCP, I: A-3
host load for, III: 11-1

- B 974 Link Errors
 - chart of, III: 27-4
 - B 974LD program
 - analyzing MLOG files, III: 1-3
 - description of, III: 1-1
 - features
 - overview, III: 1-1
 - functions
 - list of, III: 1-1
 - initiating, III: 1-1
 - NDL compilation, III: 1-3
 - tasks
 - list of, III: 1-2
 - uploading dump files, III: 1-2
 - uploading printer backup files, III: 1-3
 - BTOS errors, I: 2-4
 - buffer memory allocation, I: 4-43
 - buffered printer, I: 5-50
 - adding, II: 2-393
 - BYPASS
 - BOTH option, I: 3-12
 - DISK option, I: 3-12
 - OVER option, I: 3-12
 - PACK option, I: 3-12
 - RELOAD option, I: 3-12
- C**
- CA (List Port File Candidates), II: 2-65
 - cache
 - using QWIK Disk
 - with disk pack-based system, III: 29-11
 - with disk-based system, III: 29-12
 - CANDE
 - entering commands through, II: 1-6
 - QT (Quit Program Operation), II: 2-308
 - CARD, I: 4-9
 - card
 - BCL (Mark Beginning Of Card Data File), II: 2-43
 - BINARY (Read Binary Card Deck), II: 2-57
 - CV, CVP (Convert Punch Backup File To Pseudo Card Deck), II: 2-100
 - DATA/DATAB (Mark Beginning Of Card Data File), II: 2-105
 - card (cont.)
 - EBCDIC (Mark Beginning of Card Date File), II: 2-153
 - END/ENDCTL (End of Card File or Control Instructions), II: 2-156
 - entering commands through, II: 1-5
 - PC, PCP (Punch Backup File), II: 2-275
 - SPO (Simulate SPO Through Card Reader), II: 2-357
 - card format
 - control instructions for, III: 10-5
 - card images
 - maximum number permitted, III: 10-6
 - card punch, I: 5-49, A-1
 - adding, II: 2-385
 - configuration, I: 4-21
 - card reader, I: A-1
 - adding, II: 2-385
 - commands invalid from, II: 1-11
 - conditions for use
 - list of, III: 10-3
 - configuration, I: 4-21
 - catastrophic overvoltage, I: 1-6
 - CC (Enter Control Instructions), II: 2-66
 - <cc/u>
 - output message abbreviation, IV: C-1
 - @<cc/u>
 - output message abbreviation, IV: E-1
 - CD, CDP (List Inactive Pseudo Decks), II: 2-68
 - centerpoint priority, I: 4-43, A-13
 - central system
 - maintenance processor, I: 1-3
 - memory, I: 1-2
 - processor, I: 1-2
 - V 300, I: 1-1
 - V 500, I: 1-2
 - CG field, I: 4-61
 - CHANGE (Change File ID), II: 2-70
 - CHange command, I: 4-2
 - channel, I: A-2
 - add to exchange, II: 2-24
 - alternate, I: A-6
 - DBIC (Initialize Controller Parameters), II: 2-111
 - declaring a, I: 5-21
 - for NST (nonstatus) devices, I: 3-26
 - primary, I: A-6
 - remove from exchange, II: 2-24
 - show all on exchange, II: 2-344

Index

- channel (cont.)
 - XC (Inhibit/Uninhibit or Display Status of a Channel), II: 2-434
- channel number
 - MCP, I: ix, 3-20
 - MMDD, I: ix
 - MP, I: x
 - OCS, I: x
 - ODT, I: x
- Channel Priority screen, I: 4-19
- characters
 - old and new, III: 13-3
- CHARGE (Insert Charge Number), II: 2-74
- charge number, II: 2-422
 - field, I: 4-61
- CHECK (Verify Library Tape), II: 2-76
- CHRG
 - option, II: 2-352
 - record, I: 5-58
- CIO instruction
 - failure during system load, IV: 2-1
- CK (Test Peripheral Device), II: 2-77
- CLEAR command, I: 3-34
- CLEAR MEM command, I: 3-37
- clear memory
 - MCP command used during system load failure, IV: 2-1
- clear memory procedure errors
 - procedure for V 500 Series, IV: 3-14
- CLOCK command, I: 3-44
- CLOS
 - option, I: 4-47, II: 2-352
 - record, I: 5-58
- CN (Display Tape Number), II: 2-78
- CO keyword
 - for DISPKV program, III: 3-5
- COBOL
 - MERGE statement, III: 15-1
- code file overlays
 - with QWIKPOOL option in QWIK Disk, III: 29-4
- code file transfer
 - from 100-byte disk, III: 18-1
 - from 180-byte disk pack, III: 18-1
- code files, I: 5-7
- coded halts
 - during system initialization, IV: 2-1
- cold-start, I: A-2
 - failure
 - procedure for V 300/V 400 Series, IV: 3-9
- cold-start (cont.)
 - multisystem shared
 - configuration, I: 3-27
 - options, I: 3-12
 - BYPASS BOTH, I: 3-12
 - BYPASS DISK, I: 3-12
 - BYPASS PACK, I: 3-12
 - CONVERSE, I: 3-12
 - RELOAD, I: 3-12
 - RELOAD OVER, I: 3-12
 - options for special
 - circumstances, I: 3-12
 - V 300, I: 3-9
 - V 500, I: 3-25
 - multiple processor, I: 3-29
- Cold-Start screen, I: 3-10
- collating sequence
 - ASCENDING
 - discussion of, III: 21-3
 - changes to, III: 13-2
 - DESCENDING
 - discussion of, III: 21-3
- command restrictions, II: 1-11
- comments
 - in SRTUTL syntax, III: 23-1
- communication processor
 - host booting of, III: 2-2
 - types
 - list of, III: 2-1
- COMPARE (Compare Files On Different Media), II: 2-79
- COMPILE (Compile Program), II: 2-80
- COMPILE operation
 - with LDCNTL program, III: 10-8
- compiling with COBOL
 - for SRTUTL, III: 23-1
- compiling with RPG
 - for SRTUTL, III: 23-1
- concurrent DEBUG sessions
 - commands for
 - description of, III: 28-2
- concurrent file access
 - diagram of, III: 30-2
- CONFIG DTM screen, I: 3-16
- CONFIG utility, I: 3-13, 4-1
 - action commands, I: 4-2
 - ACTIVE, I: 4-9
 - Allocate screen, I: 4-54

- CONFIG utility (cont.)
- Badmem screen, I: 4-62
 - CARD, I: 4-9
 - Channel Priority screen, I: 4-19
 - Control Options screen, I: 4-40
 - Data Comm screen, I: 4-31
 - DCP screen, I: 4-27
 - declaring disk pack units, I: 4-13
 - declaring disk units, I: 4-13
 - Define a Disk/Pack Configuration screen, I: 4-12
 - Define a Tape Configuration screen, I: 4-17
 - Define a Unit Configuration screen, I: 4-21
 - Define DSK DLP Firmware-IDs and Inhibits screen, I: 4-15
 - Define Logical Subsystems for 100-Byte Media screen, I: 4-16
 - Disk Sectioning (MOREDISK) screen, I: 4-39
 - error messages, I: B-1
 - executing on a V 300 system, I: 4-5
 - executing on a V 500 system, I: 4-6
 - File Specification screen, I: 4-8
 - help screens, I: 4-3
 - Inactive DLP screen, I: 4-38
 - input file, I: 4-9
 - ISC screen, I: 4-34
 - Limit Options Value Override screen, I: 4-43
 - loading on a V 300 system, I: 4-5
 - loading on a V 500 system, I: 4-6
 - Logging and Dump Options screen, I: 4-52
 - Master Selection menu, I: 4-11
 - MCP Limits screen, I: 4-50
 - menus, I: 4-7
 - NCP screen, I: 4-29
 - NST screen, I: 4-33
 - OCS; ODT screen, I: 4-25
 - Other Devices menu, I: 4-20
 - output file, I: 4-9
 - Preallocated Disk Space screen, I: 4-58
 - Printer screen, I: 4-23
 - Redirect Output screen, I: 4-10
 - screens, I: 4-7
 - Security, Hostname, Connect, Label screen, I: 4-60
 - SSP screen, I: 4-36
 - Unit Selection menu, I: 4-20
- CONFIG utility (cont.)
- USE Options screen, I: 4-46
 - configuration file, I: 4-1
 - loading the, I: 1-12
 - source, I: x
 - configuration file errors
 - CONNECT record, IV: B-2, B-3, B-5
 - CONTROL LOG SUBSYSTEM record, IV: B-4
 - CONTROL STOP SUBSYSTEM record, IV: B-4
 - EXCHANGE record, IV: B-2, B-5
 - LIMIT record, IV: B-3
 - PROCESSOR record, IV: B-2
 - UNIT OCS record, IV: B-3
 - USE DUMP record, IV: B-2
 - configuration record errors
 - output messages, IV: 2-4
 - responses to, IV: 2-4
 - configuration records, I: 5-1
 - configuration rules
 - in multisystem shared environment
 - discussion of, III: 30-4
 - CONFIGURE command, I: 3-34
 - error summary report for, III: 3-17
 - CONNECT record, I: 3-27, 5-5
 - configuration file errors, IV: B-2, B-3, B-5
 - Console DLP, I: 5-15, 5-75
 - continuation pack, III: 31-13, 31-19
 - CONTINUE command, I: 3-34
 - CONTROL BACKUP record, I: A-12
 - control deck
 - requirements for, III: 10-4
 - special cards
 - ?DATA CTLDCK, III: 10-4
 - ?ENDCTL, III: 10-4
 - control file
 - creating
 - for SORT: intrinsic program, III: 22-6
 - control instructions
 - erroneous combination of, III: 10-5
 - CONTROL LOG SUBSYSTEM record
 - configuration file errors, IV: B-4
 - Control Options screen, I: 4-40
 - control panel
 - IIO cabinet, I: 1-10
 - V 300, I: 1-5
 - expansion cabinet, I: 1-7
 - V 500, I: 1-8
 - IIO cabinet, I: 1-10

Index

- CONTROL record, I: 5-6
 - BACKUP, I: 5-6
 - CODEPACK, I: 5-7
 - DEBUG MCP, I: 5-7
 - DEBUG USER, I: 5-7
 - LOGSUBSYS, I: 5-7
 - LOGXFRTIME, I: 5-7
 - PRIORITY, I: 5-8
 - STOPSUBSYS, I: 5-10
 - VCS INITFILE, I: 5-10
- CONTROL STOP SUBSYSTEM record
 - configuration file errors, IV: B-4
- control store, I: 1-2
 - errors
 - errors
 - loading the, I: 1-12
 - procedure for V 300/V 400 Series, IV: 3-3
 - procedure for V 500 Series, IV: 3-13
 - reloading the (V 500), I: 3-18
- controller, I: A-2, A-3
- controllers supported
 - by DISPKV program, III: 3-32
- conventional halt/load
 - multisystem shared
 - configuration, I: 3-31
 - V 300, I: 3-30
 - V 500, I: 3-30
- Conversation Halt/Load screen, I: 3-33
- conversational halt/load, I: 3-32
 - CLEAR command, I: 3-34
 - commands, I: 3-34
 - CONFIGURE command, I: 3-34
 - CONTINUE command, I: 3-34
 - EDIT command, I: 3-34
- CONVERSE option, I: 3-12
- conversion of
 - interlaced to sequential mode, III: 3-15
 - sequential to interlaced mode, III: 3-15
 - 100-byte to 180-byte disk packs, III: 3-15
 - 180-byte to 100-byte disk packs, III: 3-15
- converting files
 - with DMPALL program
 - restrictions, III: 5-11
 - syntax for, III: 5-11
- COPY
 - option, I: 4-47, II: 2-352
 - record, I: 5-59
 - copy files
 - to floppy disk, III: 14-4
 - COPY (Invoke File Transfer Utility), II: 2-86
 - COPYADD (Transfer Files), II: 2-99
 - copying files
 - from floppy disk
 - with MDCOPV program, III: 14-4
 - CPLOAD functions
 - for communication
 - processors, III: 2-1
 - CPLOAD utility program
 - description of, III: 2-1
 - dump file
 - naming convention, III: 2-2
 - error messages, III: 2-3
 - functions
 - list of, III: 2-1
 - initiating, III: 2-1
 - crash-out mechanism, I: 4-48
 - creating files
 - in QWIK Disk, III: 29-32
 - programmatically
 - in QWIK Disk, III: 29-32
 - CTLDCCK, II: 2-43
 - contents, III: 10-5
 - CU keyword
 - for DISPKV program, III: 3-6
 - CV command
 - with LDCNTL program, III: 10-13
 - CV, CVP (Convert Punch Backup File To Pseudo Card Deck), II: 2-100
 - CV33nA minidisk, I: 3-3, 3-8
 - CVP command
 - with LDCNTL program, III: 10-13
 - CY keyword
 - with DISPKV program, III: 3-9
 - cylinder, III: 31-5
 - description of, III: 3-9
 - maximum number
 - chart of, III: 3-9
 - number
- D**
- DA command
 - with LDCNTL program, III: 10-7
- DA, DAP (Deactivate Pseudo Reader), II: 2-103
- DAP command
 - with LDCNTL program, III: 10-7

- DASDL
 use with SRTUTL, III: 23-1
- Data Comm screen, I: 4-31
- data communications
 AT (Initiate BNA Job Transfer), II: 2-36
 AT (Pass Message to Remote Host), II: 2-38
 ENDAT (End Job Transfer), II: 2-157
 HN (Hostname), II: 2-189
 module, I: 4-49
 NW (Networking Command Prefix), II: 2-244
 SI (Set Remote Device Identification), II: 2-347
 SS (Send Message), II: 2-363
 UNIT (Add Peripheral Devices), II: 2-385
- Data Communications Processor, I: A-2, A-3
 BUFFER control instruction
 description of, III: 27-3
 declaring channels for, III: 27-2
 description of, III: 27-1
 error conditions
 chart of, III: 27-3
 firmware files for, III: 27-1
 initializing network for, III: 27-2
 maximum number of output messages, I: 4-51
 MCP interfaces for, III: 27-2
 module, I: 4-49
 system requirements for, III: 27-1
- data file
 unique name requirement, III: 10-12
- data link processor (DLP), I: A-2, A-3
 configuration, I: 4-38
 console, I: 5-15
 Console, I: 5-75
 inactive, I: 4-38
 inhibiting a, I: 5-15
 message for hung DLP, IV: E-2
 Uniline, I: 5-15
- Data Management System II (DMSII), I: 4-43
- data path, I: A-2
- data transfer module (DTM), I: 1-4
- database
 IR (Initiate DMSII Recovery), II: 2-199
 maximum number of, I: 4-44
 maximum number of users, I: 4-44
- database (cont.)
 SB (Interrogate Database Activity), II: 2-340
 SP (Display/Set/Change Database Parameters), II: 2-356
- DATA/DATAB (Mark Beginning Of Card Data File), II: 2-105
- DATE
 option, I: 4-47, II: 2-352
 record, I: 5-59
- DBCA (Clear SSP Address Contention), II: 2-107
- DBCL (Cancel DLP), II: 2-109
- DBDR (Produce SSP Content Report), II: 2-110
- DBIC (Initialize Controller Parameters), II: 2-111
- DBMM (Set/Reset Maintenance Mode), II: 2-112
- DBSR (Produce Block Lockout Table Report), II: 2-114
- DBUA (Unlock Address), II: 2-115
- DEBUG
 module option, I: 4-49
 option, II: 2-353, IV: 5-2
 record, I: 5-59
- DBUL (Clear Locked Files on Shared Systems), II: 2-117
- DC (Display Compilation Data), II: 2-118
- DC1ANL utility program
 with CPLOAD program, III: 2-2
- DCDLP, I: 5-42
- DCOM
 option, I: 4-49, II: 2-353
 record, I: 5-59
- DCP
 adding, II: 2-386
 configuration, I: 4-27
 for Data Communications Processors, III: 27-2
 module, I: A-3
 option, I: 4-49, II: 2-353
 record, I: 5-42, 5-59
 screen, I: 4-27
 UNIT (Add Peripheral Devices), II: 2-385
 UNIT record
- DCPBUF
 option, I: 4-51
 record, I: 5-29

