

**System
Software
Operation
Guide**

Volume 2

B 1000 Systems

(Relative to Mark 12.0 System Software Release)
Copyright © 1985, Burroughs Corporation, Detroit, Michigan 48232

Burroughs cannot accept any financial or other responsibilities that may be the result of your use of this information or software material, including direct, indirect, special or consequential damages. There are no warranties extended or granted by this document or software material.

You should be very careful to ensure that the use of this software material and/or information 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.

Comments or suggestions regarding this document should be submitted on a Field Communication Form (FCF) with the CLASS specified as 2 (S.SW: System Software), the Type specified as 3 (DOC), and the product specified as the 7-digit form number of the manual (for example, 1169091).

LIST OF EFFECTIVE PAGES

Page	Issue	Page	Issue
Title	Original	25-4	Blank
ii	Original	25A-1	Original
iii	Original	25A-2	Blank
iv	Blank	26-1 thru 26-5	Original
v thru xxi	Original	26-6	Blank
xxii	Blank	27-1 thru 27-8	Original
xxiii thru xxiv	Original	28-1 thru 28-7	Original
1-1 thru 1-9	Original	28-8	Blank
1-10	Blank	29-1 thru 29-6	Original
2-1 thru 2-2	Original	30-1 thru 30-8	Original
3-1	Original	31-1 thru 31-36	Original
3-2	Blank	32-1 thru 32-4	Original
4-1 thru 4-12	Original	33-1 thru 33-15	Original
5-1	Original	33-16	Blank
5-2	Blank	34-1 thru 34-7	Original
6-1 thru 6-8	Original	34-8	Blank
7-1 thru 7-5	Original	35-1 thru 35-4	Original
7-6	Blank	36-1 thru 36-2	Original
8-1 thru 8-2	Original	37-1 thru 37-8	Original
9-1 thru 9-7	Original	38-1 thru 38-4	Original
9-8	Blank	39-1 thru 39-3	Original
10-1 thru 10-6	Original	39-4	Blank
11-1 thru 11-17	Original	40-1 thru 40-2	Original
11-18	Blank	41-1	Original
12-1 thru 12-3	Original	41-2	Blank
12-4	Blank	42-1 thru 42-3	Original
13-1 thru 13-4	Original	42-4	Blank
14-1 thru 14-9	Original	43-1 thru 43-15	Original
14-10	Blank	43-16	Blank
15-1 thru 15-2	Original	44-1 thru 44-2	Original
16-1 thru 16-30	Original	45-1 thru 45-2	Original
17-1 thru 17-14	Original	46-1 thru 46-3	Original
18-1	Original	46-4	Blank
18-2	Blank	47-1 thru 47-8	Original
19-1 thru 19-26	Original	48-1 thru 48-10	Original
20-1 thru 20-14	Original	49-1 thru 49-6	Original
21-1 thru 21-6	Original	50-1 thru 50-3	Original
22-1 thru 22-51	Original	50-4	Blank
22-52	Blank	A-1 thru A-4	Original
23-1 thru 23-18	Original	B-1 thru B-3	Original
24-1 thru 24-37	Original	B-4	Blank
24-38	Blank	C-1 thru C-4	Original
25-1 thru 25-3	Original	1 thru 13	Original
		14	Blank

TABLE OF CONTENTS

Section	Title	Page
	PREFACE	xxiii
	Related Documents	xxiv
	B 1000 Software Manuals	xxiv
1	INTRODUCTION	1-1
	Categories of Utility Programs	1-1
	General Description of Utility Programs	1-3
	System Start-Up Programs	1-3
	Copy/File Copy Programs	1-3
	Disk File Copy	1-4
	Tape File Copy	1-4
	Printer Listings	1-4
	Cassette File Copy	1-6
	Card File Copy	1-6
	Diskette File Copy	1-6
	File Initializer Program	1-7
	Disk Initializer Programs	1-7
	General Maintenance Programs	1-7
	Log Analysis Programs	1-9
2	CASSETTE/MAKER	2-1
	Operating Instructions	2-1
	Input Specifications	2-1
	Sample Execution Commands	2-1
	Program Output	2-1
	Files Created	2-1
	Informative Messages	2-2
	Error Messages	2-2
3	CHECK/LOAD-DUMP	3-1
	Program Product No Longer Supported	3-1
4	CLEAR/START AND MEMORY DUMP PROCEDURE	4-1
	Clear Start Functions	4-1
	System Initialization Functions	4-2
	MCP Functions	4-2
	Clear/Start Operating Instructions	4-2
	Clear-starting B 1990 Systems	4-3
	Clear-starting Other B 1000 Systems	4-3
	Description of the Name Table	4-4
	Temporary Environment Changes	4-5
	B 1990 Systems	4-5
	Other B 1000 Systems	4-7
	System Disk Channel Selection Override	4-8
	I/O Channel Deletion	4-8
	Pseudo-Memory Size Specification	4-9
	Disk Drive Unit Order Selection	4-10
	Disk Cache Disable Function	4-11
	Supplementary Information Required with a Memory Dump	4-11

TABLE OF CONTENTS (Cont)

Section	Title	Page
5	CODE/ANALYZER	5-1
	Operating Instructions	5-1
	Program Switches	5-1
6	COLDSTART/DISK	6-1
	COLDSTART/DISK Program Overview	6-1
	Input and Output Disks	6-1
	System Software Files	6-2
	Program Options	6-2
	Operating Instructions	6-3
	Program Loading -- B 1990 Systems	6-3
	Program Loading -- All Other B 1000 Systems	6-4
	Program Execution	6-4
	Halts	6-5
	Error and Advisory Messages to the ODT	6-5
7	COLDSTART/TAPE	7-1
	Program Overview	7-1
	Input and Output Media	7-1
	System Software Files	7-2
	Program Options	7-2
	Operating Instructions	7-2
	Program Loading -- B 1990 Systems	7-2
	Program Loading -- All Other B 1000 Systems	7-3
	Program Execution	7-3
	Halts	7-4
8	CONVERT/BACKUP	8-1
	Operating Instructions	8-1
	Error Messages	8-2
9	CREATE/ISAM	9-1
	Operating Instructions	9-1
	Terminal Interface	9-1
	File Screen Semantics	9-2
	Key Screen Semantics	9-3
	ODT Interface	9-4
	ODT Interface Sample Executions	9-6
	Card Interface	9-6
	Output Messages	9-6
	Terminal Interface Messages	9-6
	ODT Interface Messages	9-7
	Error Conditions	9-7
10	CREATE/TABLE	10-1
	Translation Table	10-1
	Translate Header Format	10-1
	Operating Instructions	10-2
	Program Input	10-2
	Option Summary	10-3
	Syntax	10-3
	Semantics	10-4

TABLE OF CONTENTS (Cont)

Section	Title	Page
	Error Messages	10-5
	Internal Files	10-6
11	DC/AUDIT	11-1
	Overview	11-1
	The AUDIT Capability	11-1
	Writing The Audit File	11-1
	Restrictions	11-1
	Structure of the Audit File	11-1
	Audit File Blocking	11-2
	Audit File Buffering	11-2
	DC/AUDIT Operating Procedures	11-2
	Parameter Entry Alternatives	11-2
	Parameter Entry Format	11-3
	Parameter Entry Error Message	11-3
	Processing	11-4
	ODT Messages	11-4
	Termination	11-5
	Input Commands	11-5
	Parameters	11-5
	Printer Output Formats	11-11
	Output Format 1	11-12
	Output Format 2	11-13
	Output Format 3	11-14
	Output Format 4	11-15
	I/O Completions	11-15
	DC/AUDIT Mnemonic Characters	11-17
12	DISK/ALLOCATOR	12-1
	Operating Instructions	12-1
	Execution	12-1
	Command Specifications	12-1
	Allocated File Structure	12-2
	Sample Execution	12-2
	Error Messages	12-3
13	DISK/DUMP	13-1
	Program loading on B 1990 Systems	13-1
	Program loading on other B 1000 Systems	13-2
	Operating Instructions	13-2
	Status Command	13-3
	Error Messages	13-3
14	DISKETTE/COPY	14-1
	Operating Instructions	14-1
	Program Switches	14-1
	Input Specifications	14-1
	AUTOLOAD Facility	14-2
	COPY Command	14-2
	Routine Type Specifier	14-3

TABLE OF CONTENTS (Cont)

Section	Title	Page
	Input File Specifications	14-3
	Output File Specifications	14-4
	BYPASS Option	14-4
	PSEUDO Option	14-4
	KA Command	14-4
	PURGE Command	14-6
	RELABEL Command	14-6
	Sample Input Specifications	14-6
	Error Messages	14-7
15	DISKPACK/INTERCHANG	15-1
	Operating Instructions	15-1
	Error Messages	15-1
16	DMPALL	16-1
	Operating Instructions	16-1
	Program Switches	16-2
	Input to the DMPALL Program	16-3
	Sample Input Specification Strings	16-3
	Sample Executions	16-4
	Tape Parity Modifications	16-4
	Commands	16-5
	CAT (Concatenation) Command	16-5
	COPY (or PERFORM) Command	16-7
	LIST Command	16-10
	PD (Print Directory) Command	16-12
	Specifications	16-13
	< access-mode > Specification	16-13
	< file-id > Specification	16-14
	< file-title > Specification	16-15
	< file-name > Specification	16-16
	< hardware-type > Specification	16-17
	< options > Specification	16-18
	< output-areas-and-blocking > Specification	16-21
	< routine-type > Specification	16-22
	< hardware-type > Specification	16-24
	Files	16-25
	Output Files	16-25
	Disk Files	16-25
	Tape Files	16-25
	Printer Listings	16-25
	Card Files	16-25
	Paper Tape Files	16-25
	Internal Files	16-26
	Error Messages	16-27
17	FOREIGN/TAPECOPY	17-1
	Operating Instructions	17-1
	Input Specifications	17-3
	Error Messages	17-12
	Internal Files	17-14

TABLE OF CONTENTS (Cont)

Section	Title	Page
18	INITIALIZE/ANALYZER	18-1
	Operating Instructions	18-1
	Program Output	18-1
	Error Messages	18-1
19	LOGCONVERT	19-1
	General Information	19-1
	Operating Instructions	19-1
	Output File Format	19-1
	Error and Advisory Messages	19-2
	Record Types	19-2
	Record Format Tables	19-3
	Sample Program Record Declarations	19-13
	COBOL Record Formats	19-13
	Clear/Start	19-13
	Schedule	19-14
	Beginning-of-Job	19-14
	File Open	19-15
	File Close	19-16
	DMS Statistics	19-17
	DMS Open/Close	19-18
	Peripheral Assignment/Release	19-18
	End-of-Job	19-19
	Log Comment	19-19
	Log Transfer	19-19
	RPGII Record Formats	19-20
	Clear/Start	19-20
	Schedule	19-20
	Beginning-of-Job	19-21
	File Open	19-22
	File Close	19-23
	DMS Statistics	19-24
	DMS Open/Close	19-24
	Peripheral Assign/Release	19-24
	End-of-Job	19-25
	Log Comment	19-26
	Log Transfer	19-26
20	PACK/INIT	20-1
	Program Loading Instructions	20-2
	Program Loading on B 1990 Systems	20-2
	Program Loading on other B 1000 Systems	20-3
	Program Operating Instructions	20-3
	ODT Operation	20-4
	Card Reader Operation	20-4
	STATUS Command	20-5
	Initialization Specifications	20-6