Index

- DCPQUE
 - option, I: 4-51
 - record, I: 5-30
- DD (Delete DLP), II: 2-120
- dead space, III: 31-9
- deallocation
 - of card reader
 - definition of, III: 10-12
 - of pseudo card file
 - definition of, III: 10-12
- DEBUG (Attach Program To Debug Session), II: 2-121
- DEBUG command
 - for Debug Facility
 - description of, III: 28-3
 - examples of, III: 28-5
 - options for, III: 28-3
 - syntax for, III: 28-3
 - parameter list
 - syntax for, III: 28-4
- DEBUG commands
 - DEBUG
 - description of, III: 28-1
 - ENTER Debug (ED)
 - description of, III: 28-2
 - INTERACTIVE DEBUG (ID)
 - description of, III: 28-1
 - QUERY Debug (QD)
 - description of, III: 28-2
- Debug Facility
 - address breakpoints
 - description of, III: 28-19
 - settings for, III: 28-19
 - Breakpoint Menu
 - description of, III: 28-17
 - list of actions, III: 28-17
 - examples
 - debug sessions, III: 28-22
 - functions
 - stop user program, III: 28-22
 - Hypercall/BCT Breakpoints
 - actions for setting, III: 28-18
 - description of, III: 28-18
 - initiating
 - procedures for, III: 28-1
 - list of actions, III: 28-15
 - Main Menu
 - description of, III: 28-11
 - list of actions, III: 28-11
 - opcode breakpoint
 - description of, III: 28-19
 - opcode breakpoint actions
- Debug Facility (cont.)
 - list of, III: 28-19
 - output device options
 - list of, III: 28-16
 - overflow breakpoint
 - description of, III: 28-20
 - PEEK and POKE functions
 - using, III: 28-25
 - PEEK function
 - description of, III: 28-25
 - POKE function
 - description of, III: 28-27
 - State Menu
 - description of, III: 28-20
 - list of actions, III: 28-20
 - state of task
 - menu display, III: 28-9
 - status line
 - example of, III: 28-9
 - Status Menu
 - description of, III: 28-13
 - list of actions, III: 28-13
 - taken branch breakpoint
 - description of, III: 28-20
 - trace functions
 - using, III: 28-28
 - Trace Menu
 - description of, III: 28-15
 - Trace Menu
 - user interface menus
 - list of, III: 28-8
 - user selected opcodes
 - field for, III: 28-17
- DEBUG utility, I: 4-41
- debugger, I: 5-7
- debugging, I: A-18
 - CK (Test Peripheral Device), II: 2-77
 - DBCA (Clear SSP Address Contention), II: 2-107
 - DBCL (Cancel DLP), II: 2-109
 - DBDR (Produce SSP Content Report), II: 2-110
 - DBIC (Initialize Controller Parameters), II: 2-111
 - DBSR (Produce Block Lockout Table Report), II: 2-114
 - DBUA (Unlock Address), II: 2-115
 - DBUL (Clear Locked Files on Shared Systems), II: 2-117
 - DEBUG (Attach Program To Debug Session), II: 2-121

- debugging (cont.)
 - DISPLAY/WHATS (Display Information), II: 2-131
 - DP (Dump and Discontinue), II: 2-144
 - ED (Enter DEBUG Session), II: 2-155
 - ET (End TRAK), II: 2-161
 - GT (Initiate Trace), II: 2-183
 - ID (Invoke DEBUG Session), II: 2-191
 - MEMDUMP (Abnormal EOJ Memory Dump), II: 2-229
 - NT (Terminate Trace), II: 2-243
 - PM (Print Memory Dump), II: 2-284
 - QD (Query Debug), II: 2-307
 - TEST (Mark A Test Operation), II: 2-371
 - TRACE (Trace a Program), II: 2-381
- default disk
 - using QWIK Disk
 - discussion of, III: 29-32
- default media
 - SORT: intrinsic program, III: 22-10
- default media override
 - SORT: intrinsic program
 - description of, III: 22-6
- Define a Disk/Pack Configuration
 - screen, I: 4-12
- Define a Tape Configuration
 - screen, I: 4-17
- Define a Unit Configuration
 - screen, I: 4-21
- Define DSK DLP Firmware-IDs and Inhibits screen, I: 4-15
- Define Logical Subsystems for 100-Byte Media screen, I: 4-16
- DELAY
 - option, I: 4-51
 - record, I: 5-30
- DELeTe command, I: 4-2
- deleting files
 - any file, II: 2-315
 - backup and memory dump files, II: 2-320
 - duplicate files - new, II: 2-234
 - duplicate files - old, II: 2-322
- DFSC option, I: 4-61, 5-36
- diagnosing problems, I: A-18
- DIR (List File Directory), II: 2-125
- direct I/O, I: A-4
- directory, III: 31-10
- disk, I: 5-12, III: 31-1
 - ALLOCATE (Create Disk Space And Directory Entries), II: 2-17
- disk (cont.)
 - allocating space on a, I: 4-54
 - configuration record, I: 5-11
 - declaring a, I: 5-11
 - declaring units, I: 4-13
 - DISK (Add a Disk), II: 2-128
 - DISPLAY/WHATS (Display Information), II: 2-131
 - LAK, III: 31-1
 - Look-Alike, III: 31-1
 - maximum number of, I: 4-44
 - QWIKDISK, III: 31-1
- DISK
 - record, I: 5-11
- disk
 - RX (Return Deleted Space), II: 2-336
 - soft-sectored, III: 31-1
 - SQ (Squash Disk), II: 2-359
 - status table display, I: 5-70
 - XD (Remove Disk Segments), II: 2-436
 - 5N, III: 31-1
- DISK (Add a Disk), II: 2-128
- disk file security, I: 4-61
- disk maintenance
 - ADD (Copy Files from Library Tape to Disk or Diskpack), II: 2-11
 - CHANGE (Change File ID), II: 2-70
 - KA (Analyze Disk Directory), II: 2-201
 - KP (Print Disk Segments), II: 2-205
 - KS (Analyze Disk Space), II: 2-207
 - KX (Obtain User Disk Space), II: 2-210
 - MR (Remove Duplicate File), II: 2-234
 - NOSTANDBY option of KA command, II: 2-203
 - NOSTANDBY option of KS command, II: 2-208
 - NOSTANDBY option of PA command, II: 2-260
 - NOSTANDBY option of PS command, II: 2-304
 - PATCH (Patch File), II: 2-264
 - REMOVE (Delete Files), II: 2-315
 - RF, RFP (Remove Backup File From Disk or Pack), II: 2-320
 - RM (Remove Duplicate File), II: 2-322
- disk pack
 - ALTER PACK, II: 2-29
 - configuration, I: 4-12
 - controller firmware, I: xi

- disk pack (cont.)
 - DBMM (Set/Reset Maintenance Mode), II: 2-112
 - declaring a, I: 5-11
 - declaring units, I: 4-13
 - ID number, I: 5-32
 - initializing and verifying a
 - LAK, I: 3-24
 - maximum number of, I: 4-44
 - PA (Analyze Pack Directory), II: 2-259
 - PACK (Add Pack), II: 2-262
 - PO (Power Off Diskpack), II: 2-286
 - PP (Print Pack Sectors), II: 2-287
 - primary family, I: 4-41
 - PS (Analyze Pack Space), II: 2-303
 - RX (Return Deleted Space), II: 2-336
 - secondary family, I: 4-41
 - SQP (Squash Pack), II: 2-360
 - status table, I: 5-70
 - XP (Remove Pack Sectors), II: 2-438
 - 206 type, I: 3-24
 - 207 type, I: 3-24
- disk pack maintenance
 - CHANGE (Change File ID), II: 2-70
 - MR (Remove Duplicate File) -
 - new, II: 2-234
 - REMOVE (Delete Files), II: 2-315
 - RF, RFP (Remove Backup File From Disk or Pack), II: 2-320
 - RM (Remove Duplicate File) -
 - old, II: 2-322
- disk pack name
 - PN keyword, III: 3-8
- disk pack serial numbers
 - range of, III: 19-3
 - when to change, III: 19-1
- disk pack-based system
 - using QWIK Disk
 - as a cache, III: 29-11
- Disk Sectioning (MOREDISK)
 - screen, I: 4-39
- disk sorting
 - operation of, III: 22-5
 - SORT: intrinsic program
 - control parameters for, III: 22-5
 - default media for, III: 22-5
 - media used for, III: 22-5
 - operating considerations
 - for, III: 22-6
 - resource allocation for, III: 22-6
 - work file requirements, III: 22-5
- disk timeout
 - retrying I/O, IV: E-2
- disk-based system
 - using QWIK Disk
 - as a cache, III: 29-12
- diskette load failure
 - procedure for V 300/V 400 Series, IV: 3-2
- DISKPACKS record, I: 5-25
- DISPKV commands
 - CONFIGURE, III: 3-17
 - INITIALIZE, III: 3-14
 - LABEL, III: 3-27
 - RECONFIGURE, III: 3-25
 - RECONFIGUREL, III: 3-26
 - RELOCATE, III: 3-22
 - RENAME, III: 3-24
 - REPORT, III: 3-19
 - SINGLE, III: 3-20
 - 659IVR, III: 3-28
- DISPKV initialization
 - for disks and disk packs, III: 3-1
- DISPKV keywords
 - AC, III: 3-7
 - AD, III: 3-10
 - chart of, III: 3-5
 - CO, III: 3-5
 - CU, III: 3-6
 - CY, III: 3-9
 - ER, III: 3-9
 - FILES, III: 3-12
 - OI, III: 3-8
 - PN, III: 3-8
 - SC, III: 3-8
 - SN, III: 3-6
- DISPKV program input
 - with AX command, III: 3-2
- DISPKV program messages
 - types
 - list of, III: 3-37
- DISPKV utility program
 - available table, III: 3-14
 - card input
 - rules for, III: 3-36
 - command functions
 - list of, III: 3-29
 - CONFIGURE command, III: 3-17
 - controllers supported, III: 3-32
 - description of, III: 3-1
 - disk pack label, III: 3-14
 - empty disk pack directory
 - table, III: 3-14

- DISPKV utility program (cont.)
- END option, III: 3-13
 - error messages, III: 3-40
 - examples, III: 3-16
 - executing, III: 3-2
 - file names
 - external, III: 3-34
 - internal, III: 3-34
 - functions
 - list of, III: 3-1
 - informational messages, III: 3-37
 - INITIALIZE command, III: 3-14
 - initializing disk packs
 - to 100-byte mode, III: 3-43
 - input examples, III: 3-35
 - LABEL command, III: 3-27
 - LAK disk pack functions
 - Initialize, III: 12-1
 - Relocate, III: 12-1
 - Verify, III: 12-1
 - master available table, III: 3-14
 - operator prompts, III: 3-38
 - RECONFIGURE command, III: 3-25
 - RECONFIGUREL command, III: 3-26
 - RELOCATE command, III: 3-22
 - RENAME command, III: 3-24
 - REPORT command, III: 3-19
 - SINGLE command, III: 3-20
 - value statements
 - use of, III: 3-3
 - 659IVR command, III: 3-28
- display information
- AD (Specify ODT/OCS Screen Parameters), II: 2-4
 - BF, BFP (List Backup Files on Disk or Pack), II: 2-47
 - BFRF (Search, List, Remove Backup Files), II: 2-49
 - CA (List Port File Candidates), II: 2-65
 - CD,CDP (List Inactive Pseudo Decks), II: 2-68
 - CK (Test Peripheral Device), II: 2-77
 - CN (Display Tape Number), II: 2-78
 - DBDR (Produce SSP Content Report), II: 2-110
 - DBSR (Produce Block Lockout Table Report), II: 2-114
 - DC (Display Compilation Data), II: 2-118
 - DIR (List File Directory), II: 2-125
 - display information (cont.)
 - DISPLAY/WHATS (Display Information), II: 2-131
 - DQ (Display System Messages), II: 2-145
 - DX (Display DLPs Attached To An Exchange), II: 2-151
 - ES (End Screen), II: 2-160
 - FI (List Files in Use), II: 2-166
 - FN (List File Names), II: 2-177
 - HN (Hostname), II: 2-189
 - KA (Analyze Disk Directory), II: 2-201
 - KS (Analyze Disk Space), II: 2-207
 - KX (Obtain User Disk Space), II: 2-210
 - OL (Display Peripheral Status), II: 2-248
 - OT (Display Program Data), II: 2-256
 - PA (Analyze Pack Directory), II: 2-259
 - PD (Print Directory), II: 2-278
 - PS (Analyze Pack Space), II: 2-303
 - QD (Query Debug), II: 2-307
 - SB (Interrogate Database Activity), II: 2-340
 - SHOW (Display System Information), II: 2-343
 - TI (Display Time Statistics), II: 2-372
 - TO (Type System Options), II: 2-379
 - WB (Display Bound Programs), II: 2-405
 - WC (Display Available Memory), II: 2-406
 - WD (Display Date), II: 2-408
 - WM (Display MCP Version), II: 2-416
 - WO (Display Active User QWIK Operation Counts), II: 2-418
 - WQ (Display STOQUE Count), II: 2-420
 - WS (Display Jobs in Schedule), II: 2-423
 - WT (Display Time), II: 2-424
 - XC (Inhibit/Uninhibit or Display Status of a Channel), II: 2-434
 - DISPLAY/WHATS (Display Information), II: 2-131
 - DL (Delete Peripheral Devices), II: 2-135
 - DLP
 - contents of, III: 4-1
 - DBCL (Cancel DLP), II: 2-109

Index

- DLP (cont.)
 - DBIC (Initialize Controller Parameters), II: 2-111
 - DD (Delete DLP), II: 2-120
 - DLP (Add DLPs), II: 2-137
 - dump print file
 - DX (Display DLPs Attached To An Exchange), II: 2-151
 - for reloading firmware, III: 4-2
 - record, I: 5-15, A-13
 - requirements
 - screen display
 - SSP record, I: 5-19
- DLP (Add DLPs), II: 2-137
- DLP record
 - for Data Communications Processors, III: 27-2
- DLPXCO and DLPXNO utility programs
 - description of, III: 4-1
 - dump information
 - how produced, III: 4-1
 - error messages for, III: 4-5
- DLPXCO utility program
 - executing
 - example of, III: 4-3
 - procedures for, III: 4-2
 - use with V 300 system, III: 4-1
- DLPXNO utility program
 - executing
 - example of, III: 4-5
 - procedures for, III: 4-4
 - use with MCP, III: 4-1
- DM command
 - with DMPANL program, III: 6-11
 - with DMPCPY program, III: 7-1
- DMPALL card decks
 - examples
 - for SORTx intrinsic program, III: 21-6
- DMPALL commands
 - list of, III: 5-1
- DMPALL media conversion
 - options
 - list of, III: 5-14
- DMPALL parity error
 - examples of messages, III: 5-25
 - message for, III: 5-25
- DMPALL utility program, I: 4-42, II: 2-180, 2-308
 - description of, III: 5-1
 - examples of, III: 5-27
- DMPALL utility program (cont.)
 - functions
 - list of, III: 5-1
 - miscellaneous, III: 5-21
 - initiating, III: 5-1
 - options
 - list of, III: 5-2
 - PERFORM LIBLST function
 - example of, III: 5-21
 - PERFORM PD function
 - example of, III: 5-23
 - syntax for, III: 5-22
 - PERFORM (Schedule DMPALL), II: 2-280
 - PFM LIST DSK function
 - example of, III: 5-22
 - PFM PB67 function
 - syntax for, III: 5-22
 - SEARCH function
 - example of, III: 5-22
 - syntax for, III: 5-22
 - use to modify
 - default parameters, III: 21-5
 - use with SORT:
 - to create control file, III: 22-6
 - ZIP mechanism, III: 5-23
- DMPANL command syntax
 - used to limit
 - information to print, III: 6-2
- DMPANL parameters
 - list of
 - syntax for, III: 6-3
- DMPANL utility program, I: 4-42, II: 2-284, 2-308
 - description of, III: 6-1
 - keywords
 - list of, III: 6-3
 - parameters
 - list of, III: 6-4
 - syntax for, III: 6-4
 - raw memory selection options
 - list of, III: 6-9
 - starting with PM 1 command, III: 6-1
 - syntax errors, III: 6-3
 - table list options
 - list of, III: 6-5
 - table selection options
 - list of, III: 6-4
 - task parameters
 - list of, III: 6-10
- DMPCPY utility program
 - description of, III: 7-1

- DMPCPY utility program (cont.)
 executing
 procedure for, III: 7-1
 input error
 message for, III: 7-2
 procedure for handling, III: 7-2
 DMPMEM utility, I: 4-53
 DMPMEM utility program
 description of, III: 8-1
 executing from V 300 system
 procedure for, III: 8-1
 executing from V 500 system
 procedure for, III: 8-2
 terminating with
 halt branch instruction, III: 8-3
 DMPOUT utility program, I: 4-42,
 II: 2-284, 2-308
 DMS2
 extension module option, I: 4-49
 option, II: 2-353
 record, I: 5-59
 DMSDBP
 option, I: 4-44
 record, I: 5-25
 DMSII performance
 with QWIK Disk, III: 29-4
 DMSTATUS
 option, I: 4-44
 record, I: 5-25
 DMSUSERS
 option, I: 4-44
 record, I: 5-25
 dollar sign (\$) cards
 list of
 for SORT: intrinsic
 program, III: 22-7
 for SORTx intrinsic
 program, III: 21-6
 restrictions
 for SORT: intrinsic
 program, III: 22-8
 dollar sign (\$) options
 in SRTUTL utility program
 list of, III: 23-17
 DP command
 with DMPANL program, III: 6-11
 DP (Dump and Discontinue), II: 2-144
 DQ (Display System Messages), II: 2-145
 DR (Date Reset), II: 2-147
 DS command
 with SORTx intrinsic
 program, III: 21-5
 DS (Discontinue Program), II: 2-148
 DSKOUT
 NOSTANDBY option of KA
 command, II: 2-203
 NOSTANDBY option of KS
 command, II: 2-208
 NOSTANDBY option of PA
 command, II: 2-260
 NOSTANDBY option of PS
 command, II: 2-304
 DSKOUT utility program, I: 4-42,
 II: 2-201, 2-205, 2-207, 2-287,
 2-303, 2-308
 dump
 analysis
 DP (Dump and Discontinue), II: 2-144
 DUMP (Copy Files To Library
 Tape), II: 2-150
 files, II: B-5
 MEMDUMP (Abnormal EOJ Memory
 Dump), II: 2-229
 memory, I: A-6
 PM (Print Memory Dump), II: 2-284
 requirements for, III: 6-1
 DUMP (Copy Files To Library
 Tape), II: 2-150
 DUMP record, I: 5-60
 DUP LIB, II: 2-322
 duplicate file, I: 4-48, A-7, II: 2-234,
 2-322
 DX (Display DLPs Attached To An
 Exchange), II: 2-151
 dynamic environment
 file characteristics
 for QWIK Disk analysis, III: 29-16
E
 EBCDIC (Mark Beginning of Card Date
 File), II: 2-153
 ECM command, I: 3-44
 ECM Command menu, I: 2-4
 ECMANL utility program
 description of, III: 9-1
 functions
 list of, III: 9-1
 ED (Enter DEBUG Session), II: 2-155
 Edit Active Configuration File
 screen, I: 3-35
 EDIT command, I: 3-34
 ELSE clause, I: 5-7

Index

- emergency procedures, I: 2-7
- End command, I: 4-2
- END option
 - DISPKV program, III: 3-13
- ENDAT (End Job Transfer), II: 2-157
- END/ENDCTL (End of Card File or Control Instructions), II: 2-156
- end-of-file condition
 - detection of, III: 10-12
- ENDUSER (Mark End Of Job Stream), II: 2-158
- ENTER DEBUG command
 - for Debug facility
 - description of, III: 28-7
 - example of, III: 28-7
 - syntax for, III: 28-7
- environment item
 - input/output processor, I: ix
 - MCP channel and unit numbers, I: ix
 - MCP memory limit, I: ix
 - memory cards, I: ix
 - QWIK Disk base, I: ix
 - QWIK Disk sectors, I: ix
- environmental control module (ECM), IV: 3-11
 - screen
 - used during power-on failure, IV: 3-11, 3-12
- EOF, I: 5-3
 - pointer, I: 4-56
- EOJ
 - option, I: 4-47, II: 2-353
 - record, I: 5-60
- ER keyword
 - with DISPKV program, III: 3-9
- error address
 - description of, III: 3-9
- ERRor command, I: 4-2
- error conditions
 - Data Communications Processors chart of, III: 27-3
- error messages
 - BTOS, I: 2-4
 - CONFIG utility, I: B-1
 - DISPKV program, III: 3-40
 - for debug commands
 - list of, III: 28-32
 - for DLPXCO and DLPXNO programs, III: 4-5
 - for LOADFW program, III: 12-5
 - for SORT: intrinsic program, III: 22-11
- error messages (cont.)
 - from DLP programs
 - discussion of, III: 4-1
 - non-fatal
 - SORT: intrinsic program, III: 22-11
 - parser, I: B-2
 - recovery of
 - for SORT: intrinsic program, III: 22-11
 - SORT: intrinsic program
 - list of, III: 22-11
 - SYSTEM/COPY utility program
 - list of, III: 24-4
- error recovery
 - SYSTEM/COPY utility program, III: 24-10
- ERROR stop cause
 - recovery procedure
 - for V 500 Series, IV: 4-31
- Errors
 - debug session
 - list of, III: 28-29
- ES (End Screen), II: 2-160
- ET (End TRAK), II: 2-161
- EVENT command, I: 3-44
- EVENT STOP
 - recovery procedure
 - for V 300/V 400 Series, IV: 4-15
- Examples
 - DEBUG command
 - for Debug Facility, III: 28-5
 - Debug Facility
 - status line, III: 28-9
 - debug session, III: 28-22
 - DISPKV input, III: 3-35
 - results, III: 3-36
 - DISPKV utility program, III: 3-16
 - ENTER DEBUG command
 - for Debug facility, III: 28-7
 - EXECUTE command
 - starting DMPALL with, III: 5-24
 - file equating
 - in SRTUTL utility program, III: 23-16
 - for DMPALL utility program, III: 5-27
 - INTERACTIVE DEBUG command
 - for Debug Facility, III: 28-6
 - MDCOPV utility program, III: 14-14
 - switch setting, III: 14-3
 - NIFMRG utility program
 - DCP statement, III: 16-4

Examples (cont.)

END statement, III: 16-5
 MERGED statement, III: 16-4
 statements longer than one
 line, III: 16-6
 of DLPXCO utility program, III: 4-3
 of DLPXNO utility program, III: 4-5
 of messages
 for DMPALL parity
 errors, III: 5-25
 of SYSUP facility, III: 25-2
 PBDPRN utility program, III: 17-6
 PCOPY utility program, III: 18-2
 PERFORM LIBLST function
 DMPALL utility program, III: 5-21
 PFM LIST DSK function
 DMPALL utility program, III: 5-22
 PKCOPY utility program, III: 19-3
 shared file operations
 stalemate, III: 30-9
 shared systems
 multisystem shared disk and disk
 pack, III: 30-13
 SORTx intrinsic program
 DMPALL card decks, III: 21-6
 SRTUTL utility program, III: 23-5
 cardless execution, III: 23-15
 executing, III: 23-12
 KEY statement, III: 23-7
 SYSTEM/COPY utility program
 with QWIK Disk, III: 29-31
 typical pseudo card files, III: 10-9
 value clauses, III: 5-24, 5-25
 ZIP mechanism, III: 5-23
 exchange, I: A-2
 EXCHANGE record, I: 5-21
 configuration file errors, IV: B-2, B-5
 EXECUTE command
 starting DMPALL with
 description of, III: 5-23
 example of, III: 5-24
 procedure for, III: 5-23
 with MDCOPV program, III: 14-2
 with PBDPRN program, III: 17-7
 with PKCOPY program, III: 19-2
 with UNLODV program, III: 26-4
 EXECUTE (Execute Library
 Program), II: 2-162
 expansion cabinet, V 300 control
 panel, I: 1-7

F

fail codes
 defined, IV: 5-1
 failure codes, IV: F-1
 family name, I: A-5, III: 31-14
 fatal error messages
 correctable
 for SORT: intrinsic
 program, III: 22-12
 non-correctable
 for SORT: intrinsic
 program, III: 22-12
 fault indicators
 debug session
 list of, III: 28-29
 fault screen, I: 4-48, IV: 5-1
 recovery procedure
 for V 300/V 400 Series, IV: 4-18
 for V 500 Series, IV: 4-36
 representation of, IV: 5-2
 FEP (*See* front-end processor)
 configuration, I: 4-32
 declaration, I: 5-54
 <f-id>
 output message abbreviation, IV: C-1
 @<f-id>
 output message abbreviation, IV: E-1
 FI (List Files in Use), II: 2-166
 FIBMRL
 definition of, III: 22-3
 for SORTx intrinsic
 program, III: 21-5
 discussion of
 for SORTx intrinsic
 program, III: 21-5
 FIBVAR
 definition of
 for SORTx intrinsic
 program, III: 21-5
 field, III: 31-8
 file, III: 31-8
 copy with LOADMP, II: B-2
 copy with PACKUP, II: B-2
 dump, II: B-5
 verify library tape, II: B-4
 file area, I: 5-3
 file assignment
 description of, III: 10-12

Index

- FILE (Equate File-Name With File-Identifier Or Medium), II: 2-167
- file equating
 - for QWIK Disk
 - syntax for, III: 29-33
 - in SRTUTL utility program
 - examples of, III: 23-16
- file handling
 - ALLOCATE (Create Disk Space And Directory Entries), II: 2-17
 - ALTER FILE changes LIBMAINT and SQUASH status, II: 2-28
 - BFRF (Search, List, Remove Backup Files), II: 2-49
 - CHANGE (Change File ID), II: 2-70
 - DATA/DATAB (Mark Beginning Of Card Data File), II: 2-105
 - FILE (Equate File-Name With File-Identifier Or Medium), II: 2-167
 - FM (Direct File To Device—Special Forms Response), II: 2-175
 - FR (Designate Final Reel), II: 2-179
 - IL (Assign Labeled File), II: 2-192
 - MR (Remove Duplicate File), II: 2-234
 - OF (Optional File Indication), II: 2-245
 - OU (Direct File To Device), II: 2-257
 - PATCH (Patch File), II: 2-264
 - PERFORM (Schedule DMPALL), II: 2-280
 - REMOVE (RM) removes any file, II: 2-315
 - RF, RFP (Remove Backup File From Disk or Pack), II: 2-320
 - RM removes duplicate files - old, II: 2-322
 - SHOW FILE, II: 2-344
- file identifiers, II: 1-8
 - reserved by MCP, II: 1-9
- file I/O operations
 - rules for shared systems, III: 30-2
- file names
 - format for, III: 10-4
- file selection criteria
 - for QWIK Disk
 - file access, III: 29-20
 - file activity, III: 29-21
 - file size, III: 29-19
 - file volatility, III: 29-20
- File Specification screen, I: 4-8
- file type analysis
 - for QWIK Disk, III: 29-12
 - file category list, III: 29-13
- file types
 - for Editor files
 - in SRTUTL, III: 23-1
- FILES keyword
 - with DISPKV program, III: 3-12
- firmware
 - disk pack controller, I: xi
 - downloading
 - restrictions for, III: 11-1
 - to data communications processor (DCP), III: 11-1
 - to programmable controller, III: 11-1
 - to Uniline device, III: 11-1
 - loading
 - to disk controller, III: 12-1
 - to disk pack controller, III: 12-1
 - to Shared System Processors (SSPs), III: 12-1
 - to Uniline DLPs, III: 12-1
 - loading the, I: 1-13, 3-23
 - SSP, I: xi
 - Uniline, I: xi
- firmware code file
 - transfer of, III: 1-2
- firmware file
 - HSTLAD, I: 3-8
 - HSTLKC, I: 3-8
 - HSTLQH, I: 3-8
 - HSTLUD, I: 3-8
 - IPPFMW, I: 3-8
 - LSDIAG, I: 3-9
 - USP3BV, I: 3-8
 - UST2BH, I: 3-8
- firmware files
 - comparing, III: 2-2
 - loading, III: 2-2
- firmware interface errors
 - for Data Communications Processors, III: 27-5
- firmware loading
 - to Uniline from V 300
 - instructions, III: 26-14
 - to Uniline from V 500
 - instructions, III: 26-14
- FLAME, I: 4-49
- FLAME software graphs
 - for MCP idle time analysis
 - for QWIK Disk, III: 29-6

FLAME software graphs (cont.)
 for MCP overlay analysis
 for QWIK Disk, III: 29-9
 for program overlay analysis
 for QWIK Disk, III: 29-9
 to monitor system environment
 for QWIK Disk, III: 29-18
 use for file type analysis
 for QWIK Disk, III: 29-12
 FM (Direct File To Device—Special Forms
 Response), II: 2-175
 FN (List File Names), II: 2-177
 FP (Flush Program), II: 2-178
 FR (Designate Final Reel), II: 2-179
 front-end processor (FEP), I: 5-53
 function key
 CLOCK, I: 3-44
 ECM, I: 3-44
 EVENT, I: 3-44
 INSTR, I: 3-44
 M STAT, I: 3-44
 MSGs, I: 3-44
 REDO, I: 3-44
 REMOTE, I: 3-44

G

GCR tape drive, I: 4-17, 5-54
 adding, II: 2-389
 general messages
 output messages, IV: C-2
 general system messages
 definition, IV: 1-2
 general system performance
 with QWIK Disk, III: 29-4
 GENERATE (Schedule DMPALL For
 Execution), II: 2-180
 GO operation
 with LDCNTL program, III: 10-8
 GO (Restart A Stopped
 Program), II: 2-182
 GT (Initiate Trace), II: 2-183
 GTDK
 option, I: 4-47, II: 2-353
 record, I: 5-61
 guard file, I: 5-2
 GUARDED file access, I: 5-4
 guide
 about this, III: v

H

HALT command, I: 3-38
 halt/load, I: A-8
 automatic initiation of a, I: 4-47
 conversational, I: 3-32
 HL (Halt/Load Request), II: 2-187
 multisystem shared environment
 conventional, I: 3-31
 options, I: 3-12
 BYPASS BOTH, I: 3-12
 BYPASS DISK, I: 3-12
 BYPASS PACK, I: 3-12
 CONVERSE, I: 3-12
 RELOAD, I: 3-12
 RELOAD OVER, I: 3-12
 performing
 with QWIK Disk, III: 29-31
 V 300
 conventional, I: 3-30
 V 500
 conventional, I: 3-30
 with SYSUP facility
 chart of types, III: 25-2
 halt/load failure
 from powered off state
 procedure for V 300/V 400
 Series, IV: 3-7
 from powered on state
 procedure for V 300/V 400
 Series, IV: 3-6
 procedure for V 500 Series, IV: 3-17
 hardware mnemonic, I: 5-16
 hardware type
 for QWIK Disk, III: 29-33
 HDR option, I: 5-68
 <hdw>
 output message abbreviation, IV: C-1
 header, III: 31-10
 HElp command, I: 4-2
 help screens, I: 4-3
 <hh:mm>
 output message abbreviation, IV: C-1
 high I/O volume files
 moved to QWIK Disk
 discussion of, III: 29-6
 highest addressable sector
 chart of, III: 3-10
 HL (Halt/Load Request), II: 2-187
 HN (Hostname), II: 2-189
 HOme command, I: 4-2

Index

- hostname, I: 4-60
- HOSTNAME
 - record, I: 5-22
- HSTIMEOUT
 - option, I: 4-51
 - record, I: 5-26
- HSTLAD, I: 3-8
- HSTLKC, I: 3-8
- HSTLQH, I: 3-8
- HSTLUD, I: 3-8
- hub, I: A-8
- Hypercall/BCT Breakpoints
 - Debug Facility
 - description of, III: 28-18
- I**
- ICRAM2 verify failure
 - setting of IOP option, IV: 4-12
- ID, I: 5-3
- ID field, I: 4-59
- ID (Invoke DEBUG Session), II: 2-191
- IIO cabinet
 - control panel, I: 1-10
 - powering on the, I: 2-3
- IIO instruction
 - failure during system load, IV: 2-1
- IL (Assign Labeled File), II: 2-192
- IL command
 - with DMPANL program, III: 6-11
- Image Laser Printing System (IPP), I: 4-23
- IN file access, I: 4-57, 5-4
- IN (Insert Data into Program), II: 2-195
- Inactive DLP screen, I: 4-38
- InCRAM verify failure
 - setting of QWIK Disk option, IV: 4-12
- independent runners (IRs)
 - MCPTASKS Limit, II: 2-33
 - show all active, II: 2-344
- informational messages
 - DISPKV program, III: 3-37
- INHIBITED, I: 5-15, 5-19
- INIT command, I: 3-24
- initialization
 - V 300, I: 3-3
 - V 500, I: 3-15
- initialization failure (V 300/V 400 Series)
 - cold-start failure, IV: 3-9
 - control store errors, IV: 3-3
 - diskette fails to load, IV: 3-2
- initialization failure (cont.)
 - halt/load from powered off state
 - failure, IV: 3-7
 - halt/load from powered on state
 - failure, IV: 3-6
 - LOADFW failure, IV: 3-5
 - reload failure, IV: 3-8
 - Uniline firmware, IV: 3-4
- initialization failure (V 500 Series)
 - clear memory procedure
 - errors, IV: 3-14
 - control store errors, IV: 3-13
 - halt/load from powered on state
 - failure, IV: 3-17
 - LOADFW failure, IV: 3-16
 - power-on failure, IV: 3-11
 - SMC failure, IV: 3-12
 - Uniline firmware, IV: 3-15
- INITIALIZE command, I: 3-26
 - error summary report for, III: 3-15
 - functions
 - list of, III: 3-14
- input file for CONFIG utility, I: 4-9
- inquiry
 - AT (Initiate BNA Job Transfer), II: 2-36
 - AT (Pass Message to Remote Host), II: 2-38
 - AX (Pre-Respond to an ACCEPT), II: 2-39
 - AX (Respond to an ACCEPT), II: 2-41
 - BF, BFP (List Backup Files on Disk or Pack), II: 2-47
 - BFRF (Search, List, Remove Backup Files), II: 2-49
 - CA (List Port File Candidates), II: 2-65
 - CD,CDP (List Inactive Pseudo Decks), II: 2-68
 - CN (Display Tape Number), II: 2-78
 - DC (Display Compilation Data), II: 2-118
 - DQ (Display System Messages), II: 2-145
 - FI (List Files in Use), II: 2-166
 - HN (Hostname), II: 2-189
 - KA (Analyze Disk Directory), II: 2-201
 - KS (Analyze Disk Space), II: 2-207
 - KX (Obtain User Disk Space), II: 2-210