TABLE OF CONTENTS (Cont)

Section	Title	Page
	User Interface Information	20-6
	SERIAL NUMBER	20-8
	PACK ID	20-8
	PACK TYPE	20-8
	DATE	20-8
	OWNER'S NAME	20-8
	OPTIONS	20-9
	CYLINDER NUMBER	20-9
	Initialization Integrity Protection	20-9
	Initialization Options	20-9
	REMOVE Option	20-10
	LIMIT Option	20-10
	Marginal Sectors Option	20-10
	Dollar Sign Options	20-11
	Informative Messages	20-12
	Error Messages	20-13
21	QWIKLOG	21-1
	Preliminary Setup	21-1
	LOG Option	21-1
	RMOV Option	21-1
	Operating Instructions	21-1
	Program Switches	21-2
	Work Files	21-2
	Output - Summary Listing	21-3
	Mnemonics	21-4
	Error Messages	21-5
	Program Traps	21-6
22	RD	22-1
	Introduction	22-1
	Common Syntax	22-2
	File Identifier	22-3
	Program Name	22-3
	SKIP and PAGE Keywords	22-3
	Capabilities and Requirements	22-4
	Support Capabilities	22-4
	Hardware Requirements	22-4
	Memory Requirements	22-4
	Terminal Requirements	22-5
	Software Requirements	22-5
	Installation Procedures	22-5
	Creating the (SYSTEM)/USERCODE File	22-5
	Compiling the Network Controller	22-5
	Initializing the MCP Name Table	22-6
	Creating the SMCS Jobs File	22-6
	Setting the RD Program Priority	22-7
	Setting the RD Program Switches	22-7

TABLE OF CONTENTS (Cont)

Section	Title	Page
Operating Instructions		22-8
	Execution Of The RD Program (Batch Mode)	22-8
	Execution Of The RD Program (With SMCS)	22-9
	Termination Of The RD Program	22-9
Terminal Commands		22-10
	AVL (Available Disk)	22-10
	BF (Backup Files)	22-11
	BRK (Break)	22-12
	BYE (Terminal User Sign Off)	22-12
	CP (Compute)	22-13
	DF (Date of File)	22-15
	DIR (Directory)	22-16
	EOJ (End of Job)	22-16
	FN (File Name)	22-17
	FRZ (Freeze)	22-17
	HELP (or TEACH)	22-18
	ID (Identification)	22-18
	JOBS (Program Information)	22-19
	KA	22-20
	KB (ODT Queue)	22-22
	KC	22-24
	KP	22-26
	MIX	22-28
	OL	22-31
	OMX (Offline Mix)	22-33
	PD (Print Directory)	22-34
	PQ (Print Queue)	22-35
	PRT (Print)	22-36
	QF (Query File)	22-38
	QP (Query Program)	22-40
	TEACH (See HELP Command)	22-41
	TD (Time and Date)	22-41
	TO (Test Option)	22-41
	TRANSLATE	22-42
	UF (User of File)	22-43
	UP (User of Pack)	22-44
	US (User)	22-45
	USE (Usage Information)	22-46
	W (Wait)	22-47
	WHO (Current User Info)	22-47
	WS (Waiting Schedule)	22-48
ODT Commands		22-50
	PRG (Programs)	22-50
Batch Commands		22-51
	CLS (Close)	22-51
	PAGE	22-51

TABLE OF CONTENTS (Cont)

Section	Title	Page
23	REMOTE/LISTER	23-1
	Introduction	23-1
	Operating Instructions	23-1
	Program Switches	23-2
	Command Syntax	23-3
	Program Commands	23-4
	BACK	23-6
	BYE	23-6
	CANDE	23-6
	EOJ	23-7
	ERROR	23-7
	FIND	23-8
	GET	23-8
	HEX	23-9
	MARGIN	23-10
	NEXT	23-10
	PAGE	23-11
	PRINT	23-12
	REMOVE	23-14
	SAME	23-14
	SEARCH	23-14
	SEQ	23-15
	TEACH	23-16
	TERMINAL	23-16
	USER	23-16
	WAIT	23-17
	WARNING	23-17
	WHAT	23-18
24	REMOTE/PRINT	24-1
	Overview and Features	24-2
	Program Command Summary	24-2
	Print Option Summary	24-3
	Default Options Summary	24-4
	Error Messages	24-4
	Printer Busy Retries	24-4
	Backup File Names	24-5
	File Security Considerations	24-5
	Remote Printer Control Characters	24-6
	Printer Buffer Sizes	24-7
	Channel Skips	24-7
	Network Controller Installation	24-8
	Declaration Section	24-8
	Terminal Section	24-8
	Station Section	24-9
	Line Section	24-9
	File Section	24-9
	Example Network Controller	24-10

TABLE OF CONTENTS (Cont)

Section	Title	Page
	Default Options (Attributes and Switches)	24-11
	Output Message Blocking	24-11
	Banners	24-12
	Page Width	24-12
	Page Size	24-13
	Saving/Purging of the Backup File	24-13
	Output Device	24-13
	Remote File Title	24-14
	Printer Switching	24-14
	Backup Output Translation	24-15
	ODT Output Translation	24-15
	Program Termination	24-15
	Summary of Default Program Attributes	24-16
	Operating Instructions	24-17
	ODT Execution	24-17
	WFL Execution	24-17
	Program Commands	24-18
	DEBUG	24-20
	DEVICES	24-20
	DUMP	24-20
	EOJ	24-20
	HELP	24-20
	OK	24-20
	PRINT	24-20
	SKIP	24-21
	STATUS	24-21
	STOP	24-21
	TEST	24-22
	WHAT	24-22
	Print Command Options	24-23
	Record Size	24-23
	File Specification	24-23
	Syntax Diagram	24-24
	BANNERS	24-26
	CHARSPERINCH	24-27
	DOUBLESPEACE	24-28
	EXTRACT	24-29
	FONT	24-31
	INDENT	24-32
	LINESPERINCH	24-32
	PAGESIZE	24-33
	PAGEWIDTH	24-33
	PRINTCOPIES	24-34
	SAVEFILE	24-34
	SUPPRESS	24-35
	TOPMARGIN	24-35
	UPPERCASE	24-36
	USER	24-37

TABLE OF CONTENTS (Cont)

Section	Title	Page
25	SDL/SSC	25-1
	Operating Instructions	25-1
	Example	25-3
25 A	SDL2INTRIN/REMOVED	25A-1
	Operating Instructions	25A-1
26	SSLOAD/MAKCAS	26-1
	Operating Instructions	26-1
	Program Switches	26-2
	Modes	26-3
	Creating MIL Program Cassettes	26-3
	Creating SDL Program Cassettes	26-3
	Error Correction	26-4
	Error Messages	26-4
27	STANDALONE/DISK-DUMP	27-1
	Program Loading on B 1990 Systems	27-2
	Program Loading on Other B 1000 Systems	27-2
	Program Operating Instructions	27-3
	STATUS Command	27-4
	Error Messages	27-5
	Standalone/Disk-Dump Halts	27-8
28	SYSTEM/BACKUP	28-1
	References	28-1
	Operating Instructions	28-1
	PB System Command	28-2
	File Specifier Options	28-2
	Output Options	28-3
	AB System Command	28-3
	Options	28-3
	Program Switches	28-4
	Run-Time Control Options	28-5
	Error Messages	28-6
	Files and Internal Structures	28-7
29	SYSTEM/BUILDTRAIN	29-1
	Translate Table Format	29-1
	Operating Instructions	29-1
	Program Switches	29-2
	Input Record Format	29-2
	Standard Translate Tables	29-3
30	SYSTEM/COMPARE	30-1
	Operating Instructions	30-1
	Program Switches	30-2
	Command Specifications	30-2
	Run-Time Interrogation	30-4
	Output Messages	30-5
	Fatal Errors	30-5
	Errors	30-5
	Warning Messages	30-7
	Informative Messages	30-8

TABLE OF CONTENTS (Cont)

Section	Title	Page
31	SYSTEM/COPY	31-1
	Overview	31-1
	Terminology	31-2
	File Identifiers	31-2
	SYSTEM and User Disks	31-3
	SYSTEM/COPY Program Commands	31-4
	ADD Command	31-4
	COPY Command	31-5
	COMPARE Command	31-6
	Command String Clauses	31-9
	<options> Clause	31-9
	<copy request>	31-12
	<copy-from group>	31-12
	<file-specifications> Clause	31-13
	FROM Clause, TO Clause	31-14
	DIR Command	31-18
	Operating Instructions	31-20
	Program Switches	31-20
	SYSTEM/COPY Input: Sample Strings	31-21
	User Disk to Tape	31-21
	SYSTEM Disk to Tape	31-22
	Disk to Disk	31-22
	Tape to Tape	31-22
	Operator Input (AC or AX Messages)	31-23
	TEACH Command, HELP Command	31-23
	STATUS Command	31-23
	QUIT Command	31-23
	SYSTEM/COPY Output	31-24
	SYSTEM/COPY Speed Considerations	31-24
	Error Handling	31-25
	I/O Errors	31-25
	Comparison Errors	31-26
	In-Use Files	31-26
	Missing Files	31-27
	Duplicate File Identifier Checking	31-28
	Error and Advisory Messages	31-28
	Message Output Control (Switch 0, MCP11 COPY Option)	31-28
	Syntax and Semantics Error Messages (Fatal Errors)	31-29
	Syntax and Semantics Advisory Messages	31-32
	Library Maintenance Messages	31-33
	Messages Denoting Actions	31-34
	Explanatory Messages	31-34
	Errors When Using HOSTNAME	31-35
	COMPARE Errors (Line Printer Listings)	31-36

TABLE OF CONTENTS (Cont)

Section	Title	Page
32	SYSTEM/DISK-DUMP	32-1
	Operating Instructions	32-1
	Dump Specifications	32-1
	Program Output	32-2
	Output Disk	32-2
	Informative Messages	32-2
	Error Messages	32-3
33	SYSTEM/DISK-INIT	33-1
	Program Operation Instructions	33-2
	Program Switch Settings	33-2
	ODT Operation	33-2
	Card Reader Operation	33-3
	STATUS Command	33-3
	Initialization Specifications	33-4
	User Interface Information	33-4
	Initialization Messages	33-6
	Initialization Integrity Protection	33-7
	Initialization Options	33-7
	REMOVE Option	33-8
	LIMIT Option	33-8
	Marginal Sectors Option	33-8
	Dollar Sign Option	33-9
	Program Output	33-10
	Informative Messages	33-10
	Summary Listing	33-10
	Error Messages	33-13
34	SYSTEM/ELOGOUT	34-1
	Operating Instructions	34-1
	Program Switches	34-1
	Error Message	34-2
	Output Reports	34-2
	System Information Report	34-2
	Operator Message Report	34-2
	Memory Error Report	34-3
	Cassette Error Report	34-3
	Disk Cache Error Report	34-3
	Result Code Definitions	34-3
	Unit Error Reports	34-4
	Result Descriptor Analysis	34-5
	Disk Error Address Analysis	34-5
	Power On/Off Error Rate Report	34-6
	Configuration Disk Unit Error Rate Report	34-7
	Unit Error Summary	34-7
35	SYSTEM/FILE-INIT	35-1
	Relative Files	35-1
	Initialization	35-1
	Program Execution	35-2