- inquiry (cont.)
 - OL (Display Peripheral Status), II: 2-248
 - PD (Print Directory), II: 2-278
 - TI (Display Time Statistics), II: 2-372
 - TO (Type System Options), II: 2-379
 - WB (Display Bound Programs), II: 2-405
 - WC (Display Available Memory), II: 2-406
 - WD (Display Date), II: 2-408
 - WM (Display MCP Version), II: 2-416
 - WO (Display Active User QWIK Operation Counts), II: 2-418
 - WQ (Display STOQUE Count), II: 2-420
 - WS (Display Jobs in Schedule), II: 2-423
 - WT (Display Time), II: 2-424
- INSERT (Insert Data Into Program), II: 2-197
- INSTR command, I: 3-44
- INTERACTIVE DEBUG command
 - for Debug Facility
 - description of, III: 28-6
 - example of, III: 28-6
 - options for, III: 28-6
 - syntax for, III: 28-6
- interactive debugger, I: 5-7
- interlaced format, III: 31-5
- interprogram communication, I: 4-49
- intersystem connect (ISC), I: A-8
 - configuration, I: 4-34
 - UNIT record, I: 5-44
- intrinsic programs
 - B 974LD, III: 1-1
 - MERG:, III: 15-1
 - SORT:, III: 22-1
 - SORTx
 - with MAKTRN program, III: 13-1
 - SORTx
 - SORTy
 - with the MAKTRN program, III: 13-1
- WB (Display Bound Programs), II: 2-405
- I/O analysis
 - for QWIK Disk
 - I/O and memory bound category, III: 29-10
 - I/O bound category, III: 29-10
 - memory bound category, III: 29-10
- I/O analysis (cont.)
 - system bound category, III: 29-10
 - system category list, III: 29-10
- I/O control
 - DBCL (Cancel DLP), II: 2-109
 - DBIC (Initialize Controller Parameters), II: 2-111
 - FP (Flush Program), II: 2-178
 - PRP (Change Processing and I/O Priority), II: 2-289
 - PRP (Set Processing and I/O Priority), II: 2-301
- I/O error handling
 - SYSTEM/COPY utility program
 - description of, III: 24-8
- I/O error messages
 - SYSTEM/COPY utility program
 - description of, III: 24-8
 - format for, III: 24-8
 - list of, III: 24-8
- IO file access, I: 4-57, 5-4
- I/O memory concentrator (IOMC), I: 1-4
- I/O operating analysis
 - for QWIK Disk
 - list of types, III: 29-8
- I/O path, I: A-2
- I/O priority, I: A-13
- I/O processor, V 300, I: 1-4
- I/O queuing, I: 4-44
- I/O subsystem
 - V 300, I: 1-4
 - V 500, I: 1-4
- I/O volume analysis
 - for file classes
 - for QWIK Disk, using Run Log (RLOG), III: 29-14
 - for individual files
 - for QWIK Disk, using Run Log (RLOG), III: 29-14
 - for MCP files
 - for QWIK Disk, using FLAME software graphs, III: 29-14
- IOP ERROR
 - recovery procedure
 - for V 300/V 400 Series, IV: 4-13
- IOP option
 - effect on ICRAM2 verify, IV: 4-12
- IPP
 - adding, II: 2-387
- IPPFMW, I: 3-8
- IR (Initiate DMSII Recovery), II: 2-199

Index

IRs (*See* independent runners
(IRs)), II: 2-344
ISAM performance
with QWIK Disk, III: 29-4
ISC
adding, II: 2-388
option, II: 2-353
screen, I: 4-34
IVR functions, III: 31-1
IVR (*See* INITIALIZE command)

J

<j-s>
output message abbreviation, IV: C-1
@<j-s>
output message abbreviation, IV: E-1
job control
AFTER (Hold Program), II: 2-12
GT (Initiate Trace), II: 2-183
LOCK (Lock Function), II: 2-227
LP (Lock Program), II: 2-228
MEMORY (Allocate Memory For
Program Execution), II: 2-230
NL (Resume at New
Location), II: 2-242
NT (Terminate Trace), II: 2-243
job initiation
COMPILE (Compile Program), II: 2-80
DEBUG (Attach Program To Debug
Session), II: 2-121
EXECUTE (Execute Library
Program), II: 2-162
GENERATE (Schedule DMPALL For
Execution), II: 2-180
PC, PCP (Punch Backup
File), II: 2-275
PERFORM (Schedule
DMPALL), II: 2-280
PM (Print Memory Dump), II: 2-284
RA (Remove After-Linkage), II: 2-312
RERUN (Log a Rerun
Operation), II: 2-319
RN, RNP (Activate Pseudo
Reader), II: 2-324
RUN (Initiate Execution of
Program), II: 2-334
job maintenance
FILE (Equate File-Name With
File-Identifier Or
Medium), II: 2-167

job maintenance (cont.)
FN (List File Names), II: 2-177
GO (Restart A Stopped
Program), II: 2-182
IN (Insert Data into
Program), II: 2-195
INSERT (Insert Data Into
Program), II: 2-197
PR (Change Priority), II: 2-289
PRIORITY (Common Priority
Setting), II: 2-301
SK (Skip Backup Records), II: 2-348
ST (Suspend Program
Processing), II: 2-364
START (Restart A Stopped
Program), II: 2-365
STOP (Suspend Program
Processing), II: 2-367
TRACE (Trace a Program), II: 2-381
job termination
DP (Dump and Discontinue), II: 2-144
DS (Discontinue Program), II: 2-148
FR (Designate Final Reel), II: 2-179
MEMDUMP (Abnormal EOJ Memory
Dump), II: 2-229
QT (Quit Program
Operation), II: 2-308
RS (Remove Job from
Schedule), II: 2-333

K

KA (Analyze Disk Directory), II: 2-201
keyboard input messages
contrasted with MCP control
instructions, II: 1-1
KP (Print Disk Segments), II: 2-205
KS (Analyze Disk Space), II: 2-207
KX (Obtain User Disk Space), II: 2-210

L

label, III: 31-12
LABEL command
restrictions for, III: 3-27
LABEL1 (Change Tape Label
Format), II: 2-212
LABEL1 record, I: 5-23
LAK disk, I: A-5, III: 31-1

- LAK disk pack, initializing and verifying
 - a, I: 3-24
- LANGUAGES record, I: 5-26
- LD command
 - with LDCNTL program, III: 10-4, 10-6, 10-7
- LDCNTL utility program, I: 5-9, II: 2-308
 - card format
 - BCL, III: 10-5
 - EBCDIC, III: 10-5
 - description of, III: 10-1
 - initiating, III: 10-5
 - invalid characters
 - in control instructions, III: 10-11
 - MCP control instructions used with
 - list of, III: 10-2
 - parameters
 - discussion of, III: 10-13
 - pseudo card file
 - format of, III: 10-11
 - system commands used with
 - list of, III: 10-2
- LDHOST utility program
 - automatic operation
 - conditions required for, III: 11-2
 - description of, III: 11-1
 - failure
 - discussion of, III: 11-1
 - parameters
 - list of, III: 11-3
 - with Data Communications Processor, III: 27-2
- LDP command
 - with LDCNTL program, III: 10-4, 10-6, 10-7
- LH command
 - to reactivate a B 974, III: 1-1
 - with CP option, III: 2-1
 - with DCP option, III: 27-2
 - with DUMP option, III: 2-2
 - with LDHOST program, III: 11-3
 - with NCP option, III: 2-1
 - with WARM option, III: 1-2, 2-1
- LIB
 - option, I: 4-47, II: 2-353
 - record, I: 5-61
- LIBMAINT functions, I: 4-42
- library maintenance, I: 4-55
 - ADD (Copy Files from Library Tape to Disk or Diskpack, II: 2-11
- library maintenance (cont.)
 - ALTER FILE changes LIBMAINT and SQUASH status, II: 2-28
 - CHECK (Verify Library Tape), II: 2-76
 - COMPARE (Compare Files On Different Media), II: 2-79
 - COPY (Invoke File Transfer Utility), II: 2-86
 - COPYADD (Transfer Files), II: 2-99
 - DIR (List File Directory), II: 2-125
 - DUMP (Copy Files To Library Tape), II: 2-150
 - EXECUTE (Execute Library Program), II: 2-162
 - LOAD (Move Files From Library Tape to Disk or Diskpack), II: 2-226
 - MERGE (Transfer Files from Multiple Sources, Resolving Duplicates), II: 2-232
 - MOVE (Move Files to New Media), II: 2-233
 - UNLOAD (Move Library Files Between Media, Remove Old Files), II: 2-397
- library maintenance messages
 - SYSTEM/COPY utility program
 - description of, III: 24-7
 - format for, III: 24-7
 - list of, III: 24-7
- library tape
 - verify, II: B-4
- LIMIT DCPBUF record
 - for Data Communications Processors, III: 27-3
- LIMIT DCPQUE record
 - for Data Communications Processors, III: 27-2
- Limit Options Value Override
 - screen, I: 4-43
- LIMIT record, I: 5-25
 - BLT, I: 5-29
 - configuration file errors, IV: B-3
 - DCPBUF, I: 5-29
 - DCPQUE, I: 5-30
 - DELAY record, I: 5-30
 - DISKPACKS, I: 5-25
 - DMSDBP, I: 5-25
 - DMSTATUS, I: 5-25
 - DMSUSERS, I: 5-25
 - for special diagnostics tasks, I: 5-29
 - HSTIMEOUT, I: 5-26

Index

- LIMIT record (cont.)
 - LANGUAGES, I: 5-26
 - MIX, I: 5-26
 - PBTBLK, I: 5-26
 - PRIORITYQ, I: 5-27
 - PRNRPA, I: 5-26
 - PRNRPB, I: 5-26
 - QWIKPool, I: 5-27
 - QWKMEM, I: 5-27
 - READERS, I: 5-27
 - STDSTATUS, I: 5-30
 - STOQBLOCKS, I: 5-27
 - STOQMINBLK, I: 5-27
 - STOQNames, I: 5-28
 - STOQSIZE, I: 5-28
 - TRAKBUFFER, I: 5-30
 - TSMFENCE, I: 5-30
 - UNITS, I: 5-28
- limit specification, I: 4-50
- list files
 - with DMPALL program
 - syntax elements for, III: 5-2
 - syntax for, III: 5-2
- LN command
 - with ECMANL program, III: 9-1
- LNA command
 - with ECMANL program, III: 9-1
- LNE command
 - with ECMANL program, III: 9-1
- LNM command
 - with SNPANL program, III: 20-1
- LOAD, II: B-7
- LOAD CS command, I: 3-39
- LOAD MCP command, I: 3-40
- LOAD (Move Files From Library Tape to Disk or Diskpack), II: 2-226
- LOAD SYSTEM command, I: 3-41
- Loadable Operating System (LOS)
 - for Image Page Printer (IPP)
 - how to load, III: 11-3
- loader program, I: 3-41
- LOADFW failure
 - procedure for V 300/V 400 Series, IV: 3-5
 - procedure for V 500 Series, IV: 3-16
- LOADFW utility program, I: 3-23
 - description of, III: 12-1
 - executing
 - from a V 300 system, III: 12-2
 - from a V 500 system, III: 12-2
- LAK disk pack functions
 - Initialize, III: 12-1
- LOADFW utility program (cont.)
 - Relocate, III: 12-1
 - Verify, III: 12-1
 - messages
 - error, III: 12-5
 - normal, III: 12-3
 - operating instructions, III: 12-2
 - loading files
 - for PTD programs, III: 3-31
- LOADMP
 - and dump, II: B-5
 - and file copy, II: B-2
 - and verify, II: B-4
- LOADMP and PACKUP
 - library files
 - listing with DMPALL, III: 5-21
- LOADMP and PACKUP tape directories
 - listing
 - with DMPALL program, III: 5-22
- LOADMP utility, I: 4-40
- local mode, I: 2-2
- LOCK (Lock Function), II: 2-227
- Log
 - Maintenance, I: 4-52
 - Network, I: 4-52
 - Run, I: 4-52
 - SPO, I: 4-52
 - log size, I: 4-53
 - logging, I: A-9
 - CHARGE (Insert Charge Number), II: 2-74
- LOGGING
 - record, I: 5-61
- logging
 - RERUN (Log a Rerun Operation), II: 2-319
 - TEST (Mark A Test Operation), II: 2-371
- Logging and Dump Options
 - screen, I: 4-52
- logical disk subsystem, I: A-14
- logical record size, I: 5-3
- logical subsystems, I: A-5
- Look-Alike disk, III: 31-1
- look-alike format (LAK), I: A-5
- lowercase letters
 - translated to uppercase
 - with PBDPRN utility program, III: 17-1
- LP (Lock Program), II: 2-228
- LRS, I: 5-3
- LSDIAG, I: 3-9

M

- M STAT command, I: 3-44
- magnetic tape status table
 - display, I: 5-70
- mailbox SSP, I: 5-20
- MAIN LINK ENABLE, I: 1-9
- main memory buffer
 - Disk File Header, III: 10-12
 - IOAT entry, III: 10-12
- Main Menu
 - Debug Facility
 - description of, III: 28-11
 - ODT action, III: 28-2
- main processor (V 300), I: 1-3
- maintenance link enable switch, I: 1-9
- Maintenance Log, I: 4-52
- maintenance processor, I: 1-2
 - CLEAR MEM command, I: 3-37
 - commands, I: 3-37
 - commands
 - for QWIK Disk
 - description of, III: 29-29
 - HALT command, I: 3-38
 - LOAD CS command, I: 3-39
 - LOAD MCP command, I: 3-40
 - LOAD SYSTEM command, I: 3-41
 - power switching the V 500 IIO cabinet, I: 1-11
 - SET O command, I: 3-42
 - SET OPTIONS command, I: 3-42
 - TERM command, I: 3-43
 - V 300, I: 1-3
 - V 500, I: 1-3
 - V 500 function keys, I: 3-44
- maintenance test commands
 - for QWIK Disk
 - list of, III: 29-30
 - warning message for, III: 29-30
- MAKTRN program
 - description of, III: 13-1
 - functions
 - list of, III: 13-1
 - operating instructions, III: 13-1
 - options
 - list of, III: 13-2
 - parameter records
 - format for, III: 13-2
- master available table, III: 31-12
- master pack, III: 31-13
- Master Selection menu, I: 4-11
- MCP
 - ALTER SYSTEM changes current MCP file, II: 2-33
 - loading the, I: 1-12
 - memory
 - QWKMEM (Set MCP Overlay Buffer Pool), II: 2-311
 - overriding parameters, I: 5-6
 - placing in QWIK Disk, III: 29-28
 - MCP channel and unit numbers, I: ix, 3-20
 - MCP configuration file, I: 4-1
 - card deck, I: A-1
 - examples of, I: 6-1
 - loading from a minidisk (V 300), I: 3-4
 - loading the, I: 3-13
 - MCP configuration record, I: 5-1
 - editing a, I: 3-35
 - modifying a, I: 3-35
 - selecting a, I: 3-35
 - MCP control instructions
 - contrasted with keyboard input messages, II: 1-1
 - used with LDCNTL program
 - list of, III: 10-2
 - MCP idle time
 - determining
 - for QWIK Disk analysis, III: 29-6
 - examining
 - for QWIK Disk analysis, III: 29-7
 - MCP initialization
 - for Debug Facility
 - description of, III: 28-1
 - MCP interfaces
 - LIMIT records
 - description of, III: 27-2
 - UNIT records
 - description of, III: 27-2
 - MCP Limits screen, I: 4-50
 - MCP loader program, I: 3-41
 - MCP memory limit, I: 3-20
 - MCP name, I: xi
 - MCP options, I: 4-46
 - MCP overlays, I: A-11
 - related to performance
 - with QWIK Disk, III: 29-9
 - MCP source, I: x
 - MCP STOQUE routine, I: 5-28
 - MCP structures, III: 31-12
 - MCPNIF, I: 5-59

MCPTASKS

- Limit on Independent Runners, II: 2-33
- MCS result pool, I: 4-51
- MDCOPV utility program
 - analysis function
 - description of, III: 14-7
 - list of options, III: 14-8
 - syntax for, III: 14-8
 - bad tracks
 - description of, III: 14-17
 - conversion function
 - description of, III: 14-13
 - list of file types, III: 14-13
 - copy function
 - description of, III: 14-3
 - duplicating floppy disks, III: 14-4
 - from floppy disk, III: 14-4
 - list of options, III: 14-5, 14-12
 - restrictions for, III: 14-2
 - syntax for, III: 14-4
 - to floppy disk, III: 14-4
 - description of, III: 14-1
 - duplicating floppy disks
 - syntax for, III: 14-7
 - error messages
 - list of, III: 14-19
 - examples, III: 14-14
 - executing
 - syntax for, III: 14-2
 - with switch settings, III: 14-3
 - file names
 - external, III: 14-16
 - internal, III: 14-16
 - floppy disk
 - format requirements, III: 14-2
 - floppy disk file
 - syntax for conversion of, III: 14-13
 - floppy disk image files
 - description of, III: 14-11
 - syntax for, III: 14-12
 - functions
 - list of, III: 14-1
 - hardware supported
 - list of, III: 14-1
 - informational messages
 - list of, III: 14-16
 - input prompts
 - list of, III: 14-18
 - required records for, III: 14-2
 - switch setting
 - examples, III: 14-3

MDCOPV utility program (cont.)