TABLE OF CONTENTS (Cont)

Section	Title	Page
	Examples	35-3
	Error Messages	35-3
36	SYSTEM/ICMD-INIT	36-1
	Operating Instructions	36-1
	Error Messages	36-2
37	SYSTEM/IS-MAINT	37-1
	Operating Instructions	37-1
	Input From a Data File	37-1
	Input from the ODT	37-1
	Commands	37-2
	Sample Execution	37-5
	Error Messages (Errors and Warnings)	37-6
	WARNING Messages	37-6
	ERROR Messages	37-6
	Internal Files	37-7
38	SYSTEM/ISVERIFY	38-1
	Operating Instructions	38-1
	Program Switch Setting	38-1
	Program Output	38-1
	Sample File Output	38-2
39	SYSTEM/LDCONTRL	39-1
	Operating Instructions	39-2
	General Rules	39-3
40	SYSTEM/LOAD-CAS	40-1
	Error Messages	40-2
41	SYSTEM/LOAD-DUMP	41-1
	Program Product No Longer Supported	41-1
42	SYSTEM/LOGOUT	42-1
	Operating Instructions	42-1
	Program Output	42-1
	CLEAR/START Records	42-1
	Job Information Records	42-2
	DMS Information	42-2
	File Information records	42-3
	Comment Records	42-3
43	SYSTEM/MAKEUSER	43-1
	Operating Instructions	43-1
	File Security Attributes	43-2
	Commands	43-4
	ADD Command	43-4
	CHANGE Command	43-6
	COPY Command	43-7
	CREATE Command	43-8
	DEBUG Command	43-9
	DELETE Command	43-9
	DISPLAY Command	43-10
	END/EOJ Command	43-11
	HELP Command	43-11

TABLE OF CONTENTS (Cont)

Section	Title	Page
	LIST Command	43-11
	PUNCH Command	43-13
	The PURGE Command	43-13
	Error Messages	43-14
44	SYSTEM/MARK-SEGS	44-1
	Operating Instructions	44-1
	Input Specifications	44-1
	Program Output	44-2
45	SYSTEM/ODT	45-1
	Operating Instructions	45-1
	NDL Handler (B 1990 Systems)	45-1
	KB System Command	45-1
	Files (Reference Information)	45-1
	Memory Requirements	45-2
46	SYSTEM/ODTLOGOUT	46-1
	Operating Instructions	46-1
	Switch Settings	46-2
	Error Messages	46-2
	Internal Files	46-3
	ODTLOG File Layout	46-3
47	SYSTEM/PACKCOPY	47-1
	Overview	47-1
	Synopsis of Program Operation	47-1
	Restrictions	47-1
	Other Considerations	47-1
	Operating Instructions	47-2
	Execution Examples	47-2
	Syntax Errors	47-3
	Execution Errors	47-3
	I/O Errors	47-4
	Comparison Errors	47-5
	Special Operational Considerations	47-7
48	SYSTEM/PANDA	48-1
	Required System File	48-1
	Options Keyword Glossary	48-1
	Operating Instructions	48-2
	Program Switches	48-3
	Program Execution	48-4
	Error and Advisory Messages	48-4
	MAP and CHECK Related Messages	48-4
	Other Advisory and Error Messages	48-6
	Notes On Save Option File	48-7
	User File Heading	48-10

TABLE OF CONTENTS (Cont)

Section	Title	Page
49	SYSTEM/SQUASH	49-1
	Operating Instructions	49-1
	Program Switches	49-2
	Switch 1	49-2
	Switch 2	49-2
	Integrity Check	49-2
	Run-Time Options	49-3
	Trace Functions	49-3
	Program Output	49-4
	Summary Listing	49-4
	Squashed Disk	49-4
	Error Recovery	49-4
	Error Messages	49-5
	Nonfatal errors	49-5
	Fatal Errors	49-6
50	TAPECOPY	50-1
	Operating Instructions	50-1
	Input Specifications	50-1
	Program Switches	50-2
A	DISK DEVICE CHARACTERISTICS	A-1
	Disk Cartridge Characteristics	A-1
	Physical Characteristics	A-1
	Initialization Error Limits	A-1
	Initialization Characteristics	A-1
	B 9484 (205 and 206) Disk Pack Characteristics	A-1
	Physical Characteristics	A-1
	Initialization Error Limits	A-2
	Initialization Characteristics	A-2
	B 9499 (223, 225 and 207) Disk Pack Characteristics	A-2
	Physical Characteristics	A-2
	Initialization Error Limits	A-3
	Initialization Characteristics	A-3
	B 9484-12, B 9494-5, and B 9494-10 Disk Pack Characteristics	A-3
	Physical Characteristics	A-3
	Initialization Error Limits	A-4
	Initialization Characteristics	A-4
B	SYNTAX CONVENTIONS	B-1
	Required Items	B-2
	Optional Items	B-2
	Loops	B-2
	Bridges	B-3
C	BACKUP FILE FORMAT	C-1
	Control Information	C-1
	File Parameter Block (FPB)	C-2
	Label	C-2
	Data Record	C-2
	Carriage Control Field	C-2

LIST OF ILLUSTRATIONS

Figure	Title	Page
9-1	File Screen	9-1
9-2	Key Screen	9-3
10-1	Sample Card Deck Execution	10-2
10-2	Sample Disk File Execution	10-2
10-3	Sample Deck Using All Options	10-3
11-1	Header and Record Formats, Output Format 1	11-12
11-2	Header and Record Formats, Output Format 2	11-13
11-3	Header and Record Formats, Output Format 3	11-14
11-4	Header and Record Formats, Output Format 4	11-15
31-1	Sample COMPARE Output Listing	31-8
31-2	Tape Directory Listing from DIR Command	31-19

LIST OF TABLES

Table	Title	Page
4-1	Name Table Functions, Entries, and Mnemonics	4-4
10-1	Character Sets and Record Sizes	10-1
11-1	Special Control Characters and Codes	11-17
12-1	DISK/ALLOCATOR Attribute Keywords and Default Values	12-1
16-1	DMPALL Program Switches	16-2
17-1	FOREIGN/TAPECOPY Program Switches	17-1
19-1	LOGCONVERT CLEAR/START Record Format	19-3
19-2	LOGCONVERT SCHEDULE Record Format	19-4
19-3	LOGCONVERT BOJ Record Format	19-4
19-4	LOGCONVERT File Open Record Format	19-5
19-5	LOGCONVERT File Close Record Format	19-7
19-6	LOGCONVERT DMS Statistics Record Format	19-7
19-7	LOGCONVERT DMS Open/Close Record Format	19-8
19-8	LOGCONVERT Peripheral Assignment/Release R. F. Record Format	19-8
19-9	LOGCONVERT EOJ Record Format	19-9
19-10	LOGCONVERT Log Comment Record Format	19-9
19-11	LOGCONVERT Log Transfer Record Format	19-9
19-12	LOGCONVERT Hardware Type Definitions	19-10

LIST OF TABLES (Cont)

Table	Title	Page
19-13	LOGCONVERT File Type Definitions	19-11
19-14	LOGCONVERT Access Mode Definitions	19-12
19-15	LOGCONVERT Execute Type Definitions	19-13
24-1	Remote Printer Control Characters	24-6
24-2	Printer and Network Controller Buffer Sizes	24-7
24-3	File and Program Switch Attributes	24-16
24-4	File Attributes for WFL Execution	24-18
25-1	Identifiers and Size Units of SDL Stacks	25-2
26-1	SSLOAD/MAKCAS Program Switches (Defaults = 0)	26-2
28-1	SYSTEM/BACKUP Program Switches	28-4
29-1	SYSTEM/BUILDTRAIN Program Switches	29-2
29-2	1100/1500 LPM Train Printer	29-5
29-3	400/750 LPM Train Printer	29-6
31-1	SYSTEM/COPY Program Switches	31-21
35-1	File Attributes	35-2
43-1	SYSTEM/MAKEUSER Security Attributes	43-2
46-1	ODTLOG File Layout	46-3
48-1	SYSTEM/PANDA Program Switches 1-5	48-3
48-2	FILE-TYPE Codes	48-9
50-1	TAPECOPY Program Switches	50-3
C-1	Control Information Record	C-1
C-2	TYPE Field Definition	C-2
C-3	Values for Printer Spacing/Skipping	C-3
C-4	Values for Punch Stacker-Selection	C-3
C-5	Some Carriage Control Field Values	C-4

PREFACE

The B 1000 Systems System Software Operation Guide (SOG), in two volumes, describes the operator interface to Burroughs B 1000 computer systems.

Volume 1 (form 1169000) provides an introduction to the B 1000 system. It includes an overview of the system control instructions (program control instructions and system commands) that are recognized by the Master Control Program II (MCP II). Also included is information on initializing the system, network controller operations, output messages, and halts. Appendixes to the manual provide background information on memory management, performance monitoring, disk file access, and file security.

Volume 2 (form 1169091) is devoted to the system utility programs that are available to system users, and includes an overview as well as detailed descriptions and operating instructions for these programs. The overview is presented in section 1 (Introduction) and the programs are described individually in the sections that follow. The sequence of the sections is alphabetical, according to the following list of program names.

CASSETTE/MAKER	SSLOAD/MAKCAS
CLEAR/START	STANDALONE/DISK-DUMP
CODE/ANALYZER	SYSTEM/BACKUP
COLDSTART/DISK	SYSTEM/BUILDTRAIN
COLDSTART/TAPE	SYSTEM/COMPARE
CONVERT/BACKUP	SYSTEM/COPY
CREATE/ISAM	SYSTEM/DISK-DUMP
CREATE/TABLE	SYSTEM/DISK-INIT
DC/AUDIT	SYSTEM/ELOGOUT
DISK/ALLOCATOR	SYSTEM/FILE-INIT
DISK/DUMP	SYSTEM/ICMD-INIT
DISKETTE/COPY	SYSTEM/IS-MAINT
DISKPACK/INTERCHANG	SYSTEM/ISVERIFY
DMPALL	SYSTEM/LDCTRL
DUMP/ANALYZER	SYSTEM/LOAD-CAS
FOREIGN/TAPECOPY	SYSTEM/LOGOUT
INITIALIZE/ANALYZER	SYSTEM/MAKEUSER
LOGCONVERT	SYSTEM/MARK-SEGS
PACK/INIT	SYSTEM/ODT
QUIKLOG	SYSTEM/ODTLOGOUT
RD	SYSTEM/PACKCOPY
REMOTE/LISTER	SYSTEM/PANDA
REMOTE/PRINT	SYSTEM/SQUASH
SDL/SSC	TAPECOPY
SDL2INTRIN/REMOVER	

Volume 2 also includes three appendixes:

Appendix A, Disk Device Characteristics, provides physical characteristics and initialization information on disk cartridge and disk pack units.

Appendix B, Syntax Conventions, provides a definitive description of the use of railroad syntax.

Appendix C, Backup File Format, provides details on the format of backup files produced by the MCP.

RELATED DOCUMENTS

The following documents are referenced in this document:

B 1000 Software Manuals

B 1000 Systems System Software Operation Guide, Volume 1, form number 1169000.

SECTION 1 INTRODUCTION

The system utility programs for Burroughs B 1000 systems are designed to facilitate efficient and consistent operation of any B 1000 system. In this volume, these programs are categorized and then individually documented.