- types of messages
 - description of, III: 14-16
- MEDIA, I: 4-61, 5-36
- MEM CARD XFER CLOBBER, I: 3-6
- MEMDUMP (Abnormal EOJ Memory Dump), II: 2-229
- memory
 - DP (Dump and Discontinue), II: 2-144
 - MEMDUMP (Abnormal EOJ Memory Dump), II: 2-229
 - MEMORY (Allocate Memory For Program Execution), II: 2-230
 - PM (Print Memory Dump), II: 2-284
 - PRM (Change Memory Priority), II: 2-289
 - PRM (Set Memory Priority), II: 2-301
 - QWKMEM (Set MCP Overlay Buffer Pool), II: 2-311
 - RQ (Remove Storage Queue Entries), II: 2-331
 - RX (Return Deleted Space), II: 2-336
 - WC (Display Available Memory), II: 2-406
 - XM (Remove Memory from System), II: 2-437
- MEMORY (Allocate Memory For Program Execution), II: 2-230
- memory cards, I: ix
- memory division
 - MCP
 - diagram of, III: 29-3
 - QWIK Disk
 - diagram of, III: 29-3
- memory dump, I: A-6
- memory parity error
 - recovering from, IV: E-2
- memory requirement
 - for object program
 - determining, III: 21-2
- memory use characteristic
 - with QWIK Disk, III: 29-5
- MERG: intrinsic program
 - functional description of, III: 15-1
 - input records
 - order of, III: 15-2
 - memory requirements for, III: 15-1
 - operation of, III: 15-1
 - overview of, III: 15-1
- MERGE (Transfer Files from Multiple Sources, Resolving Duplicates), II: 2-232

- message display, I: 5-70
 - message for
 - successful load, III: 11-2
 - unsuccessful load, III: 11-2
 - message from B 974
 - displayed on ODT, III: 1-1
 - messages
 - action, IV: 1-2
 - disposition
 - with firmware code file, III: 1-2
 - error messages, IV: 1-2
 - general system, IV: 1-2
 - recovery of, III: 1-2
 - system initialization, IV: 2-1
 - types, IV: 1-2
 - metatokens, IV: C-1
 - in output messages, IV: E-1
 - in system messages, IV: C-1
 - <mfid>
 - output message abbreviation, IV: C-1
 - @<mfid>
 - output message abbreviation, IV: E-1
 - MICR
 - option, I: 4-49, II: 2-353
 - record, I: 5-63
 - minidisks, V 300, I: 3-3
 - minimum peripheral configuration (V 500), I: 3-19
 - mix
 - DQ (Display System Messages), II: 2-145
 - MIX
 - option, I: 4-45
 - record, I: 5-26
 - mix control
 - RS (Remove Job from Schedule), II: 2-333
 - mix status, I: 5-70
 - MIXED, I: 5-54
 - MLGOUT
 - executing, III: 20-1
 - MLOG, I: 4-52
 - <mmss>
 - output message abbreviation, IV: C-1
 - MMDD channel and unit numbers, I: ix, x
 - MODify command, I: 4-2
 - MOREDISK record, I: 5-31
 - MOREDISK screen, I: 4-39
 - MOVE (Move Files To New Media), II: 2-233
 - MPCOPY program
 - with LNA command, III: 9-1
 - with LNE command, III: 9-1
 - MPE tape drive, I: 4-17, 5-54
 - adding, II: 2-389
 - MR (Remove Duplicate File), II: 2-234
 - MSGs command, I: 3-44
 - MT9 tape drive, I: 4-17, 5-54
 - adding, II: 2-389
 - MTC tape drive, I: 4-17, 5-54
 - multifile identifier (MFID), I: 5-23
 - multiple characters
 - assigning collated values to, III: 13-1
 - multiple processor, cold-start, I: 3-29
 - multisystem shared
 - configuration, I: A-1
 - address locks in a, I: 3-31
 - cold-start, I: 3-27
 - configuration files for a, I: 6-6
 - conventional halt/load, I: 3-31
 - declaring processors in a, I: 4-60
 - declaring systems in a, I: 5-5
 - environment for, III: 30-4
 - processor numbers, I: 4-61
 - shared disk SSP, I: 4-37
 - SSP channel number, I: 4-37
 - MX table, I: 5-70
- ## N
- NAME, I: 5-55
 - NAMESERIAL, I: 5-55
 - NCP
 - adding, II: 2-391
 - NCP screen, I: 4-29
 - network communication port (NCP)
 - NCP, I: 5-45
 - Network Definition Language
 - compilation on a B 974, III: 1-1
 - Network Log, I: 4-52
 - NEXt command, I: 4-3
 - NIFMRG utility program
 - DCP statement
 - examples of, III: 16-4
 - syntax for, III: 16-2
 - DCP statement options
 - list of, III: 16-3
 - description of, III: 16-1
 - END statement
 - description of, III: 16-4
 - examples of, III: 16-5

Index

- NIFMRG utility program (cont.)
 - syntax for, III: 16-4
 - file processing for, III: 16-1
 - initiating, III: 16-1
 - listing
 - description of, III: 16-6
 - MERGED statement
 - description of, III: 16-4
 - example of, III: 16-4
 - syntax for, III: 16-4
 - QUIT statement
 - description of, III: 16-5
 - syntax for, III: 16-5
 - statements longer than one line
 - examples of, III: 16-6
 - syntax errors
 - description of, III: 16-6
 - types of statements
 - list of, III: 16-2
 - with Data Communications Processors, III: 27-1
 - NL (Resume at New Location), II: 2-242
 - NLOG, I: 4-52
 - NO LIB, I: 5-3
 - NOA, I: 5-3
 - NONE file access, I: 5-4
 - nonstatus (NST) device, I: A-4
 - configuration, I: 4-33
 - declaring a, I: 5-46
 - declaring channels for a, I: 3-26
 - NOSTANDBY option
 - of KA command, II: 2-203
 - of KS command, II: 2-208
 - of PA command, II: 2-260
 - of PS command, II: 2-304
 - NOT READY condition, IV: C-6
 - NST
 - adding, II: 2-392
 - NST screen, I: 4-33
 - NT (Terminate Trace), II: 2-243
 - NW (Networking Command Prefix), II: 2-244
- O**
- object code files
 - listing from disk
 - with DMPALL program, III: 5-21
 - OCS
 - adding, II: 2-392
 - channel number, I: x
 - OCS (cont.)
 - configuration, I: 4-25
 - making the V 500 OCS
 - operational, I: 3-22
 - placement of an, I: 1-2
 - OCS, ODT screen, I: 4-25
 - ODT
 - AD (Specify ODT/OCS Screen Parameters), II: 2-4
 - adding, II: 2-392
 - CC (Enter Control Instructions), II: 2-66
 - channel number, I: x
 - configuration, I: 4-25
 - entering commands through the, II: 1-2
 - ES (End Screen), II: 2-160
 - input messages
 - how to respond to, III: 3-31
 - MCP response to ODT input, II: 1-3
 - SS (Send Message), II: 2-363
 - WRU (Display ODT Access Code), II: 2-422
 - OF (Optional File Indication), II: 2-245
 - offline, setting processor, II: 2-32
 - OI keyword
 - with DISPKV program, III: 3-8
 - OK command
 - used to recover from block count error, IV: C-5
 - OK (Continue Processing Suspended Program), II: 2-246
 - OL (Display Peripheral Status), II: 2-248
 - online, setting processor, II: 2-31
 - opcode breakpoint
 - Debug Facility
 - description of, III: 28-19
 - opcode breakpoint actions
 - Debug Facility
 - list of, III: 28-19
 - OPEN
 - option, I: 4-47, II: 2-353
 - record, I: 5-63
 - operating system, loading the, I: 1-12
 - operator control station (*See* OCS)
 - operator display terminal (*See* ODT)
 - operator prompts
 - DISPKV program, III: 3-38
 - OPRTR HALT/TERM stop cause
 - recovery procedure
 - for V 500 Series, IV: 4-34

- optional files
 - description of, III: 3-12
 - ORS-DLP, I: A-3
 - OT (Display Program Data), II: 2-256
 - Other Devices menu, I: 4-20
 - OU (Direct File To Device), II: 2-257
 - OUT file access, I: 4-57, 5-4
 - output device options
 - for Debug Facility
 - list of, III: 28-16
 - output file for CONFIG utility, I: 4-9
 - output messages
 - action messages, IV: D-1
 - general messages, IV: C-2
 - program error, IV: E-2
 - system error, IV: E-2
 - overflow breakpoint
 - Debug Facility
 - description of, III: 28-20
 - Overlay
 - QWIKPOOL (Set Program Overlay Buffer Pool), II: 2-310
 - overlay analysis
 - for QWIK Disk
 - list of types, III: 29-7
 - overlays
 - amount of memory reserved
 - for, I: 4-44
 - management of, I: A-11
 - QWIKPOOL, I: A-11
 - setting limits for, I: 4-43
- P**
- <p-n>
 - output message abbreviation, IV: C-2
 - @<p-n>
 - output message abbreviation, IV: E-2
 - PA (Analyze Pack Directory), II: 2-259
 - PACK, I: 5-16
 - pack, III: 31-1
 - ALTER (Change System Parameters), II: 2-29
 - base, III: 31-13, 31-19
 - building MCP structures on
 - a, III: 31-18
 - coexistence of versions, III: 31-26
 - continuation, III: 31-13, 31-19
 - converting version 1 to version
 - 2, III: 31-25
 - pack (cont.)
 - DBMM (Set/Reset Maintenance Mode), II: 2-112
 - family, III: 31-9
 - familyname, III: 31-14
 - IVR procedures, III: 31-1
 - label, III: 31-12
 - labeling a, III: 31-9
 - labeling a version 1, III: 31-18
 - labeling a version 2, III: 31-23
 - masking the family name, III: 31-14
 - master, III: 31-13
 - PA (Analyze Pack Directory), II: 2-259
 - PACK (Add Pack), II: 2-262
 - PO (Power Off Diskpack), II: 2-286
 - PP (Print Pack Sectors), II: 2-287
 - PS (Analyze Pack Space), II: 2-303
 - purging files from a version
 - 1, III: 31-19
 - purging files from a version
 - 2, III: 31-23
 - rebuilding MCP structures on
 - a, III: 31-23
 - rebuilding version 1 structures on
 - a, III: 31-18
 - renaming a version 1, III: 31-19
 - renaming a version 2, III: 31-23
 - resource pack, III: 31-20
 - restricted, III: 31-14, 31-20
 - restrictions, III: 31-20
 - RX (Return Deleted Space), II: 2-336
 - SQP (Squash Pack), II: 2-360
 - structures, III: 31-12
 - system resource, III: 31-14
 - unrestricted, III: 31-14
 - unrestricted pack, III: 31-20
 - version 1, III: 31-12
 - version 1 families, III: 31-13
 - version 2, III: 31-12
 - version 2 families, III: 31-19
 - versions, III: 31-12
 - versions
 - comparison of, III: 31-24
 - XP (Remove Pack Sectors), II: 2-438
 - PACK (Add Pack), II: 2-262
 - pack controller, III: 31-4
 - pack drive, III: 31-4
 - pack maintenance
 - CHANGE (Change File ID), II: 2-70
 - MR (Remove Duplicate File) -
 - new, II: 2-234

Index

- pack maintenance (cont.)
 - REMOVE (Delete Files), II: 2-315
 - RF, RFP (Remove Backup File From Disk or Pack), II: 2-320
 - RM (Remove Duplicate File) - old, II: 2-322
- PACK record, I: 5-32
- pack types, III: 31-1
- PACKUP
 - and dump, II: B-5
 - and verify, II: B-4
- PACKUP utility, I: 4-40
- PAGE option, I: 5-69
- PANDM3 minidisk, I: 3-3
 - inserting the, I: 3-5
- PANSM3 minidisk, I: 3-3
 - inserting the, I: 3-5
- parity error
 - response to, IV: C-6
 - system message for, IV: C-5
- parser error messages, I: B-2
- password field, I: 4-61
- PATCH (Patch File), II: 2-264
- PATCH record, I: 5-34
- PATCHK, II: 2-269
- PB command
 - with PBDPRN program, III: 17-6
- PBD
 - option, I: 4-47, II: 2-353
 - record, I: 5-63
- PBDOUT, II: B-11
- PBDPRN utility program, I: 5-9, II: 2-308
 - auto printing
 - instructions for, III: 17-7
 - syntax for, III: 17-7
- AX errors
 - list of, III: 17-9
- description of, III: 17-1
- error recovery messages
 - for image page printer, III: 17-14
- examples of, III: 17-6
- executing, III: 17-7
- image page printer messages
 - list of, III: 17-12
- messages for printers
 - list of, III: 17-10
- options
 - list of, III: 17-2
- SEARCH string
 - syntax for, III: 17-4
- PBDPRN utility program (cont.)
 - syntax errors
 - list of, III: 17-8
 - syntax for, III: 17-1
 - used for auto printing, III: 17-7
- PBP
 - option, I: 4-47, II: 2-353
 - record, I: 5-63
- PBPS
 - option, I: 4-47, II: 2-353
 - record, I: 5-63
- PBT
 - option, I: 4-47, II: 2-353
 - record, I: 5-63
- PBTBLK
 - option, I: 4-44
 - record, I: 5-26
- PBTOUT, II: B-11
- PC, PCP (Punch Backup File), II: 2-275
- PCD
 - option, I: 4-47, II: 2-353
 - record, I: 5-64
- PCH
 - option, I: 4-47, II: 2-353
 - record, I: 5-64
- PCHOUT utility program, II: 2-308
- PCOPY utility program
 - description of, III: 18-1
 - examples of, III: 18-2
 - operating instructions for, III: 18-1
- PCP
 - option, I: 4-48, II: 2-353
 - record, I: 5-64
- PCRM
 - option, I: 4-48, II: 2-353
 - record, I: 5-64
- PCRXRF (See Pseudo-Reader Cross Reference Directory)
- PD (Print Directory), II: 2-278
- PEEK function
 - Debug Facility
 - description of, III: 28-25
- peirpherals
 - OL (Display Peripheral Status), II: 2-248
- PERFORM (Schedule DMPALL), II: 2-280
- performance factors
 - for QWIK Disk, III: 29-3
- performance improvement
 - with QWIK Disk, III: 29-5

- peripheral activity characteristic
 - with QWIK Disk, III: 29-5
- peripheral analysis
 - for QWIK Disk
 - using FLAME software
 - graphs, III: 29-15
- peripheral maintenance
 - AC (Purge And Place Tape Number In Label), II: 2-2
 - AD (Specify ODT/OCS Screen Parameters), II: 2-4
 - PG (Purge Magnetic Tape), II: 2-282
 - PO (Power Off Diskpack), II: 2-286
 - RP (Ready and Purge Magnetic Tape), II: 2-329
 - RW (Rewind and Unload Tape), II: 2-335
 - RY (Ready Peripheral Device), II: 2-338
 - SV (Save a Peripheral Unit), II: 2-368
 - UL (Assign Unlabeled File), II: 2-383
- peripheral performance
 - with QWIK Disk, III: 29-4
- Peripheral Test Driver (PTD) program
 - to initialize
 - certain disk pack types, III: 3-31
- Peripheral Test Driver (PTD) programs
 - list of, III: 3-15
- peripherals
 - CK (Test Peripheral Device), II: 2-77
 - COPY (Invoke File Transfer Utility), II: 2-86
 - COPYADD (Transfer Files), II: 2-99
 - DA, DAP (Deactivate Pseudo Reader), II: 2-103
 - DIR (List File Directory), II: 2-125
 - DISPLAY/WHATS (Display Information), II: 2-131
 - DL (Delete Peripheral Devices), II: 2-135
 - FILE (Equate File-Name With File-Identifier Or Medium), II: 2-167
 - FM (Direct File To Device—Special Forms Response), II: 2-175
 - FP (Flush Program), II: 2-178
 - IL (Assign Labeled File), II: 2-192
 - MERGE (Transfer Files from Multiple Sources, Resolving Duplicates), II: 2-232
 - MOVE (Move Files to New Media), II: 2-233
 - peripherals (cont.)
 - MR (Remove Duplicate File), II: 2-234
 - OU (Direct File To Device), II: 2-257
 - RD, RDP (Remove Pseudo Decks), II: 2-313
 - REMOVE (Delete Files), II: 2-315
 - RM (Remove Duplicate File), II: 2-322
 - SI (Set Remote Device Identification), II: 2-347
 - UNIT (Add Peripheral Devices), II: 2-385
 - UR (Inhibit or Uninhibit Unit), II: 2-399
 - PG (Purge Magnetic Tape), II: 2-282
 - PKCOPY utility program
 - description of, III: 19-1
 - examples of, III: 19-3
 - messages
 - list of, III: 19-3
 - operating instructions for, III: 19-2
 - options
 - list of, III: 19-2
 - syntax for, III: 19-2
 - platter, III: 31-5
 - PM 1 command
 - starting DMPANL with, III: 6-1
 - PM command
 - with DMPANL program, III: 6-11
 - with DMPCPY program, III: 7-1
 - PM (Print Memory Dump), II: 2-284
 - PN keyword
 - with DISPKV program, III: 3-8
 - PO (Power Off Diskpack), II: 2-286
 - POKE function
 - Debug Facility
 - description of, III: 28-27
 - powering on
 - requirements for
 - with QWIK Disk, III: 29-31
 - power-off procedures, I: 2-6
 - V 300
 - ac power, I: 2-6
 - dc power, I: 2-6
 - V 500
 - ac power, I: 2-6
 - dc power, I: 2-6
 - power-on error
 - recovery procedure
 - for V 500 Series, IV: 4-21
 - power-on failure
 - procedure for V 500 Series, IV: 3-11

Index

- power-on procedures
 - dc
 - V 300, I: 2-1
 - V 500, I: 2-2
 - V 500
 - I/O cabinet, I: 2-3
 - local mode, I: 2-2
 - remote mode, I: 2-3
- PP (Print Pack Sectors), II: 2-287
- PR (Change Priority), II: 2-289
- Preallocated Disk Space screen, I: 4-58
- precold-start procedure
 - V 300, I: 3-5
 - V 500, I: 3-15
- PREvious command, I: 4-3
- primary channel, I: A-6
- primary disk pack family, I: 4-41
- PRInt command, I: 4-3
- PRINT (Print Files), II: 2-291
- printer
 - buffered, I: 5-50
 - configuration, I: 4-23
 - status, I: 5-70
 - train, I: 5-56
- printer backup files, I: 5-6, A-12
 - blocking factor, I: 4-44
 - PBDPRN utility program
 - used to create, III: 17-1
 - records per area, I: 4-44
 - with SNAP images
 - creating, III: 20-1
- printer backup tapes
 - blocking factor, I: 4-44
 - listing A Series
 - with DMPALL program, III: 5-22
- Printer screen, I: 4-23
- printing
 - KP (Print Disk Segments), II: 2-205
 - PM (Print Memory Dump), II: 2-284
 - PP (Print Pack Sectors), II: 2-287
 - PRINT (Print Files), II: 2-291
 - SK (Skip Backup Records), II: 2-348
- PRIORITY, I: 5-16
- PRIORITY (Common Priority Setting), II: 2-301
- priority setting, I: 5-9
- PRIORITYQ
 - option, I: 4-44
 - record, I: 5-27
- PRIVATE file access, I: 5-4
- PRM, I: 5-9
- PRM (Change Memory Priority), II: 2-289
- PRM (Set Memory Priority), II: 2-301
- PRN
 - option, I: 4-48, II: 2-353
 - record, I: 5-64
- PRN table, I: 5-70
- PRNRPA
 - option, I: 4-44
 - record, I: 5-26
- PRNRPB
 - option, I: 4-44
 - record, I: 5-26
- PROC ERROR
 - recovery procedure
 - for V 300/V 400 Series, IV: 4-12
- PROC HALT
 - recovery procedure
 - for V 300/V 400 Series, IV: 4-11
 - for V 500 Series, IV: 4-30
- PROCESSOR
 - record, I: 5-35
- processor
 - SHOW PROCESSOR, II: 2-344
 - start running, II: 2-31
 - stop running, II: 2-31
- processor memory modules (V 500),
 - clearing the, I: 3-21
- PROCESSOR record, I: 6-6
 - configuration file errors, IV: B-2
- program
 - AX (Pre-Respond to an ACCEPT), II: 2-39
 - AX (Respond to an ACCEPT), II: 2-41
 - COMPILE (Compile Program), II: 2-80
 - DC (Display Compilation Data), II: 2-118
 - DISPLAY/WHATS (Display Information), II: 2-131
 - DS (Discontinue Program), II: 2-148
 - EXECUTE (Execute Library Program), II: 2-162
 - FI (List Files in Use), II: 2-166
 - FN (List File Names), II: 2-177
 - FP (Flush Program), II: 2-178
 - GO (Restart A Stopped Program), II: 2-182
 - IN (Insert Data into Program), II: 2-195
 - INSERT (Insert Data Into Program), II: 2-197
 - LOCK (Lock Function), II: 2-227