CATEGORIES OF UTILITY PROGRAMS

Each of the B 1000 system utility programs falls into one of two broad categories:

1. Normal-state utility programs, which are executed under MCPPII control.
2. Stand-alone utility programs, which are loaded from a cassette and require the MCPPII to be inoperative when they are executed.

Each utility program, whether normal-state or stand-alone, may be further categorized according to function. Awareness of these functional categories helps an operator to find the most suitable utility program for a given task. The fact that many of these utility programs can accomplish a number of tasks leads to some overlap among the functional categories; this presents no problems in practice.

Following are the functional categories:

- System Start-up Programs
- Copy/File Copy Programs
 - Disk File Copy
 - Tape File Copy
 - Printer Listings
 - Cassette File Copy
 - Card File Copy
 - Diskette File Copy
 - File Initializer
- Disk Initializer Programs
- General Maintenance Programs
- Log Analysis Programs

First the normal-state and then the stand-alone utility programs are listed in alphabetical order below, along with the major functional category of each program. The subsection that follows contains brief descriptions of each program, organized within functional category.

Normal-State Program Name	Classification
CASSETTE/MAKER	Copy/File Copy
CODE/ANALYZER	General Maintenance
CONVERT/BACKUP	Copy/File Copy
CREATE/ISAM	General Maintenance
CREATE/TABLE	General Maintenance
DC/AUDIT	Log Analysis
DISK/ALLOCATOR	General Maintenance
DISKETTE/COPY	Copy/File Copy
DISKPACK/INTERCHANG	General Maintenance
DMPALL	Copy/File Copy
FOREIGN/TAPECOPY	Copy/File Copy
INITIALIZE/ANALYZER	General Maintenance
LOGCONVERT	Log Analysis
QWIKLOG	Log Analysis
RD	General Maintenance
REMOTE LISTER	Copy/File Copy
REMOTE/PRINT	Copy/File Copy
SDL/SSC	General Maintenance
SDL2INTRIN/REMOVER	General Maintenance
SSLOAD/MAKCAS	Copy/File Copy
SYSTEM/BACKUP	Copy/File Copy
SYSTEM/BUILDTRAIN	Start-up
SYSTEM/COMPARE	General Maintenance
SYSTEM/COPY	Copy/File Copy
SYSTEM/DISK-DUMP	Copy/File Copy
SYSTEM/DISK-INIT	Disk Initializer
SYSTEM/ELOGOUT	Log Analysis
SYSTEM/FILE-INIT	File Initializer
SYSTEM/ICMD-INIT	Disk Initializer
SYSTEM/IS-MAINT	General Maintenance
SYSTEM/ISVERIFY	General Maintenance
SYSTEM/LDCONTRL	Copy/File Copy
SYSTEM/LOAD-CAS	Copy/File Copy
SYSTEM/LOGOUT	Log Analysis
SYSTEM/MAKEUSER	Start-up
SYSTEM/MARK-SEGS	General Maintenance
SYSTEM/ODT	General Maintenance
SYSTEM/ODTLOGOUT	Log Analysis
SYSTEM/PACKCOPY	Streamer Tape Copy
SYSTEM/PANDA	General Maintenance
SYSTEM/SQUASH	General Maintenance
* TABS	Log Analysis
TAPECOPY	Copy/File Copy

* Title included for information only; for details, see *B 1000 Systems Time Analysis and Billing System (TABS) Reference Manual*.

Stand-Alone Program Name	Classification
* CASSETTE/LOADER	Copy/File Copy
CLEAR/START	Start-up
COLDSTART/TAPE	Start-up
COLDSTART/DISK	Start-up
DISK/DUMP	Copy/File Copy
PACK/INIT	Disk Initializer
STANDALONE/DISK-DUMP	Copy/File Copy

* Not separately documented because it is not explicitly executed. For information, see the section titled SSLOAD/MAKCAS in this volume.

GENERAL DESCRIPTION OF UTILITY PROGRAMS

The subsections that follow give general descriptions of all the programs in each functional classification.

System Start-Up Programs

The System Start-Up utility programs bring the system and the MCP II to an operable state. A brief description of each program follows.

CLEAR/START

The CLEAR/START program initiates the MCP II and gives the MCP II control of the system.

COLDSTART/DISK

The COLDSTART/DISK program initially loads system software and constructs required tables. The software source medium is a disk pack.

COLDSTART/TAPE

The COLDSTART/TAPE program initially loads system software and constructs required tables. The software source medium is a library (COPY) tape labeled SYSTEM.

SYSTEM/BUILDTRAIN

The SYSTEM/BUILDTRAIN program creates translate tables used by line printers. For additional information, refer to the LT system command in section 5 of volume 1 of the *B 1000 Systems System Software Operation Guide*.

SYSTEM/MAKEUSER

The SYSTEM/MAKEUSER program supports the file (SYSTEM)/USERCODE which is required for file security. File security is optional.

NOTE

All removable disk packs and disk cartridges must be initialized prior to use. Refer to Disk Initializer Programs in this section for additional information.

Copy/File Copy Programs

Copy/File Copy utility programs reproduce files. Files can be copied exactly, altered during the copy, or moved from one medium to another.

Disk File Copy

Disk dumping implies the copying of some or all the files on one disk to another disk or to a tape. There are both normal-state and stand-alone utility programs that can dump a disk.

Single-file copying implies the copying of specified individual files or file families. All single-file copy programs are normal-state utility programs.

Brief descriptions of each disk file copy program follow.

CREATE/ISAM

The CREATE/ISAM program generates a COBOL74 type ISAM file from a data file or the data portion of an existing ISAM file. In the latter case, a relative read of the data file is performed, which results in a newly created ISAM file that does not include the records that were marked for deletion.

CREATE/TABLE

The CREATE/TABLE program generates a translation table for use by other programs when translation from one character set to another is needed. The translation table, in the form of a disk file, translates forward and backward from ASCII 8-bit, ASCII 7-bit, BCL, BCL 8-bit, and B500. All translation is to or from internal 8-bit EBCDIC. In addition, any other type of translation can be specified by the operator.

DISK/DUMP

The DISK/DUMP program performs a sector-by-sector copy.

DMPALL

The DMPALL program is a generalized media conversion program that lists disk file contents, copies data files between hardware devices, and also can move disk files from one pack to another.

STANDALONE/DISK-DUMP

The STANDALONE/DISK-DUMP program performs a file-by-file copy that results in a squashed output disk. This program can also copy a disk with missing disk areas to a disk with no missing disk areas.

SYSTEM/COPY

The SYSTEM/COPY program is a generalized library maintenance program that copies files between tape and disk, disk and disk, and tape and tape. (Tape functions are listed in the subsection titled Tape File Copy.)

SYSTEM/PACKCOPY

The SYSTEM/PACKCOPY program is a normal-state utility program that drives a B 9498 PE tape unit in the 100-ips mode for fast backup of user disk packs. It produces a tape that is in the same format as a SYSTEM/COPY tape.

SYSTEM/DISK-DUMP

The SYSTEM/DISK-DUMP program performs either a file-by-file or a sector-by-sector copy.

Tape File Copy

The B 1000 computer system reads library (COPY) tapes and unlabeled tapes; both 9-track and 7-track tapes can be read. A brief description of each tape file copy program follows.

DMPALL

The DMPALL program is a generalized media conversion routine. It prints the directory of Load Dump library tapes, copies a file to or from tape, handles 7-track tape conversions with even or odd parity, and handles translation to or from EBCDIC code.

FOREIGN/TAPECOPY

The FOREIGN/TAPECOPY program copies or lists any 7-track or 9-track magnetic tape to disk, tape or printer.

SYSTEM/COPY

The SYSTEM/COPY program is a generalized library maintenance program. It prints the directory of COPY library tapes, copies files to or from tape, and can handle 7-track tape conversions. Handles translation to or from the EBCDIC code recognized by the system.

TAPECOPY

The TAPECOPY program duplicates, merges, compares, and concatenates non-library multifile or single-file tapes.

Printer Listings

A line printer listing of the contents of any file can be generated. When a line printer is not available to a printer file, a printer backup file is created. The operator can request that a printer backup file be created for specific files. Line printer listings can be generated programmatically without using either the DMPALL or SYSTEM/BACKUP utility programs. The following utility programs produce printer listings of file contents.

CONVERT/BACKUP

The CONVERT/BACKUP program converts B 1000 printer backup files into a format acceptable to the B 9270 Page Printer. The converted files are in the B 6000/B 7000 printer backup format.

DMPALL

The DMPALL program lists any file in alphanumeric, numeric, or hexadecimal format.

REMOTE/LISTER

The REMOTE/LISTER program displays disk files on a screen terminal such as a TD 830, TD 820, or MT 983 terminal. The program displays data files and printer backup files, scans printer backup compiler output listings for error messages and warnings, and examines printer backup listings of MCP dumps produced by the MCP/ANALYZER program.

REMOTE/PRINT

The REMOTE/PRINT program is designed to read a B 1000 printer backup file and convert the records to a format that is acceptable to any of a number of types of remote printers as well as the system line printer. The printer backup files must have been created by the Mark 9.0 or later system software release.

SYSTEM/BACKUP

The SYSTEM/BACKUP program lists printer backup files. Primary operation is through the PB system command. For further information on printing backup files, refer to the AB and PB system commands in section 5 of volume 1 of the *B 1000 Systems System Software Operation Guide*.

Cassette File Copy

The cassette is the system input medium for stand-alone operations. On some systems, cassettes can also be used as peripheral input/output devices. The programs used to create and to read cassettes are described in the following paragraphs.

CASSETTE/LOADER

The CASSETTE/LOADER program is a stand-alone program that is loaded to a cassette by the SSLOAD/MAKCAS program. It is a bootstrap loader for the remainder of the cassette.

CASSETTE/MAKER

The CASSETTE/MAKER program writes files to a cassette. The cassette system must be in the input/output mode.

SSLOAD/MAKCAS

The SSLOAD/MAKCAS program creates stand-alone utility cassettes.

SYSTEM/LOAD-CAS

The SYSTEM/LOAD-CAS program loads files from a cassette to disk for access by the system.

Card File Copy

The punched card is both an input and an output medium for the B 1000 system. The following programs are used to read and punch card files.

DMPALL

The DMPALL program copies files to or from cards.

SYSTEM/BACKUP

The SYSTEM/BACKUP program enables the printing and punching of backup files and the printing of WFL Job Logs and the backup files associated with them.

SYSTEM/LDCONTRL

The SYSTEM/LDCONTRL program loads a card deck into a special disk file called a pseudo file and named DECK/<#>. This pseudo file is read when a pseudo reader is activated with the RN system command. Primary operation is through the LD system command.

Diskette File Copy

The industry-compatible mini-disk (diskette or floppy-disk) is both an input and output medium for the B 1000 computer system. All access to data on mini-disks is through the utility program DISKETTE/COPY. Mini-disks are initialized with the utility program SYSTEM/ICMD-INIT.

DISKETTE/COPY

The DISKETTE/COPY program copies files to or from industry compatible mini-disks (diskettes). The PSR option provides automatic loading of pseudo reader files stored on the diskette. The program also purges diskettes, generates a KA listing of files on a diskette, and relabels diskettes.