- program (cont.)
- LP (Lock Program), II: 2-228
 - MEMORY (Allocate Memory For Program Execution), II: 2-230
 - OF (Optional File Indication), II: 2-245
 - OK (Continue Processing Suspended Program), II: 2-246
 - OT (Display Program Data), II: 2-256
 - ST (Suspend Program Processing), II: 2-364
 - START (Restart A Stopped Program), II: 2-365
 - STOP (Suspend Program Processing), II: 2-367
 - SW (Set Programmatic Switches), II: 2-369
 - UP (Unlock Program), II: 2-398
 - VALUE (Insert Value into a Program), II: 2-403
- program error
- output messages, IV: E-2
- program overlays, I: A-11
- related to performance with QWIK Disk, III: 29-9
- programming considerations
- for SYSUP facility, III: 25-2
- PROPS
- entering commands through, II: 1-6
- PRP, I: 5-9
- PRP (Change Processing and I/O Priority), II: 2-289
- PRP (Set Processing And I/O Priority), II: 2-301
- PRS, I: 5-9
- PRS (Change Scheduling Priority), II: 2-289
- PRS (Set Schedule Priority), II: 2-301
- PS (Analyze Pack Space), II: 2-303
- pseudo card
- BCL (Mark Beginning Of Card Data File), II: 2-43
 - CD,CDP (List Inactive Pseudo Decks), II: 2-68
 - CV, CVP (Convert Punch Backup File To Pseudo Card Deck), II: 2-100
 - DA, DAP (Deactivate Pseudo Reader), II: 2-103
 - EBCDIC (Mark Beginning of Card Date File), II: 2-153
 - pseudo card (cont.)
 - END/ENDCTL (End of Card File or Control Instructions), II: 2-156
 - RD, RDP (Remove Pseudo Decks), II: 2-313
 - RN, RNP (Activate Pseudo Reader), II: 2-324
 - SD (Set Deck Limit), II: 2-342
- pseudo card deck, I: A-13
- pseudo card files
- activating, III: 10-5
 - assignment of
 - procedure for, III: 10-12
 - contents of, III: 10-10
 - creating
 - for efficiency, III: 10-1
 - functions of, III: 10-1
 - manual procedure, III: 10-2
 - more than one, III: 10-10
 - procedure for, III: 10-1
 - procedures for, III: 10-4
 - deleting, III: 10-7
 - file identifier, III: 10-4
 - format of, III: 10-10
 - instruction link for, III: 10-13
 - LIMIT records
 - used with, III: 10-3
 - maximum number permitted, III: 10-6
 - MCP options
 - USE records for, III: 10-3
 - naming convention, III: 10-4
 - not ready status
 - description of, III: 10-13
 - ODT control instructions, III: 10-4
 - operating procedures
 - automatic, III: 10-6
 - discussion of, III: 10-6
 - manual, III: 10-6
 - recovery of, III: 10-8
 - RJE link for, III: 10-5
 - stacked
 - description of, III: 10-10
 - description of link field, III: 10-10
 - example of, III: 10-10
 - typical
 - examples, III: 10-9
 - pseudo card readers, maximum number of, I: 4-45
- Pseudo-Reader Cross Reference Directory (PCRXRf), III: 10-6

PTD (*See* Peripheral Test Driver programs)
PT-DLP, I: A-3
PUBLIC file access, I: 5-4
punch backup files, I: 5-6, A-1
 converting
 to pseudo card file, III: 10-13
PW field, I: 4-61

Q

QD (Query Debug), II: 2-307
QT (Quit Program Operation), II: 2-308
QUERY DEBUG command
 for Debug facility
 description of, III: 28-8
 syntax for, III: 28-8
QWIK
 option, I: 4-49, II: 2-353
 record, I: 5-64
QWIK Disk, I: 5-16
 applying to your system
 discussion of, III: 29-5
 as default disk
 discussion of, III: 29-32
 creating files for, III: 29-32
 programmatically, III: 29-32
 error conditions
 description of, III: 29-35
 file selection for, III: 29-18
 file sizing
 with COBOL compiler, III: 29-22
 with DMSII, III: 29-23
 with GEMCOS, III: 29-26
 with ISAM, III: 29-25
 with SORT, III: 29-23
 hardware type, III: 29-33
 installing
 procedures for, III: 29-1
 loading files
 discussion of, III: 29-32
 loading firmware
 discussion of, III: 29-32
 memory base, I: 3-20
 options for
 description of, III: 29-28
 overview of, III: 29-1
 performance expectations
 with COBOL compiler, III: 29-22
 with DMSII, III: 29-24
 with GEMCOS, III: 29-27

QWIK Disk (cont.)
 with ISAM, III: 29-26
 with SORT, III: 29-23
 placing files on
 procedures for, III: 29-31
 with COBOL compiler, III: 29-22
 with DMSII, III: 29-24
 with GEMCOS, III: 29-26
 with ISAM, III: 29-25
 with SORT, III: 29-23
 placing work files on
 procedure for, III: 29-32
 purging the, I: 3-7
 sectors, I: ix
 unloading files
 discussion of, III: 29-32
 with random I/O
 discussion of, III: 29-33
 with sequential I/O
 discussion of, III: 29-33
 with shared systems
 discussion of, III: 29-29
 write operation
 method for handling, III: 29-35
QWIK Disk analysis
 of system characteristics, III: 29-5
QWIK Disk errors
 result descriptors
 in maintenance log
 (MLOG), III: 29-36
 single bit
 in maintenance log
 (MLOG), III: 29-36
QWIK Disk operations
 description of, III: 29-29
QWIK Disk option
 effect on InCRAM verify, IV: 4-12
QWIK extension module, I: 4-49
QWIKDISK, III: 31-1
QWIKPOOL, I: 4-44, A-11
 for code file overlays
 with QWIK Disk, III: 29-4, 29-34
 for overlays
 with QWIK Disk, III: 29-5
 for QWIK Disk
 description of, III: 29-28
 option, I: 4-44
 record, I: 5-27
QWIKPOOL (Set Program Overlay Buffer Pool), II: 2-310

QWKMEM
 for QWIK Disk
 description of, III: 29-28
 option, I: 4-44
 record, I: 5-27
 used with QWIK Disk, III: 29-9
 QWKMEM (Set MCP Overlay Buffer
 Pool), II: 2-311

R

RA (Remove After-Linkage), II: 2-312
 railroad diagrams
 understanding, II: A-1
 RAM (See random access memory)
 random access memory (RAM), I: 1-12
 RAW action command, I: 4-63
 Raw Editor menu, I: 4-63
 RD command
 with LDCNTL, III: 10-7
 RD, RDP (Remove Pseudo
 Decks), II: 2-313
 RDP command
 with LDCNTL, III: 10-7
 reader sorter
 adding, II: 2-395
 READERS
 option, I: 4-45
 record, I: 5-27
 reader-sorter
 configuration, I: 4-21
 module, I: 4-49
 UNIT record, I: 5-51
 RECONFIGURE command
 restrictions for, III: 3-25
 RECONFIGUREL command
 restrictions for, III: 3-26
 record size, I: 4-55
 records, III: 31-8
 truncated
 with SORTx intrinsic
 program, III: 21-3
 records per area, I: 4-44, 4-56
 records per block, I: 4-56
 recovering from abnormal system
 operation, IV: 4-1
 recovery procedure
 fault screen displayed
 for V 300/V 400 Series, IV: 4-18
 for V 500 Series, IV: 4-36
 identifying

recovery procedure (cont.)
 for V 500 Series, IV: 4-19
 V 300 Series abnormal
 operation, IV: 4-2
 indeterminate problem
 for V 300/V 400 Series, IV: 4-26
 OCS unresponsive
 for V 500 Series, IV: 4-27
 ODT unresponsive
 for V 300/V 400 Series, IV: 4-6
 ODT unresponsive, status line of
 RUNNING
 for V 300/V 400 Series, IV: 4-9
 power-on error
 for V 500 Series, IV: 4-21
 slow OCS or data communications
 response
 for V 500 Series, IV: 4-25
 slow ODT or data communications
 response
 for V 300/V 400 Series, IV: 4-8
 status line displayed
 for V 500 Series, IV: 4-23
 for V 300/V 400 Series, IV: 4-5
 status line of EVENT STOP
 for V 300/V 400 Series, IV: 4-15
 status line of IOP ERROR
 for V 300/V 400 Series, IV: 4-13
 status line of PROC ERROR
 for V 300/V 400 Series, IV: 4-12
 status line of PROC HALT
 for V 300/V 400 Series, IV: 4-11
 for V 500 Series, IV: 4-30
 status line of SNAP HALT
 for V 300/V 400 Series, IV: 4-17
 status line of STOPPED
 for V 300/V 400 Series, IV: 4-16
 stop cause of ERROR
 for V 500 Series, IV: 4-31
 stop cause of OPRTR HALT/TERM
 for V 500 Series, IV: 4-34
 stop cause of SOFT SNAP
 for V 500 Series, IV: 4-35
 stop cause of STOP EVENT
 for V 500 Series, IV: 4-33
 system powered off
 for V 300/V 400 Series, IV: 4-4
 for V 500 Series, IV: 4-22
 unusual MCP messages
 for V 300/V 400 Series, IV: 4-7
 for V 500 Series, IV: 4-24
 Redirect Output screen, I: 4-10

Index

- REDO command, I: 3-44
- related product information, I: vii
- reload failure
 - procedure for V 300/V 400 Series, IV: 3-8
- RELOCATE command
 - restrictions for, III: 3-22
- REMOTE command, I: 3-44
- remote job entry (*See* RJE)
- remote mode, I: 2-2, 2-3
- REMOVE (Delete Files), II: 2-315
- removing files
 - MR for duplicate (new), II: 2-234
 - REMOVE (RM) for any file, II: 2-315
 - RF, RFP for backup, memory dump, II: 2-320
 - RM for duplicate (old), II: 2-322
- RENAME command
 - restrictions for, III: 3-24
- RERUN (Log a Rerun Operation), II: 2-319
- RESERVED, I: 5-13, 5-32, 5-54
- Resident Operating System (ROS)
 - for Image Page Printer, III: 11-3
- resource pack, III: 31-20
- REStart command, I: 4-3
- restricted pack, III: 31-14, 31-20
- restrictions
 - access level, II: 1-12
 - commands invalid from program for SPOM BCT, II: 1-11
 - commands invalid from ZIP or ZIPSPO, II: 1-11
 - commands invalid through card reader, II: 1-11
 - commands invalid through RJE, II: 1-12
 - commands valid from mix number 0, II: 1-12
 - commands valid through WFL, II: 1-12
 - for dollar sign (\$) cards
 - for SORT: intrinsic program, III: 22-8
 - for sort keys, III: 21-3
- LABEL command, III: 3-27
- RECONFIGURE command, III: 3-25
- RECONFIGUREL command, III: 3-26
- RELOCATE command, III: 3-22
- RENAME command, III: 3-24
- SINGLE command, III: 3-21
- SORT: intrinsic program
 - restrictions (cont.)
 - for output file, III: 22-3
 - system commands, II: 1-11
 - 659IVR command, III: 3-29
- result descriptor
 - examining
 - for control instructions, III: 10-11
- RETRY, I: 5-54
- RF, RFP (Remove Backup File From Disk or Pack), II: 2-320
- RJE
 - adding, II: 2-394
 - commands invalid through, II: 1-12
 - configuration, I: 4-32
 - devices, I: A-3
 - devices
 - entering commands through, II: 1-7
 - handler, I: 4-49
 - handlers, I: 5-9
 - module, I: A-3
 - option, I: 4-49, II: 2-353
 - record, I: 5-65
 - station, I: 4-49
- RLOG, I: 4-52
- RM (Remove Duplicate File), II: 2-322
- RMOV
 - option, I: 4-48, II: 2-353
 - record, I: 5-65
- RN command, I: 4-47
 - with LDCNTL program, III: 10-6
- RN, RNP (Activate Pseudo Reader), II: 2-324
- RNP command
 - with LDCNTL program, III: 10-6
- RO command
 - with APCR option, III: 10-7
- RO (Reset System Options), II: 2-327
- roll-out files, I: A-14
- rotational latency, III: 31-6
- RP (Ready and Purge Magnetic Tape), II: 2-329
- RPA, I: 5-3
- RPB, I: 5-3
- RQ (Remove Storage Queue Entries), II: 2-331
- RS (Remove Job from Schedule), II: 2-333
- RTSL
 - option, I: 4-48, II: 2-353
 - record, I: 5-65
- RUN (Initiate Execution of Program), II: 2-334

- Run Log (RLOG), I: 4-52
 use for file type analysis
 for QWIK Disk, III: 29-12
- RW (Rewind and Unload
 Tape), II: 2-335
- RX (Return Deleted Space), II: 2-336
- RY command
 with DMPCPY program, III: 7-1
 with PKCOPY program, III: 19-1
- RY (Ready Peripheral Device), II: 2-338
- S**
- S4A
 adding, II: 2-395
- S4B
 adding, II: 2-395
- sample development environment
 for QWIK Disk, III: 29-16
- sample disk pack environment
 for QWIK Disk, III: 29-17
- sample production environment
 for QWIK Disk, III: 29-17
- SAVED, I: 5-13, 5-32, 5-54
- SB (Interrogate Database
 Activity), II: 2-340
- SC keyword
 with DISPKV program, III: 3-8
- schedule
 PRS (Change Scheduling
 Priority), II: 2-289
 PRS (Set Schedule Priority), II: 2-301
 status, I: 5-70
- SCHM
 option, I: 4-48, II: 2-353
 record, I: 5-65
- SD command
 with LDCNTL program, III: 10-6
- SD (Set Deck Limit), II: 2-342
- secondary disk pack family, I: 4-41
- sector, III: 31-5
- SECURED file access, I: 4-57, 5-4
- security, I: 4-60, A-15
 ALLOCATE record, I: 5-2
 attribute
 NON, I: 4-56
 PRI, I: 4-56
 PUB, I: 4-56
 BEGINUSER (Mark Beginning of Job
 Stream), II: 2-45
- security (cont.)
 CHARGE (Insert Charge
 Number), II: 2-74
 charge number field, I: 4-61
 DFSC option, I: 4-61
 disk file, I: 4-61, 5-36
 ENDUSER (Mark End Of Job
 Stream), II: 2-158
 file access
 IO, I: 4-57
 OUT, I: 4-57
 SECURED, I: 4-57
 FILE (Equate File-Name With
 File-Identifier Or
 Medium), II: 2-167
 file use, I: 4-56
 guard file, I: 5-2
 guard filename, I: 4-57
 password field, I: 4-61
- SECURITY
 record, I: 5-36
- security
 sensitive data, I: 4-57
 USER (Assign Access Code), II: 2-401
 user code, I: 4-56, 5-3
 usercode field, I: 4-61
- SECURITY GUARDED file access, I: 5-4
- Security, Hostname, Connect, Label
 screen, I: 4-60
- seek time, III: 31-6
- selecting files
 for QWIK Disk, III: 29-18
- sensitive data, I: 4-57
- SENSITIVEDATA file access, I: 5-4
- sequential format, III: 31-5
- SERIAL, I: 5-55
- SET O command, I: 3-42
- set option MCP command
 used during system load
 failure, IV: 2-1
- SET OPTIONS command, I: 3-42
- setting switches
 to display status information
 for DISPKV program, III: 3-31
- SHARED, I: 5-54
 system feature
 description of, III: 30-1
- shared environment
 with a single system, III: 30-4
 with multisystems, III: 30-4