FILE INITIALIZER PROGRAM

SYSTEM/FILE-INIT

The SYSTEM/FILE-INIT program is used to allocate and initialize a relative file for COBOL74 programs.

Disk Initializer Programs

Disk packs and disk cartridges must be initialized prior to their use on the B 1000 computer system. During initialization, disks are assigned a pack or cartridge type, all sectors are verified to be readable, and the disk is written with an initialization pattern. Industry compatible mini-disks must be initialized with the utility program SYSTEM/ICMD-INIT. The following programs initialize disk devices.

PACK/INIT

The PACK/INIT program initializes, verifies, reconfigures, and relocates sectors on disk packs. Allowable types are System and User (Unrestricted) packs.

SYSTEM/DISK-INIT

The SYSTEM/DISK-INIT program initializes, verifies, reconfigures and relocates sectors on disk packs and disk cartridges. Allowable types are System and User (Unrestricted) packs. In addition, removable disk packs can be initialized as Interchange packs. These are packs that contain data that can be exchanged among Burroughs small, medium, and large systems.

SYSTEM/ICMD-INIT

The SYSTEM/ICMD-INIT program initializes industry compatible mini-disks (diskettes).

General Maintenance Programs

The utility programs listed in this subsection provide access to disk file information and provide standard, useful system maintenance functions.

CODE/ANALYZER

The CODE/ANALYZER program generates a line printer listing of program code segments, and internal file identifiers and attributes.

DISK/ALLOCATOR

The DISK/ALLOCATOR program creates a file located at a specific disk address.

DISKPACK/INTERCHANG

The DISKPACK/INTERCHANG program copies and translates data to or from an interchange pack. Interchange packs allow data to be exchanged among Burroughs small, medium, and large systems.

INITIALIZE/ANALYZER

The INITIALIZE/ANALYZER program analyzes an initialized disk and provides a printer listing of relocated and removed sectors and of pertinent label information.

RD

The RD program allows an operator of a remote display terminal to obtain system status information. The information available includes file and program attributes and current system configuration. Inquiry is made by using the standard B 1000 system commands.

SDL/SSC

The SDL/SSC (SDL stack size changer) is an SDL support program that enables an operator to display or change the sizes of the various stacks in an SDL/UPL program code file without recompiling the program. Because the SDL/SSC program deals exclusively with the scratchpad format expected by the SDL interpreter, it cannot be used to display or modify the scratchpad of any programs other than those compiled by the SDL or UPL compilers.

SDL2INTRIN/REMOVER

The SDL2INTRIN/REMOVER program is a special purpose utility program that is used during the changing of the SDL2 Intrinsic file.

STANDALONE/DISK-DUMP

The STANDALONE/DISK-DUMP program performs a file-by-file copy of a disk. This results in a squashed disk. The STANDALONE/DISK-DUMP program can also be used to copy a disk with missing disk areas to a disk with no missing areas.

SYSTEM/COMPARE

The SYSTEM/COMPARE program compares files and flags the differences between them.

SYSTEM/IS-MAINT

The SYSTEM/IS-MAINT program performs library functions on index-sequential files.

SYSTEM/ISVERIFY

The SYSTEM/ISVERIFY program tests the integrity of an indexed sequential (ISAM) file. The program lists information from the global (cluster) file, optionally lists information from the index file(s), verifies the mapping of index files onto data files, verifies cluster file information, and validates the overall structure of the index files. The ISAM file is unlocked and may be opened for input and output if it passes all the verification procedures. Verification errors result in a warning that unpredictable results can occur if the file is opened for update without first rebuilding the index files.

SYSTEM/MARK-SEGS

The SYSTEM/MARK-SEGS program allows the operator to mark code segments as important for use by the System Memory Management algorithms.

SYSTEM/ODT

The SYSTEM/ODT program handles all I/O operations on the Operator Display Terminal (ODT).

SYSTEM/PACKCOPY

SYSTEM/PACKCOPY is a normal-state utility program that is used to drive a B 9498 PE tape unit in the 100-ips mode for fast backup of user disk packs.

SYSTEM/PANDA

The SYSTEM/PANDA program provides a listing of the status and the integrity of disk areas.

SYSTEM/SQUASH

The SYSTEM/SQUASH program moves files and consolidates available space on disks.

Log Analysis Programs

The system optionally maintains three logfiles (ODTLOG, SYSTEM/LOG, and ELOG) which record all system activity. The following programs access these logfiles.

DC/AUDIT

The primary function of the DC/AUDIT program is to read an audit file and to print its contents in a readable format. The audit file may be one created by the network controller or by a data communications program (for example, RJE or HASP) that performs its own datacomm input and output functions.

LOGCONVERT

The LOGCONVERT program creates a copy of a system log file in a packed COBOL-readable and RPGII-readable format. Creates file NEWLOG/<#>.

QWIKLOG

The QWIKLOG program provides a compact analysis of a system log file. The report can be restricted to specified job numbers. A graph and summary report can be produced.

SYSTEM/ELOGOUT

The SYSTEM/ELOGOUT program analyzes the error log, which contains hardware failure information. Primary operation is through the ET system command. Refer to section 5 of volume 1 of the *B 1000 Systems System Software Operation Guide* for additional information.

SYSTEM/LOGOUT

The SYSTEM/LOGOUT program analyzes the system log in complete detail. Primary operation is through the LG system command. Refer to section 5 of volume 1 of the *B 1000 Systems System Software Operation Guide* for additional information.

SYSTEM/ODTLOGOUT

The SYSTEM/ODTLOGOUT program transfers, formats, and prints a log of ODT (operator display terminal) messages. Primary operation is through the LG system command.

TABS package

The TABS package analyzes the log files and provides system use and billing statistics. Refer to the *B 1000 Systems Time Analysis and Billing System (TABS) Reference Manual* for additional information.

SECTION 2

CASSETTE/MAKER

The CASSETTE/MAKER program is a normal-state utility program that writes files to a cassette in a format that can be read by the SYSTEM/LOAD-CAS utility program. Cassettes can be created only by a B 1800 system that has a magnetic tape cassette I/O control or by a standalone cassette drive on a B 1900 system.

OPERATING INSTRUCTIONS

The CASSETTE/MAKER program is executed from a card reader or from the ODT. In either case, the program expects to receive input specifications from the card file labeled CARDS.

INPUT SPECIFICATIONS

The optional control keyword VERIFY requests verification of each cassette to be performed immediately after the cassette is written. If verification is desired, the first input specification must be a VERIFY control card. The following is the syntax of this control option. (The dollar sign (\$) character must appear in position 1 of the record.)

Syntax:

— \$VERIFY —————

The specification strings in the input file, CARDS, contain the file-identifiers of the files to be written to the cassette, one per card, in columns 1 through 80.

SAMPLE EXECUTION COMMANDS

The following set of sample execution commands creates a cassette containing the files RPGII, PAYROLL/APPLICATION, and USER/TEST/FILE. If necessary, the files are continued on additional cassettes. Each cassette is verified after it is written.

```
?EX CASSETTE/MAKER
?DATA CARDS
$V
RPG
PAYROLL/APPLICATION
USER/TEST/FILE
?END
```

PROGRAM OUTPUT

Program output consists of files written to a cassette as well as informative messages and error messages as required.

Files Created

The CASSETTE/MAKER program writes files to a scratch cassette tape. Several files can be written to a single cassette. If necessary, additional scratch cassettes are used to hold continuation files.

Informative Messages

After processing each cassette, the CASSETTE/MAKER program issues the following message on the Operator Display Terminal (ODT):

VERIFICATION COMPLETE ON REEL <n>
CSx LOCKED [<serial number>]

Error Messages

<file-identifier> NOT ON DISK
No files are written to the cassette.

DISK PARITY ERROR – UNABLE TO VERIFY
The temporary disk file used for verification has a parity error.

VERIFY ERROR REC# <record number>
An error was encountered during verification of the cassette.

WOULD YOU LIKE TO REMAKE THIS CASSETTE? <YES OR NO>
The cassette has a verification error at the location specified in the VERIFY ERROR message. A YES response to this inquiry causes the CASSETTE/MAKER program to remake the cassette in error. A NO response causes CASSETTE/MAKER to continue, ignoring the cassette in error.

INVALID REMAKE – DISK PARITY ERROR
The temporary disk file used to remake the cassette contains a parity error.

SECTION 3 CHECK/LOAD-DUMP

PROGRAM PRODUCT NO LONGER SUPPORTED

As of the Mark 12.0 software release, the CHECK/LOAD-DUMP program is no longer supported by the B 1000 system software.

SECTION 4

CLEAR/START AND MEMORY DUMP PROCEDURE

The CLEAR/START program is a stand-alone utility that is used to bring a B 1000 system to an operable state. The process has three parts: (1) bootstrapping: loading the CLEAR/START program from the cassette into system memory, (2) clear/starting, performed by the CLEAR/START program itself, and (3) system initialization, performed by the SYSTEM/INIT program. This program is brought into memory from disk at the appropriate time by the CLEAR/START program.

Execution of the CLEAR/START program is required after any of the following events:

- Execution of the COLDSTART/DISK or COLDSTART/TAPE programs.
- A system power-up; to start a day, for example.
- A system failure that results in an irrecoverable system loop or system halt.
- Execution of another stand-alone program such as PACK/INIT, DISK/DUMP, or STANDALONE/DISK-DUMP.
- Certain of the system software changes that can be specified by the CM system command.
- Setting or resetting of any of the following options: BRGR, FLMP, MPRI, LOG, REAC, THR, TOUT, VLCP, VLIO.
- A change in the size of the Interpreter Dictionary specified by the IC system command.

CLEAR START FUNCTIONS

The CLEAR/START program performs the following functions:

1. Dumps a copy of main memory to the disk file SYSTEM/DUMPFIL~~E~~ when a memory dump is requested.
2. Clears main memory, writing zeros and correct parity throughout.
3. Scans the I/O subsystem to locate the system disk.
4. Saves certain registers and toggles that specify temporary environment changes for the SYSTEM/INIT (system initializer) program.
5. Loads the SYSTEM/INIT program into memory from the system disk.
6. Turns control over to the SYSTEM/INIT program.

SYSTEM INITIALIZATION FUNCTIONS

After receiving control from the CLEAR/START program, the SYSTEM/INIT program performs the following functions:

1. Supplements the information in SYSTEM/DUMPFIL, when a memory dump is requested, by adding (copying) certain MCP II structures such as the Name Table and the ODT Queue.
2. Allocates space for the initial structures in memory required for MCP operation, such as the Interpreter Dictionary and the MCP Stacks.
3. Loads the GISMO program into memory from the system disk, builds the Interpreter Dictionary entry for GISMO, and discards all GISMO segments not required because of the system hardware configuration, or because certain MCP options (such as MPRI) are not set.
4. Loads segment zero (non-overlayable) and segment one (overlayable initialization routines) of the MCP II into memory from the system disk, and sets up the initial memory link structures required.
5. Loads segment zero of the SDL2 Interpreter into memory from the system disk and builds the required Interpreter Dictionary entry.
6. Turns control over to the MCP.

MCP FUNCTIONS

When the MCP receives control from the SYSTEM/INIT program, a number of operations are performed before the system is actually ready to begin program execution. The MCP performs the following routines as part of system initialization:

1. Tests the I/O subsystem to determine the system configuration. The Input/Output Assignment Table (IOAT), constructed from this information, contains entries that describe the characteristics of all peripheral units present on the system.
2. Constructs the initial chain of DISK I/O descriptors, as well as TEST I/O descriptors for all other peripheral units present on the system.
3. Restores entries in the Temporary Disk Available Table to the Working Available Table.
4. Reads the disk directory and every disk file header (DFH) present on the system disk, clearing the user count in any DFH that was in use prior to the CLEAR/START. Disk files marked as TEMPORARY (such as compile and go code files) are removed from the disk directory.
5. Places entries identifying the CLEAR/START in the SYSTEM/LOG (if the LOG option is set) and the SYSTEM/ELOG.
6. Initiates the SYSTEM/ODT program and, if the AMCS option is set, the MCS program.
7. Displays the CLEAR/START message on the operator display terminal (ODT), and then makes the system available for use.

CLEAR/START OPERATING INSTRUCTIONS

Two sets of procedures for beginning the CLEAR/START operation follow. The first is for B 1990 systems and the second is for all other B 1000 systems.

Clear-starting B 1990 Systems

To begin a clear/start for B 1990 systems:

1. Halt the system by entering the HALT system command. The RUN A light goes out in a single-processor (B 1990SP) system. Both RUN A and RUN B go out in a dual-processor (B 1990DP) system.

NOTE

Instructions for using the B 1990 Soft Panel in this fashion are hereafter presented as follows:

<keyword-1> <keyword-2> ... [XMT]

A looping condition could deny access to the ODT. In such cases, press the INTRPT push button. If the system still does not halt, press the HALT/RUN push button.

2. Place the CLEAR/START cassette into the cassette reader and observe that the tape rewinds to BOT. (Blue light on the drive unit comes on; BOT appears on the ODT. If rewind does not complete, enter REWIND [XMT] from the keyboard.
3. Enter any temporary environment changes (such as a memory dump request) with the appropriate TEXT Soft Panel command. For details, refer to the subsection titled Temporary Environment Changes.
4. CLEAR MTR GO [XMT]

The clear/start operation automatically rewinds the cassette on B 1990 systems.

Clear-starting Other B 1000 Systems

To begin a clear/start for all other B 1000 systems:

1. Halt the system using either the INTRPT push button or the console INTERRUPT switch. If the INTERRUPT switch is used, return it to its initial position following the halt. If the system continues to run, press the HALT push button. If the system still continues to run, press the HALT and CLEAR push buttons simultaneously. For dual-processor systems be sure both processors are halted (both RUN lights are out).
2. Place the CLEAR/START cassette into the cassette drive and ensure that the tape rewinds to BOT.
3. Press (1) the CLEAR push button, (2) the MODE push button to obtain the MTR mode, and (3) the START push button.
4. After the bootstrap loader is read from the cassette tape, the processor halts. The L register must contain @AAAAAA@ at this time; if not, the cassette must be rewound and the procedure restarted from step 3, above.
5. Any temporary environment changes to be made (such as a memory dump request) must now be entered into the appropriate registers now. For details, refer to the subsection titled Temporary Environment Changes.
6. Press the MODE push button to enter the NORMAL mode; then press the START push button. Reading of the cassette resumes. When the CLEAR/START program is completely loaded, it takes control and executes through the system initialization procedure to completion.
7. Rewind the cassette tape by pressing REWIND. (Rewind is automatic in B 1990 systems.)

DESCRIPTION OF THE NAME TABLE

The Name Table is a structure that is initialized on a system disk by the COLDSTART/TAPE or COLDSTART/DISK programs. The Name Table contains entries identifying the names of system firmware and software available for use in the operational environment of the system. Name Table functions are shown in table 4-1.

Software and firmware routines are selected by the CLEAR/START and SYSTEM/INIT programs from entries in the Name Table. This is based on the system's hardware configuration as well as on certain optional specifications that can be entered by the system operator. Normally, it is unnecessary for the system operator to enter any specifications for the CLEAR/START program. Changes to the default selections are usually made only by Burroughs software development and support personnel in the course of system design and debugging, and are not required during normal system operation.

The Name Table design allows certain system software or firmware routines to be identified as "experimental" without affecting those routines identified as "standard". This permits a CLEAR/START to be performed with non-standard or untested routines by Burroughs software development personnel without the danger of not being able to recover to the "standard" software in an irrecoverable failure occurs in an experimental routine. Such a design also has application in a normal operating environment: backup copies of system software or firmware can be created and identified as experimental in the Name Table to enable recovery in the event the standard routines are corrupted.

The Name Table itself consists of 19 entries, each identified by an entry number and a mnemonic. The entry number is used by the CLEAR/START and SYSTEM/INIT programs and the mnemonic is used by the MCP and the system operator. Each entry contains the file identifier of the code file on the system disk. Each Name Table entry also has a specific defined function (for example, MCP or GISMO) and a condition (standard or experimental) for its selection by the CLEAR/START routine. These are shown in table 4-1.

Table 4-1. Name Table Functions, Entries, and Mnemonics

Name Table Item	Type	Entry #	Mnemonic
System Initializer	– Standard	0	N
System Initializer	– Experimental	1	NX
GISMO	– Standard	2	G
GISMO	– Experimental	3	GX
SDL2 Interpreter	– Standard	4	I
SDL2 Interpreter	– Experimental	5	IX
MCPII	– Standard	6	M
MCPII	– Experimental	7	MX
Micro-MCP	– Standard	8	MM
Micro-MCP	– Experimental	9	MMX
Network Controller	– Standard	10	C
Network Controller	– Experimental	11	CX
MCS Program	– Standard	12	MCS
MCS Program	– Experimental	13	MCX
SYSTEM/ODT Program	– Standard	14	ODT
SYSTEM/ODT Program	– Experimental	15	ODX
SYSTEM/COPY Program	– –	16	CPY
Usercode/Password File	– –	17	US
SYSTEM/PANDA Program	– –	18	PAN

Both the COLDSTART/TAPE and COLDSTART/DISK stand-alone utility program construct the initial Name Table, and load and identify the following default files on the system disk:

Name Table Mnemonic	File Identifier
N	SYSTEM/INIT
G	GISMO3 or GISMO2
I	SDL2/INTERP3M
M	MCPII
C	SYSTEM/CONTROLLER
MM	MCPII/MICRO-MCP
ODT	SYSTEM/ODT
PAN	SYSTEM/PANDA

These defaults are selected from the Name Table when a standard CLEAR/START operation is performed. If temporary changes to the operating environment have been specified, the CLEAR/START program overrides the default Name Table selections with the entries required by the environment specified.

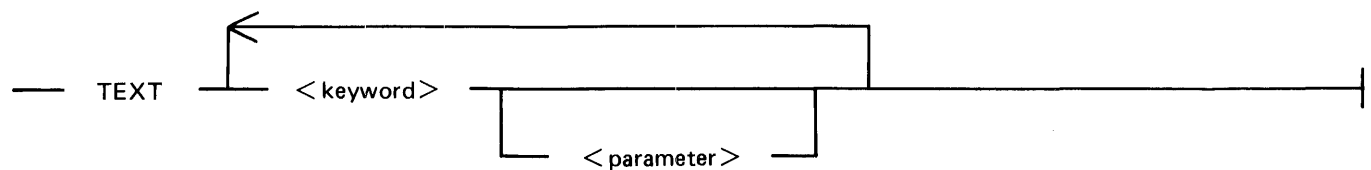
TEMPORARY ENVIRONMENT CHANGES

Several changes to the default selections made by the CLEAR/START routine can be specified. On B 1990 systems these changes are made before entering CLEAR MTR GO [XMT] to begin the clear/start process. On other B 1000 systems, the changes are made after the bootstrap loading process halts with @AAAAAA@ in the L register.

B 1990 Systems

Temporary environment changes for B 1990 systems are made by using the Soft Panel command TEXT before the CLEAR MTR GO [XMT] entry.

Syntax:



<keyword> descriptions follow. <parameter> applies to the keywords DC, IC, PM, and DD; which are discussed following the keyword descriptions.

Keyword	Function Requested
AMC	Initiate the MCS program that is in the MCS Name Table entry if the AMCS option is reset. Do not initiate the MCS program if the AMCS option is set.
CX	Select the experimental Controller (CX entry), rather than the standard NDL Controller (C entry).
DC	Override the default selection of the system disk channel. Refer to the System Disk Channel Selection Override subsection for details.
DD	(B 1990 systems only.) Change the disk drive unit order. See the Disk Drive Unit Order subsection for details.
DM	Memory dump. Refer to the subsection entitled Supplementary Information Required With a Memory Dump for additional information.
DUMP	Same as DM.
GX	Select the experimental GISMO (GX entry), rather than the default GISMO (G entry).
IC	Delete I/O channels artificially. Refer to the I/O Channel Deletion subsection for details.
IH	Enable debug halts in the System Initializer. For use by Burroughs software development personnel only.
IX	Select the experimental SDL2 Interpreter (IX entry), rather than the default SDL2 Interpreter (I entry).
M	Select the MCP. (This is the default.)
MCX	Select the experimental MCS (MCX entry) rather than the standard MCS program (MCS entry).
MMX	Select the experimental Micro MCP (MMX entry), rather than the default Micro MCP (MM entry).
MX	Select the experimental MCP II (MX entry), rather than the default MCP (M entry).
NX	Select the experimental System Initializer (NX entry), rather than the standard System Initializer (N entry).
ODX	Select the experimental ODT (ODX entry), rather than the standard SYSTEM/ODT program (ODT entry).
PM	Artificially reduce the system memory size. Refer to the Pseudo-memory Size Specification subsection for details.
TG	Set GISMO trace flags. See <i>B 1000 Systems System Software Operation Guide, Volume 1</i> , for a complete description of TG.

The following example requests a memory dump and specifies the experimental MCP (MX entry):

```
TEXT DUMP MX [XMT]
```

Other B 1000 Systems

Changes to the default selections made by the CLEAR/START routine can be specified by entering parameters in certain registers after the bootstrap loading process completes with @AAAAAA@ in the L register during the clear/start process.

Register entries are made by using the REGISTER GROUP and REGISTER SELECT rotary switches to access the desired register, setting the appropriate bits by means of the 24 toggle switches, pressing the LOAD push button, and observing the pattern that appears in the 24 display lights associated with the toggles.

Changes to the system software and firmware that is selected by the CLEAR/START program, as well as a request for a memory dump, are specified by setting certain T register bits. Bit meanings follow (bit 0 is leftmost):

Bit	Function Requested
0	Memory dump. Refer to the following subsection entitled Supplementary Information Required With A Memory Dump for additional information.
1-3	Not used.
4	Select the experimental MCPII (MX entry), rather than the default MCPII (M entry).
5	Select the experimental System Initializer (NX entry), rather than the standard System Initializer (N entry).
6	Select the experimental SDL2 Interpreter (IX entry), rather than the default SDL2 Interpreter (I entry).
7	Select the experimental GISMO (GX entry), rather than the default GISMO (G entry).
8	Select the experimental Micro MCP (MMX entry), rather than the standard Micro MCP (MM entry).
9	Select the experimental Controller (CX entry), rather than the standard NDL Controller (C entry).
10	Select the experimental MCS (MCX entry), rather than the standard MCS program (MCS entry).
11	Select the experimental ODT (ODX entry), rather than the standard SYSTEM/ODT program (ODT entry).
12	Enable debug halts in System Initializer. For use by Burroughs software development personnel only.
13-15	Not used.
16	Initiates the MCS program that is in the MCS Name Table entry if the AMCS system option is reset. Does not initiate the MCS program if the AMCS option is set.
17-23	Not used.