Index

- shared file operations
 - BLT contention
 - definition of, III: 30-7
 - contention
 - definition of, III: 30-7
 - contention cycle
 - definition of, III: 30-7
 - definition of, III: 30-7
 - LOCK
 - definition of, III: 30-7
 - LOCK NO CONTEND
 - definition of, III: 30-8
 - LOCK READ
 - definition of, III: 30-8
 - LOCK READ UNLOCK
 - definition of, III: 30-8
 - LOCK SEEK
 - definition of, III: 30-8
 - LOCK SEEK UNLOCK
 - definition of, III: 30-8
 - MAILBOX
 - definition of, III: 30-8
 - READ
 - definition of, III: 30-8
 - RLT
 - definition of, III: 30-8
 - SEEK
 - definition of, III: 30-10
 - SSP contention
 - definition of, III: 30-9
 - stalemate
 - definition of, III: 30-9
 - example of, III: 30-9
 - stalemate use routine
 - definition of, III: 30-9
 - UNLOCK
 - definition of, III: 30-10
 - WRITE
 - definition of, III: 30-10
 - WRITE NO UNLOCK
 - definition of, III: 30-10
- shared files
 - blocking factor for, III: 30-4
 - I/O operations for
 - chart of, III: 30-4
 - restrictions for
 - discussion of, III: 30-2
- shared system
 - DBCA (Clear SSP Address Contention), II: 2-107
 - DBSR (Produce Block Lockout Table Report), II: 2-114
- shared system (cont.)
 - DBUA (Unlock Address), II: 2-115
 - DBUL (Clear Locked Files on Shared Systems), II: 2-117
 - SHOW SHARED, II: 2-345
 - SHOW SSP shows shared system processors, II: 2-345
- shared system processor (SSP), I: 3-31
 - configuration, I: 4-36
 - firmware, I: xi
 - inhibiting a, I: 4-37
 - name, I: 5-32
 - SHOW SSP, II: 2-345
- shared system type
 - multisystem shared
 - description of, III: 30-5
 - shared disk and disk pack system
 - description of, III: 30-6
 - diagram of, III: 30-6
 - shared disk only system
 - description of, III: 30-6
 - shared disk pack only system
 - description of, III: 30-6
 - diagram of, III: 30-6
 - shared system
 - description of, III: 30-5
 - single-system shared
 - description of, III: 30-5
- shared systems
 - components
 - list of, III: 30-10
 - file I/O operations
 - rules for, III: 30-2
 - goals
 - list of, III: 30-1
 - multisystem shared disk and disk pack
 - examples of, III: 30-13
 - types
 - list of, III: 30-5
- shared systems initialization
 - discussion of, III: 30-12
 - for multiple systems
 - discussion of, III: 30-12
 - for single systems
 - discussion of, III: 30-12
- SHOW (Display System Information), II: 2-343
 - BACKUP, II: 2-344
 - BLT, II: 2-344
 - EXCHANGE, II: 2-344
 - FILE, II: 2-344
 - IRS, II: 2-344

- SHOW (cont.)
 PROCESSOR, II: 2-344
 SHARED, II: 2-345
 SSP, II: 2-345
 SYSTEM, II: 2-345
 TASK, II: 2-345
 TASKS, II: 2-346
 TRAK, II: 2-346
 VCS, II: 2-346
- SI (Set Remote Device Identification), II: 2-347
- SINGLE command
 restrictions for, III: 3-21
- single system
 shared environment for, III: 30-4
- SK option, I: 5-68
- SK (Skip Backup Records), II: 2-348
- SLOG, I: 4-52
- SMC failure
 procedure for V 500 Series, IV: 3-12
- SMC screen, I: 3-15
- SN keyword
 for DISPKV program, III: 3-6
- SNAP
 option, I: 4-48, II: 2-354
 picture, I: 4-48
 record, I: 5-65
- SNAP HALT
 recovery procedure
 for V 300/V 400 Series, IV: 4-17
- SNAP picture
 hard copy
 creating, III: 20-1
- SNPANL utility program
 description of, III: 20-1
 executing, III: 20-1
- SO command
 with DCP option, III: 1-1
 to load firmware, III: 27-2
- SO (Set System Options), II: 2-352
- SOFT SNAP stop cause
 recovery procedure
 for V 500 Series, IV: 4-35
- soft-sectored disk, III: 31-1
- SORT: intrinsic program
 control file
 creating, III: 22-6
 control parameters
 for disk sorting, III: 22-5
 for tape sorting, III: 22-4
 default media override
 description of, III: 22-6
- SORT: intrinsic program (cont.)
 description of, III: 22-1
 discussion of tape files
 with variable-length
 records, III: 22-3
 disk sorting
 default media for, III: 22-5
 media used for, III: 22-5
 work file requirements, III: 22-5
 error messages
 discussion of, III: 22-11
 non-fatal, III: 22-11
 recovery of, III: 22-11
 error messages and
 warnings, III: 22-11
 fatal error messages
 correctable, III: 22-12
 non-correctable, III: 22-12
 file specification
 requirements for, III: 22-3
 input files
 in EBCDIC code, III: 22-4
 input/output media
 requirements for, III: 22-3
 input/output requirements
 discussion of, III: 22-2
 keys
 definition of, III: 22-2
 rules for, III: 22-2
 maximum block size, III: 22-4
 memory requirements
 discussion of, III: 22-1
 list of, III: 22-1
 operating considerations
 for disk sorting, III: 22-6
 for tape sorting, III: 22-5
 output file
 restrictions, III: 22-3
 record arrangement
 by keys, III: 22-2
 record size
 guidelines for, III: 22-3
 sort requirements for, III: 22-3
 resource allocation
 for disk sorting, III: 22-6
 sequence of output files
 rules for, III: 22-4
 signed key fields
 how sorted, III: 22-2
 how treated, III: 22-2
 sort keys
 discussion of, III: 22-2

(cont.)SORT: intrinsic program

- tape sorting
 - description of, III: 22-4
 - operation of, III: 22-4
- virtual collating sequence, III: 22-10
 - functions of, III: 22-10
- work file tapes, III: 22-4
- work file units
 - requirements for, III: 22-4
- work files
 - description of, III: 22-1

sort key

- restrictions for, III: 21-3

sort performance

- with QWIK Disk, III: 29-4

SORTx intrinsic program

- bound
 - description of, III: 21-1
- description of, III: 21-1
- disk requirements for, III: 21-2
- dollar sign (\$) cards
 - description of, III: 21-6
 - list of, III: 21-6
- error messages
 - list of, III: 21-7
- hardware types
 - restrictions for, III: 21-1
- information required
 - list of, III: 21-3
- input files
 - restrictions for, III: 21-1
- input limits
 - list of, III: 21-3
- input media
 - list of, III: 21-3
- input records
 - maximum length
 - (FIBMRL), III: 21-5
- memory requirements for, III: 21-1
- minimum memory requirements
 - for, III: 21-2
- output media
 - list of, III: 21-4
- parity errors
 - list of, III: 21-9
- site-specific parameters
 - discussion of, III: 21-7
- SORT BCT**
 - definition of, III: 21-5
- SORT.. file**
 - used defined, III: 21-1
 - with user programs, III: 21-5

SORTx intrinsic program (cont.)

- SORT work file**
 - discussion of, III: 21-7
- sorting procedure
 - description of, III: 21-3
- SORT.n file**
 - use defined, III: 21-1
 - with user programs, III: 21-5
- specific site defaults
 - how to create, III: 21-5
- user-provided
 - description of, III: 21-1
- variable-length records
 - provisions for, III: 21-4
- source code files
 - B 974 ND L
 - transfer of, III: 1-1
- SP (Display/Set/Change Database Parameters), II: 2-356**
- specific characters
 - interchanging, III: 13-1
- spindles, I: A-2, III: 31-5
- SPO Log, I: 4-52
- SPO (Simulate SPO Through Card Reader), II: 2-357**
- SPOMESSAGE BCT**
 - commands invalid from, II: 1-11
 - entering commands through, II: 1-5
- SQ (Squash Disk), II: 2-359**
- SQP (Squash Pack), II: 2-360**
- SQUASH, II: 2-360**
- SQUASH status of a file, ALTER FILE**
 - changes, II: 2-28
- SRTUTL utility program**
 - ADDROUT statement**
 - description of, III: 23-9
 - syntax for, III: 23-9
 - COMPARE statement**
 - description of, III: 23-9
 - reserved words for, III: 23-10
 - specifying displacement, III: 23-10
 - specifying length of
 - field, III: 23-10
 - specifying length of
 - operand, III: 23-10
 - specifying offset of key, III: 23-10
 - syntax for, III: 23-10
 - description of, III: 23-1
 - dollar sign (\$) options
 - description of, III: 23-17
 - list of, III: 23-17
 - example of, III: 23-5

- SRTUTL utility program (cont.)
- executing
 - examples of, III: 23-12
 - procedures for, III: 23-12
 - executing without cards
 - examples of, III: 23-15
 - procedures for, III: 23-13
 - FILE statement
 - list of options, III: 23-3
 - requirements for, III: 23-2
 - syntax for, III: 23-3
 - IDENT statement
 - description of, III: 23-9
 - syntax for, III: 23-9
 - KEY statement
 - description of, III: 23-5
 - examples of, III: 23-7
 - list of options, III: 23-6
 - syntax for, III: 23-6
 - MEMORY statement
 - description of, III: 23-10
 - syntax for, III: 23-10
 - MULTIFILE statement
 - description of, III: 23-5
 - syntax for, III: 23-5
 - PARITY statement
 - description of, III: 23-11
 - syntax for, III: 23-11
 - RECORDS statement
 - description of, III: 23-11
 - syntax for, III: 23-11
 - SORT statement
 - description of, III: 23-8
 - list of options, III: 23-8
 - syntax for, III: 23-8
 - syntax for, III: 23-1
 - SS (Send Message), II: 2-363
 - SSP
 - DBCA (Clear SSP Address Contention), II: 2-107
 - DBDR (Produce SSP Content Report), II: 2-110
 - DBUA (Unlock Address), II: 2-115
 - DBUL (Clear Locked Files on Shared Systems), II: 2-117
 - SHOW SSP shows shared system processors, II: 2-345
 - SSP (*See* shared system processor)
 - SSP screen, I: 4-36
 - ST command, I: 4-48
 - ST (Suspend Program Processing), II: 2-364
 - START (Restart A Stopped Program), II: 2-365
 - State Menu
 - Debug Facility
 - description of, III: 28-20
 - list of actions, III: 28-20
 - static environment
 - file characteristics
 - for QWIK Disk analysis, III: 29-16
 - status information
 - how to display
 - for DISPKV program, III: 3-31
 - Status Menu
 - Debug Facility
 - description of, III: 28-13
 - STDSTATUS
 - option, I: 4-51
 - record, I: 5-30
 - STGO
 - option, I: 4-48, II: 2-354
 - record, I: 5-65
 - STOP command, I: 4-48
 - STOP EVENT stop cause
 - recovery procedure
 - for V 500 Series, IV: 4-33
 - STOP record, I: 5-38
 - in a configuration file, I: 6-1
 - in a stacked deck, I: 6-6
 - STOP (Suspend Program Processing), II: 2-367
 - stop user program
 - Debug Facility
 - functions, III: 28-22
 - STOPPED
 - recovery procedure
 - for V 300/V 400 Series, IV: 4-16
 - STOQ
 - option, I: 4-49
 - record, I: 5-65
 - STOQBLOCKS
 - option, I: 4-45
 - record, I: 5-27
 - STOQMINBLK
 - option, I: 4-45
 - record, I: 5-27
 - STOQNAMES
 - option, I: 4-45
 - record, I: 5-28
 - STOQSIZE
 - option, I: 4-45
 - record, I: 5-28

- stoque
 - RQ (Remove Storage Queue Entries), II: 2-331
 - WQ (Display STOQUE Count), II: 2-420
- STOQUE queue entries, I: 4-45
- SUB, I: 5-3
- subsystems
 - disk, III: 31-1
 - pack, III: 31-1
- successful load
 - message for, III: 11-2
- SV (Save a Peripheral Unit), II: 2-368
- SW (Set Programmatic Switches), II: 2-369
- switch
 - maintenance link enable, I: 1-9
 - setting
 - description of, III: 3-3
- V 300
 - CONSOLE/ON LINE, I: 1-6
 - DC OFF (I/O cabinet), I: 1-6
 - DC OFF (I/O expansion cabinet), I: 1-7
 - DC OFF (processor cabinet), I: 1-6
 - DC ON (I/O cabinet), I: 1-6
 - DC ON (I/O expansion cabinet), I: 1-7
 - DC ON (processor cabinet), I: 1-6
 - DISK ERR/TIMEOUT, I: 1-6
- V 500
 - CAB/SYS, I: 1-11
 - ENABLE, I: 1-8
 - LOCAL, I: 1-9
 - MAIN LINK ENABLE, I: 1-9
 - ON/STAND BY, I: 1-11
 - POWER OFF (processor cabinet), I: 1-9
 - POWER ON (processor cabinet), I: 1-9
 - PWR FAIL/AIR FAIL, I: 1-11
 - REMOTE, I: 1-9
- syntax errors
 - DMPANL utility program, III: 6-3
- SYST
 - extension module, I: 4-49
 - option, II: 2-354
 - record, I: 5-66
- system analysis
 - for QWIK Disk
 - overview of, III: 29-5
 - steps for performing, III: 29-6
- system commands, II: 2-1
 - access level restrictions, II: 1-12
 - entering through a SPOMESSAGE BCT, II: 1-5
 - entering through BNA, II: 1-7
 - entering through CANDE PROPS WFL, II: 1-6
 - entering through punched cards, II: 1-5
 - entering through RJE, II: 1-7
 - entering through ZIP or ZIPSPO, II: 1-4
 - entering through ZIPSPO, II: 1-5
 - general comments, II: 1-7
 - restrictions, II: 1-11
 - through the ODT, II: 1-2
 - used with LDCNTL program
 - list of, III: 10-2
 - used with QWIK Disk
 - to generate reports, III: 29-17
 - ways to enter, II: 1-1
- system configuration
 - V 500 options, I: 3-20
 - worksheet, I: ix
- system configuration file
 - adding records
 - for QWIK Disk, III: 29-27
 - AUHL option
 - with SYSUP facility, III: 25-1
 - CONNECT record
 - for shared systems, III: 30-12
 - CONTROL DEBUG MCP option
 - description of, III: 28-1
 - CONTROL DEBUG USER option
 - description of, III: 28-1
 - DISK record
 - for shared systems, III: 30-12
 - for shared systems
 - list of required records, III: 30-12
 - LIMIT BLT record
 - for shared systems, III: 30-12
 - LIMIT DELAY record
 - for shared systems, III: 30-13
 - loading the, I: 1-12
 - PACK record
 - for shared systems, III: 30-12
 - SSP DLP record
 - for shared systems, III: 30-12
 - SYSTEM record
 - for shared systems, III: 30-12
 - UNIT SHARED record
 - for shared systems, III: 30-12

- system configuration file (cont.)
 - USE record
 - with APCR option, III: 10-7
 - with AURD option, III: 10-5
 - with DCP option, III: 1-1
 - with DUMP option, III: 6-1, 6-2, 7-1
- system control panel, I: 2-2
- system environment analysis
 - for QWIK Disk
 - dynamic, III: 29-16
 - static, III: 29-16
- system error
 - output messages, IV: E-2
- system features
 - Data Communications Processor (DCP)
 - description of, III: 27-1
 - Debug Facility
 - description of, III: 28-1
 - SHARED
 - description of, III: 30-1
- system initialization, I: 1-12, 3-1
 - messages, IV: 2-1
 - from system loader, IV: 2-2
 - output messages
 - configuration record errors, IV: 2-4
 - errors, IV: 2-4
 - MCP informative, IV: 2-3
- system interchange code
 - restriction for, III: 3-7
 - SC keyword, III: 3-8
- system loader coded halts
 - during system initialization, IV: 2-1
- system maintenance
 - ALLOCATE (Create Disk Space And Directory Entries), II: 2-17
 - DBCA (Clear SSP Address Contention), II: 2-107
 - DBCL (Cancel DLP), II: 2-109
 - DBDR (Produce SSP Content Report), II: 2-110
 - DBIC (Initialize Controller Parameters), II: 2-111
 - DBMM (Set/Reset Maintenance Mode), II: 2-112
 - DBSR (Produce Block Lockout Table Report), II: 2-114
 - DBUA (Unlock Address), II: 2-115
 - DBUL (Clear Locked Files on Shared Systems), II: 2-117
 - DD (Delete DLP), II: 2-120
 - DISK (Add a Disk), II: 2-128
- system maintenance (cont.)
 - DL (Delete Peripheral Devices), II: 2-135
 - DLP (Add DLPs), II: 2-137
 - DR (Date Reset), II: 2-147
 - DX (Display DLPs Attached To An Exchange), II: 2-151
 - ET (End TRAK), II: 2-161
 - HL (Halt/Load Request), II: 2-187
 - ID (Invoke DEBUG Session), II: 2-191
 - IR (Initiate DMSII Recovery), II: 2-199
 - NW (Networking Command Prefix), II: 2-244
 - PACK (Add Pack), II: 2-262
 - QD (Query Debug), II: 2-307
 - QWIKPOOL (Set Program Overlay Buffer Pool), II: 2-310
 - QWKMEM (Set MCP Overlay Buffer Pool), II: 2-311
 - RO (Reset System Options), II: 2-327
 - RQ (Remove Storage Queue Entries), II: 2-331
 - RX (Return Deleted Space), II: 2-336
 - SD (Set Deck Limit), II: 2-342
 - SHOW SYSTEM shows system information, II: 2-345
 - SI (Set Remote Device Identification), II: 2-347
 - SO (Set System Options), II: 2-352
 - SP (Display/Set/Change Database Parameters), II: 2-356
 - SQ (Squash Disk), II: 2-359
 - SQP (Squash Pack), II: 2-360
 - TR (Time Reset), II: 2-380
 - UNIT (Add Peripheral Devices), II: 2-385
 - UR (Inhibit or Uninhibit Unit), II: 2-399
 - XC (Inhibit/Uninhibit or Display Status of a Channel), II: 2-434
 - XD (Remove Disk Segments), II: 2-436
 - XM (Remove Memory from System), II: 2-437
 - XP (Remove Pack Sectors), II: 2-438
- system messages, IV: C-2
- system performance, I: A-18
- system performance monitoring, I: 4-49
- system powered off
 - recovery procedure
 - for V 500 Series, IV: 4-22

system powered off (cont.)
 for V 300/V 400 Series, IV: 4-4
 system resource pack, III: 31-14
 system status information, I: A-16
 system status table displays, I: 5-68
 system status tables, I: A-16
 AD (Specify ODT/OCS Screen
 Parameters), II: 2-4
 system use characteristic
 with QWIK Disk, III: 29-5
 SYSTEM/COPY command, I: 4-54, 5-2
 SYSTEM/COPY utility, I: 4-47
 SYSTEM/COPY utility program, I: 4-42
 additional error possibilities, III: 24-9
 BNA file transfer
 description of, III: 24-4
 description of, III: 24-1
 duplicate files
 on disk and disk pack, III: 24-13
 error messages
 list of, III: 24-4
 error recovery, III: 24-10
 functions
 list of, III: 24-1
 ICTAPE format
 directory format, III: 24-13
 discussion of, III: 24-13
 performing a read, III: 24-15
 implementing
 maximum values for, III: 24-3
 I/O error handling
 description of, III: 24-8
 I/O error messages
 description of, III: 24-8
 format for, III: 24-8
 list of, III: 24-8
 library maintenance messages
 list of, III: 24-7
 LOADMP/PACKUP tape handling
 discussion of, III: 24-13
 missing disk packs
 discussion of, III: 24-11
 missing files
 discussion of, III: 24-11
 on disk and disk pack, III: 24-11
 on disk pack, III: 24-11
 on tape, III: 24-12
 nonlibrary tape handling
 discussion of, III: 24-13
 program flow
 description of, III: 24-2
 reliability information