The system disk channel, I/O channel deletion, and pseudo memory size can also be specified. Refer to the System Disk Channel Selection Override subsection, the I/O Channel Deletion subsection, and the Pseudo-Memory Size Specification subsection for details.

System Disk Channel Selection Override

The CLEAR/START program normally selects Electronics Unit Zero (EU 0) or Drive Zero of the high-speed disk device present as the base systems disk. The selection hierarchy is as follows (from highest to lowest):

- B 9470 Head-per-Track disk
- Head-per-Track disk (other than B 9470)
- Disk Pack

When two or more channels contain identical disk device types, the CLEAR/START program selects the lowest-numbered channel as the systems disk channel.

This default system disk selection can be overridden during the clear/start operation by using the TEXT DC command on B 1990 systems, or by loading the X register on other B 1000 systems.

System disk selection is valid for the current clear/start operation only; that is, the default disk is always selected during a clear/start unless the override is invoked.

The syntax of the TEXT DC command is

TEXT DC x [XMT]

where x is a hexadecimal literal representing the desired system disk channel. For example, to specify channel 10, the TEXT command would be

TEXT DC A [XMT]

On other B 1000 systems, channel selection is made by selecting the X register and loading it during the intermediate halt of the clear/start process. The port and channel of the desired system disk channel are entered in the X register as follows:

Bits	Contents
0-16	Not used
17-19	Port number (must be 7)
20-23	Channel number

For example, to specify Port 7, channel 10, the X register would be set to:

@00007A@ (0000 0000 0000 0000 0111 1010)

I/O Channel Deletion

The ability is provided to artificially delete I/O devices (that is, to make certain channels appear unassigned to the system software) during the clear/start process. This feature is primarily of value to Burroughs Field Engineers when trying to isolate hardware malfunctions involving the I/O control subsystem. The specifications are made by entering the TEXT IC command on B 1990 systems, or by loading the FA register on other B 1000 systems.

The syntax of the TEXT IC command is as follows:

```
TEXT IC xxxx [XMT]
```

The xxxx refers to four hexadecimal digits representing the channel specification.

On the other B 1000 systems, the FA register is loaded during the intermediate halt of the clear/start process. The FA register is divided into two groups, as follows:

Bits	Contents
0-15	Channel specification
16-23	Not used.

The channels to be deleted are specified by setting one or more of bits 0 through 14 of the channel specification. Bit 0 represents channel 0, bit 1 represents channel 1, and so forth. (Channel 15 cannot be deleted.)

As an example, the following value causes the CLEAR/START and SYSTEM/INIT programs to delete channels 2, 5, 7, and 12 (port 7 is always implied):

```
@2508@ (0010 0101 0000 1000)
```

For B 1990 systems, the appropriate TEXT command to delete these channels is

```
TEXT IC 2508 [XMT]
```

For other B 1000 systems, the FA register would be set to

```
@250800@ (0010 0101 0000 1000 0000 0000)
```

Channels deleted during the CLEAR/START operating procedure appear unassigned to the MCP.

Pseudo-Memory Size Specification

The memory used on a system can be artificially reduced. This can be done by the TEXT PM command on B 1990 systems, or by loading the LR register on other B 1000 systems.

The syntax of the TEXT PM command used in B 1990 systems is

```
TEXT PM xxxxxx [XMT]
```

The x's refers to six hexadecimal digits reflecting a hexadecimal value. The meaning of the value is explained next.

On the other B 1000 systems, the LR register is loaded during the intermediate halt of the clear/start process. The LR register is functionally divided into two portions, as follows:

Bit	Function
0-11	Hexadecimal value
12-23	Not used

The CLEAR/START program multiplies the value specified by 512 bytes to obtain the pseudo-size of the main memory. The following table of values depicts sample main memory sizes that might be specified:

Value	Main Memory Size	
@400000@	524,288 bytes	(0.5 MB)
@800000@	1,048,576 bytes	(1.0 MB)
@C00000@	1,572,864 bytes	(1.5 MB)

For example, to artificially set the main memory of a B 1990 system to 524,288 bytes, the following TEXT command would be used:

```
TEXT PM 400000 [XMT]
```

To obtain the equivalent pseudo-memory size on other B 1000 systems, the LR register would be set to

```
@400000@ (0100 0000 0000 0000 0000 0000)
```

NOTE

Values larger than the actual physical size of main memory should not be specified.

Disk Drive Unit Order Selection

On B 1990 systems only, the DSC can be programmed to change the order of the disk units. For example, DPD (unit 3) can be made to be DPA (unit 0). This is done with the TEXT DD command as follows:

```
TEXT DD xxxxxxxx [XMT]
```

Each of the eight x's stands for a digit from 0 through 7. From left to right, the x's represent the order of the disk units: leftmost is DPA, next is DPB, and so on. The default order is 01234567.

For example, to specify that unit 3 is to be DPA and unit 0 is to be DPD (that is, to interchange DPA with DPD) while leaving the remaining units as is, the following TEXT command is used:

```
TEXT DD 31204567 [XMT]
```

All eight digits must be specified, even if the system does not have eight drives on the DSC. The changing of the disk unit order is in effect for the current clear/start only; if a change is not specified upon another clear/start, the DSC reverts to the default order.

DISK CACHE DISABLE FUNCTION

A Disk Cache subsystem can be optionally added to any DSC (maximum of two DSC's with one disk cache each) on B 1965/B 1995 system only. The ability to disable the disk cache is provided with the NDC1 and NDC2 text commands as follows.

```
TEXT NDC1 [XMT]
TEXT NDC2 [XMT]
```

NDC1 disables the disk cache associated with the first DSC (lower channel ID) and NDC2 disables the disk cache associated with the second DSC (higher channel ID). The default for disk cache is enabled. Any subsequent CLEAR/START without entering a cache disable command will cause the cache to be enabled.

SUPPLEMENTARY INFORMATION REQUIRED WITH A MEMORY DUMP

If the system comes to an orderly halt, with the RUN and ERROR lights out (RUN light out, no Soft Panel ERROR indication on B 1990 systems), then this usually means that the software halted the system. The L register defines the halt. (A list of system halts is available in section 8 of Volume 1 of the *B 1000 Systems System Software Operation Guide*.) Any relevant facts about the halt should be documented and given to the Burroughs Technical Representative with the dump that is taken.

If the RUN light is out and the ERROR indicator is on, register settings must be recorded. The halt should be treated as if a system hang had occurred and the HALT push button had been pressed, as is described in the following paragraphs. When ERROR is indicated, either the PERP register or the PERM register has a value indicating the cause of the halt.

Many system problems are defined to be hangs, meaning that the system is running (RUN light on) but no work is being completed. This situation is a loop; the system is repetitively executing an instruction sequence but not exiting from it.

Before attempting a clear/start operation, two tests must be tried and the results documented:

1. Determine whether ODT interaction is possible by attempting a simple system command such as MX or WT.
2. Press the INTRPT push button.

If the INTRPT push button halts the system (RUN light goes out), the L register displays @0D0010@ or @000010@.

If the INTRPT push button fails to halt the system (RUN light stays on), it is extremely important to record the values of certain registers (for both processors in a dual-processor system) after the halt is achieved. The dump is almost always useless if it is submitted without register contents or with erroneous contents such as @FFFFFF@ in all registers, obtained because the system was still running.

Press the HALT push button to stop the system. (In B 1990 systems, this is labeled HALT/RUN.)

In the rare instance that the HALT push button fails to stop the system (that is, the RUN light remains lit), press the HALT and CLEAR push buttons simultaneously. This halts the system. This situation reflects a processor problem and a Burroughs Field Engineer must be contacted. B 1990 systems have no CLEAR push button, but it is extremely unlikely that the HALT/RUN push button would fail to halt one of these systems.

Following are the registers of interest:

L, T, X, Y, A, FA, LR, CC, CD, PERM, PERP

On dual-processor systems, results must be recorded for each processor.

At this point, the operator can clear/start the system and take the memory dump previously described. After the clear/start operation, the dump should be "packaged" as soon as possible. This is done with the PM system command. (Refer to the *B 1000 Software Operation Guide, volume 1.*) This creates a packaged dumpfile labeled "DUMPFIL/PM" which can be printed or copied to another medium and given to the Burroughs Technical Representative, along with the supplementary information outlined previously.

SECTION 5

CODE/ANALYZER

The CODE/ANALYZER program, a normal-state utility program, evaluates a code file to produce and list information about its associated files, data and code segments, and non-zero switch settings, and to calculate an estimate of memory required for the program to run. The code file and any intrinsic files it uses must be present on disk.

NOTE

The memory estimate calculated by the CODE/ANALYZER program is a minimum memory requirement, not a working-set. The program can use more memory if it is available, or less if not all of the files are open at the same time.

OPERATING INSTRUCTIONS

The CODE/ANALYZER program is executed from the ODT with an EXECUTE command. Upon execution, the CODE/ANALYZER program prompts the operator to enter the identifier of the object-code file that is to be analyzed. If the file identified is not a code file or is not on disk, an error message is displayed on the ODT. After the named file is analyzed, the prompt is repeated until the operator enters a blank input string.

CODE/ANALYZER = <mix = number> ENTER PROGRAM NAME

A valid response has the following format:

<mix-number> AX <file-identifier>

PROGRAM SWITCHES

The CODE/ANALYZER program is sensitive to the values of program switches 0, 1, and 9. The default value for each switch is 0 (reset). Action when each switch is set (1) is described next.

Switch 0

Causes the CODE/ANALYZER program to accept the file-identifier specifications from a card file labeled CARDS. File IDs are entered in free-form format in the first 80 columns of each card, one per card. When switch 0 is set, the ODT prompts are not issued.

Switch 1

Causes additional information contained in the File Parameter Blocks (FPB) to be printed.

Switch 9

Controls printer spacing and skipping. When switch 9 is set, the listing generated by CODE/ANALYZER is printed single-spaced with no skips to headings.

SECTION 6

COLDSTART/DISK

The term "coldstart" refers to the process of loading the basic system software and firmware from a disk or a library tape to a disk that is to become the system disk. This section describes COLDSTART/DISK, a utility program that is used when the input software and firmware are on either a disk pack or a head-per-track disk. Section 7 describes COLDSTART/TAPE, a similar program that is used when the input software and firmware are on a library tape.

COLDSTART/DISK PROGRAM OVERVIEW

COLDSTART/DISK is a stand-alone program that loads the basic B 1000 system software and firmware from an existing disk to another disk that is to become the system disk. The program is furnished on a cassette tape and loaded through the cassette reader.

The COLDSTART/DISK program performs the following functions:

1. Displays prompts requesting initial information.
2. Builds and initializes the disk directory and available tables on the new system disk.
3. Loads the system software and firmware files.
4. Builds the name table and makes the appropriate system software and firmware entries.
5. Builds the coldstart variables.
6. Builds the SYSTEM/ODT-QUEUE file and the hardware maintenance log (ELOG).
7. If requested, loads additional files and compares data.
8. Displays error messages as required during program execution.
9. Displays a CLEAR/START request following program execution.