SYSTEM/COPY utility program (cont.)
 block error message, III: 24-10
 discussion of, III: 24-10
 security attributes
 of files copied, III: 24-3
 TAPE format
 description of, III: 24-16
 determining
 programmatically, III: 24-18
 determining visually, III: 24-18
 performing a read, III: 24-17
 use with QWIK Disk
 example of, III: 29-31
 warning messages
 list of, III: 24-7
 SYSUP facility
 examples of, III: 25-2
 programming considerations
 for, III: 25-2
 SYSUP (*See Automatic System Recovery
 Facility*)
 SYSUP program, I: 4-48, 5-66
 SYUP
 option, I: 4-48, II: 2-354
 record, I: 5-66

T

TAB option, I: 5-68
 taken branch breakpoint
 Debug Facility
 description of, III: 28-20
 TAPE, I: 5-16
 tape
 AC (Purge And Place Tape Number In
 Label), II: 2-2
 ADD (Copy Files from Library Tape to
 Disk or Diskpack), II: 2-11
 adding tape drives, II: 2-389
 CN (Display Tape Number), II: 2-78
 configuration, I: 4-17
 drives, I: 5-53
 format, I: 5-53
 FR (Designate Final Reel), II: 2-179
 GCR, I: 5-53
 LABEL1 (Change Tape Label
 Format), II: 2-212
 labels, I: 4-61, 5-23
 MPE, I: 5-53
 MT9, I: 5-53
 MTC, I: 5-53

- tape (cont.)
 - PG (Purge Magnetic Tape), II: 2-282
 - RP (Ready and Purge Magnetic Tape), II: 2-329
 - RW (Rewind and Unload Tape), II: 2-335
 - status table display, I: 5-70
 - UL (Assign Unlabeled File), II: 2-383
- tape directories
 - displaying
 - with DMPALL program, III: 5-22
- tape drives, I: A-16
- tape sorting
 - SORT: intrinsic program
 - control parameters for, III: 22-4
 - operating considerations, III: 22-5
- tasks
 - maximum number of, I: 4-45
 - show all tasks, II: 2-346
 - show individual task, II: 2-345
- TC5 configuration, I: 4-32
- TC5 device
 - adding, II: 2-394
- TD 800, I: A-3
- TELCOM DLP, I: A-3
- teletypewriter, I: A-4
- temporary halts
 - DEBUG option, IV: 5-2
- TERM
 - command, I: 3-43
 - option, I: 4-48, II: 2-354
 - record, I: 5-66
- TEST (Mark A Test Operation), II: 2-371
- THLT
 - option, I: 4-48
 - record, I: 5-66
- TI (Display Time Statistics), II: 2-372
- TIME
 - option, I: 4-48, II: 2-354
 - record, I: 5-66
- time option for system status table
 - displays, I: 5-69
- timesharing
 - RUN (Initiate Execution of Program), II: 2-334
- timesharing tasks, I: 4-51
- TMSFENCE record, I: 5-30
- TO ALL command
 - for QWIK Disk, III: 29-28
- TO (Test Option) command, I: 3-4
- TO (Type System Options), II: 2-379
- TPLB
 - option, I: 4-48, II: 2-354
 - record, I: 5-66
- TPMK
 - option, I: 4-48, II: 2-354
 - record, I: 5-67
- TPNO
 - option, I: 4-48
 - record, I: 5-67
- TR (Time Reset), II: 2-380
- trace
 - GT (Initiate Trace), II: 2-183
 - NT (Terminate Trace), II: 2-243
- trace functions
 - using
 - in Debug Facility, III: 28-28
- Trace Menu
 - Debug Facility
 - description of, III: 28-15
 - list of actions, III: 28-15
- TRACE (Trace a Program), II: 2-381
- track, III: 31-5
- train printer, I: 5-56
 - adding, II: 2-393
- TRAK
 - module, I: 4-49
 - option, I: 4-51, II: 2-354
 - record, I: 5-67
 - utility, II: 2-161, 2-346
- TRAKBUFFER
 - option, I: 4-51
 - record, I: 5-30
- transaction system performance
 - with QWIK Disk, III: 29-4
- transfer time, III: 31-6
- translate file
 - file name of, III: 13-2
- translate table
 - description of, III: 13-1
- translation
 - MAKTRN
 - file generator program, III: 13-1
- TRMD
 - option, I: 4-48, 4-49, II: 2-354
 - record, I: 5-67
- TSMFENCE option, I: 4-51
- TWX configuration, I: 4-32
- TWX device
 - adding, II: 2-394

U

UC field, I: 4-61
 UL (Assign Unlabeled File), II: 2-383
 Uniline DLP, I: 5-40
 Uniline firmware, I: xi
 Uniline firmware load failure
 procedure for V 300/V 400
 Series, IV: 3-4
 procedure for V 500 Series, IV: 3-15
 UNIT (Add Peripheral
 Devices), II: 2-385
 unit number
 MCP, I: ix, 3-20
 MMDD, I: ix
 UNIT OCS record
 configuration file errors, IV: B-3
 UNIT record, I: 5-39
 CRD, I: 5-39
 DCP, I: 5-42
 for Uniline DLP, I: 5-40
 front-end processor (FEP), I: 5-53
 IPP, I: 5-43
 ISC, I: 5-44
 NCP, I: 5-45
 NST, I: 5-46
 OCS/ODT, I: 5-47
 PCH, I: 5-49
 PRN, I: 5-50
 S4A/S4B, I: 5-51
 tape drives, I: 5-53
 TPR, I: 5-56
 Unit Selection menu, I: 4-20
 UNITS, I: 5-28
 Universal Console DLP, I: 4-33
 UNLOAD, II: B-9
 UNLOAD (Move Library Files Between
 Media, Remove Old
 Files), II: 2-397
 UNLODV utility program
 commands
 list of, III: 26-12
 data communications options
 chart of, III: 26-6
 displaying, III: 26-3
 list of, III: 26-7
 description of, III: 26-1
 executing
 procedures for, III: 26-4
 firmware file USP3BH
 setting, III: 26-1

UNLODV utility program (cont.)
 settings with TD8, III: 26-9
 settings with U2AC, III: 26-10
 settings with U2B, III: 26-12
 firmware file UST3BH
 settings, III: 26-2
 firmware files
 discussion of, III: 26-1
 format for, III: 26-13
 for Uniline devices, III: 26-14
 system configuration records
 list of, III: 26-3
 unrestricted pack, III: 31-14, 31-20
 unsuccessful load
 message for, III: 11-2
 UP (Unlock Program), II: 2-398
 UR (Inhibit or Uninhibit Unit), II: 2-399
 USE DUMP Record
 configuration file errors, IV: B-2
 USE Options screen, I: 4-46
 USE record, I: 5-57, II: 2-352
 AFTR, I: 5-57
 APBD, I: 5-57
 APCR, I: 5-57
 AUHL, I: 5-58
 AURD, I: 5-58
 BOJ, I: 5-58
 CHRG, I: 5-58
 CLOS, I: 5-58
 COPY, I: 5-59
 DATE, I: 5-59
 DEBUG, I: 5-59
 DCOM, I: 5-59
 DCP, I: 5-59
 DMS2, I: 5-59
 DUMP, I: 5-60
 EOJ, I: 5-60
 GTDK, I: 5-61
 LIB, I: 5-61
 LOGGING, I: 5-61
 MICR, I: 5-63
 OPEN, I: 5-63
 PBD, I: 5-63
 PBP, I: 5-63
 PBPS, I: 5-63
 PBT, I: 5-63
 PCD, I: 5-64
 PCH, I: 5-64
 PCP, I: 5-64
 PCRM, I: 5-64
 PRN, I: 5-64
 QWIK, I: 5-64

USE record (cont.)

RJE, I: 5-65
 RMOV, I: 5-65
 RTSL, I: 5-65
 SCHM, I: 5-65
 SNAP, I: 5-65
 STGO, I: 5-65
 STOQ, I: 5-65
 SYST, I: 5-66
 SYUP, I: 5-66
 TERM, I: 5-66
 THLT, I: 5-66
 TIME, I: 5-66
 TPLB, I: 5-66
 TPMK, I: 5-67
 TPNO, I: 5-67
 TRAK, I: 5-67
 TRMD, I: 5-67
 with APCR option, III: 10-3
 with AURD option, III: 10-3
 with PCRM option, III: 10-3
 WRKP, I: 5-67
 ZIPM, I: 5-67
 USER (Assign Access Code), II: 2-401
 user code, II: 2-422
 security, I: 4-56, 5-3
 user Overlay
 QWIKPOOL (Set Program Overlay
 Buffer Pool), II: 2-310
 USER QWIK memory
 WO (Display Active User QWIK
 Operation Counts), II: 2-418
 user selected opcodes
 for Debug Facility
 field for, III: 28-17
 usercode field, I: 4-61
 USERFL file, I: 5-36
 USERHO program, I: 5-36
 USERLS program, I: 5-36
 USP3BV, I: 3-8
 UST2BH, I: 3-8
 utility programs
 CPLOAD, III: 2-1
 DC1ANL, III: 2-2
 DISPKV, III: 3-1
 DLPXCO, III: 4-1
 DLPXNO, III: 4-1
 DMPALL, III: 5-1, 21-5, 22-6
 DMPANL, III: 6-1
 in relation to DMPMEM, III: 8-1
 DMPCPY, III: 7-1
 DMPMEM, III: 8-1

utility programs (cont.)

ECMANL, III: 9-1
 LDCNTL, III: 10-1
 LDHOST, III: 11-1
 with Data Communications
 Processors, III: 27-2
 LOADFW, III: 12-1
 MDCOPV, III: 14-1
 NIFMRG, III: 16-1
 with Data Communications
 Processors, III: 27-1
 PBDPRN, III: 17-1
 PCOPY, III: 18-1
 PKCOPY, III: 19-1
 SNPANL, III: 20-1
 SRTUTL, III: 23-1
 SYSTEM/COPY, III: 24-1
 UNLODV, III: 26-1
 use with QWIK Disk
 to generate reports, III: 29-17
 UVSFCO program, I: 3-22

V

V Series Communication System (VCS)

 initialization file, I: 4-41
 validity of CONFIGURE for
 659 disk packs, III: 3-18
 validity of INITIALIZE for
 659 disk packs, III: 3-16
 validity of RELOCATE for
 659 disk packs, III: 3-23
 validity of REPORT for
 659 disk packs, III: 3-19
 validity of SINGLE for
 659 disk packs, III: 3-21
 value clauses
 examples of, III: 5-24, 5-25
 to start DMPALL
 from card file, III: 5-24
 from executing program, III: 5-24
 VALUE (Insert Value into a
 Program), II: 2-403
 VCS
 SHOW VCS, II: 2-346
 VDD configuration, I: 4-32
 VDD device
 adding, II: 2-394
 VER command, I: 3-24
 verify
 library tape, II: B-4

- version 1 pack, III: 31-12
- version 1 pack families, III: 31-13
- version 2 pack, III: 31-12
- virtual collating sequence
 - functions
 - list of, III: 22-10
 - SORT: intrinsic program, III: 22-10
 - translate option
 - description of, III: 22-10
 - translate tables
 - description of, III: 22-10
 - naming convention, III: 22-11
- V 300
 - cold-start, I: 3-9
 - control panel, I: 1-5
 - conventional halt/load, I: 3-30
 - executing the CONFIG utility, I: 4-5
 - initializing, I: 3-3
 - I/O processor, I: 1-4
 - I/O subsystem, I: 1-4
 - loading the CONFIG utility, I: 4-5
 - main processor, I: 1-3
 - maintenance processor, I: 1-3
 - minimum configuration, I: 3-9
 - precold-start procedure, I: 3-5
- V 500
 - clearing processor memory
 - modules, I: 3-21
 - cold-start, I: 3-25
 - conventional halt/load, I: 3-30
 - ECM Command menu, I: 2-4
 - executing the CONFIG utility, I: 4-6
 - I/O subsystem, I: 1-4
 - loading the CONFIG utility, I: 4-6
 - loading the firmware, I: 3-23
 - maintenance processor, I: 1-3
 - maintenance processor function
 - keys, I: 3-44
 - minimum peripheral
 - configuration, I: 3-19
 - multiple processor
 - cold-start, I: 3-29
 - precold-start procedure, I: 3-15
 - setting the processor offline, I: 3-22
 - system configuration options, I: 3-20
 - system control panel, I: 2-2
- V Series Communication System (VCS), III: 2-1
 - initiation, III: 2-2
 - SHOW VCS, II: 2-346

W

- waiting I/O analysis
 - use of FLAME software graphs
 - for QWIK Disk, III: 29-10
 - use of Run Log (RLOG)
 - for QWIK Disk, III: 29-10
- waiting jobs status, I: 5-70
- waiting mix, I: 5-71, II: 2-8
- warning messages
 - SYSTEM/COPY utility program
 - list of, III: 24-7
- WB (Display Bound Programs), II: 2-405
- WC (Display Available Memory), II: 2-406
- WD (Display Date), II: 2-408
- WFL
 - commands invalid through, II: 1-12
 - entering commands through, II: 1-6
 - QT (Quit Program Operation), II: 2-308
- WFL (See Work Flow Language)
- WM (Display MCP Version), II: 2-416
- WM table, I: 5-70
- WO (Display Active User QWIK Operation Counts), II: 2-418
- work files, I: 4-49
 - placing in QWIK Disk
 - procedure for, III: 29-32
- Work Flow Language (WFL), I: 5-9
- WQ (Display STOQUE Count), II: 2-420
- wrap, I: 4-53
- write operation
 - with regard to QWIK Disk, III: 29-35
- WRKP
 - option, I: 4-49, II: 2-354
 - record, I: 5-67
- WRU (Display ODT Access Code), II: 2-422
- WS (Display Jobs in Schedule), II: 2-423
- WT (Display Time), II: 2-424

X

- XC (Inhibit/Uninhibit or Display Status of a Channel), II: 2-434
- XD (Remove Disk Segments), II: 2-436
- XM (Remove Memory from System), II: 2-437

- XP command
 with PKCOPY program, III: 19-1
 XP (Remove Pack Sectors), II: 2-438
- Z**
- ZIP
 commands invalid from, II: 1-11
 entering commands through, II: 1-4
- ZIP command, I: 4-49, 5-58
- ZIP mechanism
 DMPALL utility program
 examples of, III: 5-23
- ZIPM
 option, I: 4-49, II: 2-354
 record, I: 5-67
- ZIPSP0
 commands invalid from, II: 1-11
 entering commands through, II: 1-4,
 1-5
- @<cc/u>
 output message abbreviation, IV: E-1
- @<j-s>
 output message abbreviation, IV: E-1
- @<f-id>
 output message abbreviation, IV: E-1
- @<mfid>
 output message abbreviation, IV: E-1
- @<p-n>
 output message abbreviation, IV: E-2
- <cc/u>
 output message abbreviation, IV: C-1
- <f-id>
 output message abbreviation, IV: C-1
- <hdw>
 output message abbreviation, IV: C-1
- <hh:mm>
 output message abbreviation, IV: C-1
- <j-s>
 output message abbreviation, IV: C-1
- <mfid>
 output message abbreviation, IV: C-1
- <mmss>
 output message abbreviation, IV: C-1
- <p-n>
 output message abbreviation, IV: C-2
- 100-byte mode
 initializing disk packs to
 with DISPKV program, III: 3-43
- 659 disk packs
 rules for, III: 3-34
 validity of CONFIGURE for, III: 3-18
 validity of INITIALIZE for, III: 3-16
 validity of RELOCATE for, III: 3-23
 validity of REPORT for, III: 3-19
 validity of SINGLE for, III: 3-21
- 659IVR command
 restrictions for, III: 3-29