Input and Output Disks

The input disk, which contains the system software and firmware and, possibly, additional files, may be a head-per-track disk unit (DKA) or any single disk pack (DP <letter or number>). The program does not accept input from multiple packs.

The output disk, which is to become the system disk, may be either a head-per-track disk unit (DKA) or DPA, the disk pack on drive 0 of a disk pack subsystem.

If the input disk is DKA, the output disk must be DPA. If the input disk is DPA, the output disk must be DKA. If the input disk is any pack except DPA, the output disk must be DKA if there is one on the system or DPA if there is no DKA.

The COLDSTART/DISK program clears and initializes the existing directory of the target disk. Therefore, all files that were in that directory are lost.

System Software Files

The COLDSTART/DISK program must find the following standard system software code files on the input disk:

MCPII
MCPII/MICRO-MCP
GISMO2 or GISMO3
SDL2/INTERP3M
SDL2INTRIN/AGGREGATE
SYSTEM/ODT
SYSTEM/INIT
SYSTEM/COPY
SYSTEM/CONTROLLER
SYSTEM/PANDA
SYSTEM/LOAD-CAS *

* Needed if the system has no magnetic tape capability.

If any of these names cannot be located, the program requests that alternate code file names representing acceptable substitute files be entered from the ODT.

During program execution, the system code file headers are modified to mark the files as protected. This is done to prevent certain MCPII commands (for example, REMOVE) from affecting these files.

Program Options

Two program options may be invoked in response to prompts that are provided by the program.

The first option is called COMPLETE COPY. When invoked, the COMPLETE COPY option causes any additional files on the input disk to be copied to the new system disk after the system software and firmware files have been successfully copied. Thus, these additional files can be copied during the coldstart process rather than after the subsequent CLEAR/START. This is an important capability if the additional files are not available on other media.

The following files are not copied:

SYSTEM/ELOG
SYSTEM/LOG
SYSTEM/DUMPFIL
SYSTEM/ODT-QUEUE

The COMPLETE COPY option must not be used if there are more files on the input disk than can be contained on the output disk.

The second program option is called DATA COMPARE. When invoked, the DATA COMPARE option causes the input and output data to be compared and provides the following advisory message for each file in which a copying error occurred:

COMPARISON ERROR <filename>

OPERATING INSTRUCTIONS

The coldstart process is done in two steps. First, the COLDSTART/DISK program is loaded to system memory from the cassette tape. Then, the program loads the system software and firmware from the input disk to the output disk. At the start of the second step, the program displays prompts on the ODT to request information identifying the disks and specifying options. During program execution, error and advisory messages are also displayed on the ODT.

Before beginning the actual program-loading steps, the disk media are mounted as follows (HPT = head-per-track = DKA):

1. Pack to pack: The pack on drive 0 (DPA) is the output disk pack; the input disk pack can be mounted on any drive except DPA.
2. Pack to HPT: The HPT is the output disk (DKA); the input disk pack may be mounted on any drive (DP<letter or number>).
3. HPT to pack: The pack on drive 0 (DPA) is the output disk pack; the input disk is the HPT (DKA).

NOTE

The new system disk will function only on a disk unit that is addressed by the same port, channel, and unit numbers that addressed the unit where it was coldstarted.

Two program-loading sequences are provided. The first is used for B 1990 systems and the second is for all other B 1000 systems.

Program Loading -- B 1990 Systems

To load the COLDSTART/DISK program to a B 1990 system:

1. Halt the system by entering the MCP command HALT via the ODT and pressing XMT. For dual-processor systems be sure both processors are halted (RUN A and RUN B lights are both out).

NOTE

Instructions for using the B 1990 Soft Panel in this fashion are hereafter presented as follows:

<keyword-1> <keyword-2> ... [XMT]

A looping condition could deny access to the ODT. In such cases, press the INTRPT push button. If the system still does not halt, press the HALT/RUN push button.

2. Place the COLDSTART/DISK cassette into the cassette reader and ensure that the tape rewinds to BOT.
3. If the output disk is DPA but drive 0 is not operating, then a TEXT DD entry must be made. The following example makes drive 5 addressable as DPA:

TEXT DD 51234067 [XMT]

4. CLEAR MTR GO [XMT]

Program Loading -- All Other B 1000 Systems

To load the COLDSTART/DISK program on other B 1000 systems:

1. Halt the system using either the INTRPT push button or the console INTERRUPT switch. If the INTERRUPT switch is used, return it to its initial position following the halt. If the system continues to run, press the HALT push button. If the system still continues to run, press the HALT and CLEAR push buttons simultaneously. For dual-processor systems be sure both processors are halted.
2. Place the COLDSTART/DISK cassette into the cassette drive and ensure that the tape rewinds to BOT.
3. Press (1) the CLEAR push button, (2) the MODE push button to obtain the MTR mode, and (3) the START push button.
4. After the bootstrap loader is read from the cassette tape, the processor halts. The L register must contain @AAAAAA@ at this time; if not, the cassette must be rewound and the procedure restarted from step 3.
5. Press the MODE push button to return to the NORMAL mode, then press the START push button. Reading of the cassette resumes. Loading is complete when the STATE light goes on.

NOTES

If the tape does not automatically rewind when it is inserted into the console cassette drive, press REWIND. If the tape is wound completely onto the right spool, remove the cassette and manually rewind it until the clear leader is no longer at the read gap.

If the system halts with @000004@ or @0D0053@ in the L register or with the PERP register equal to 1, a cassette error during program loading is indicated. Restart the procedure from step 1.

Program Execution

COLDSTART/DISK program execution now begins, and the following prompts appear in turn on the ODT display:

```
COLDSTART/DISK - <mark-level>.<patch-level> <compile-date-time>
-- PLEASE PUSH SPCFY IF YOU SEE AN ETX ON THIS SCREEN --
ENTER OUTPUT DRIVE - <DPA OR DKA>
```

If there is an ETX character on the screen, press the SPCFY key once. This is required because of the variety of ODT device types and firmware settings available on the B 1000 family of computer systems.

A valid response to ENTER OUTPUT DRIVE causes the next message to be displayed.

```
ENTER INPUT DRIVE - <DP? OR DKA>
A valid response causes the next message to be displayed.
```


IS COMPLETE COPY DESIRED? <YES OR NO>

A YES response invokes the COMPLETE COPY option which causes all files rather than only the system software files to be copied to the system disk. The additional files are copied after the system software is successfully loaded.

Transmitting "STATUS" or "ST" at any time during execution of the COMPLETE COPY option produces the following status message:

CURRENT FILE: <file-name> APPROX: <percentage>% COMPLETE

CURRENT FILE is the file processed when the query is answered; <percentage> relates to the overall task.

IS DATA COMPARISON DESIRED? <YES OR NO>

A YES response invokes the DATA COMPARE option. This option is used to verify that each file of input data is copied correctly to the output disk. An advisory message is printed for any file in which errors are found. Each such file is copied as completely as possible but it might be unusable.

When program execution is complete, the following message is displayed:

COLDSTART COMPLETE - CLEAR/START REQUIRED

The following is a typical sequence:

```
COLDSTART/DISK - MARK 12.0.000 (07/27/84 16:42)
ENTER OUTPUT DRIVE - <DPA OR DKA>
DKA

ENTER INPUT DRIVE - <DP? OR DKA>
DPB

IS COMPLETE COPY DESIRED? <YES OR NO>
YES

IS DATA COMPARISON DESIRED? <YES OR NO>
YES

COLDSTARTING DKA FROM DPB

<error-or-information-messages>

COLDSTART COMPLETE - CLEAR/START REQUIRED
```

As a result of this response sequence, the head-per-track (DKA) disk is coldstarted using the pack on drive 1 (DPB) as the input disk. The standard program names are used for the COLDSTART procedure, and the remaining files on DPB are copied to the new output disk. Data comparison is performed.

HALTS

The value @000011@ appears in the L register when a halt occurs. (B 1990: Soft Panel; all other B 1000 systems: console lights.) The value in the T register at this time indicates the nature of the halt, as specified in the following paragraphs. Except for ODT halts, each halt invokes the display of an explanatory message on the ODT. This message appears on page 2 of the terminal if running with B 1990 soft panel.

@AAAAAA@

The COLDSTART/DISK program completed successfully.

@F0F0F0@

The COLDSTART/DISK program aborted. Refer to the ODT for the reason.

FC0000@

No ODT was found on the system.

@FC1Dxx@

The dispatch to the ODT was denied. xx contains the port and channel number.

@FC1Exx@

Data error encountered on the ODT. xx contains the ODT port and channel number.

@FC20xx@

Could not load the multiline control ODT firmware. xx contains the unit control number indicating the failure with the MLC 4/5 firmware load.

@FC2Dxx@

Remote ODT dispatch was denied. xx contains the ODT port and channel number.

@FD0000@

No disk was found on the system.

@FD0100@

Not enough output disk space for the current file or structure. This may indicate a bad available table on the output pack.

@FD0200@

Invalid Master Available Table sector on the output pack.

ERROR AND ADVISORY MESSAGES TO THE ODT

DISK ERROR – RESULT IN 'T'
PUSH START TO RETRY ANOTHER 100 TIMES

The T register contains the I/O result descriptor that is in error. Press START to retry the I/O operation. (B 1990: transmit RUN or GO.)

INVALID RESPONSE – TRY AGAIN

The response to a prompt was invalid.

FILE <file-name> HAS A BAD HEADER AT <disk-address>
FILE NOT COPIED

The input file has a bad header, or the header is located on another disk. The program continues unless the file is a required system file.

BAD DISK FILE HEADER ADDRESS FOUND FOR FILE <file-name>
FILE NOT COPIED

A bad header address was found in the directory. The program continues unless the file is a required system file.

INSUFFICIENT DISK FOR COLDSTART
THE ENTIRE PROGRAM MUST BE RELOADED AND RESTARTED

The files required for a coldstart need more disk space than is available on the output disk. Obtain another disk for output and begin the coldstart process again. The problem might be a bad available table on the output disk, too many deleted (XD'ed) sectors on the output disk, or a size mismatch between the input and output disks.

NO DISK DEVICE ON SYSTEM

The program cannot find the appropriate disk units in the system configuration. (The program cannot be used if there are no user disks on the system.) Ensure that the required disk devices are available, connected properly, and on line. Begin the coldstart process again.

<multi-file-id> INVALID SUB DIRECTORY

The program cannot access some or all of the files with the <multi-file-id> that has been specified. This implies that the subdirectory for that <multi-file-id> is bad or cannot be located because of a bad main directory entry.

MISSING REQUIRED SYSTEM FILE <file-name>
PLEASE ENTER THE CORRECT FILENAME FOR <file-name>

The specified program was not located on the input disk or could not be copied because of some error. Another name may be entered in its place if the file specified by the new name has the same function as the file expected by the COLDSTART/DISK program.

MISSING REQUIRED SYSTEM FILE GISMO3 OR GISMO2
PLEASE ENTER THE CORRECT FILENAME FOR GISMO3 OR GISMO2

Similar to the preceding messages.

<multi-file-id> <file-id> IS A MULTI PACK FILE - CANNOT COPY

A multi-pack file cannot be copied to a system disk.

DISK NOT READY <unit-mnemonic>

Ready the disk and press START to continue.
(B 1990: transmit RUN or GO.)

<multi-file-id> <file-id> IS A PSEUDO DECK - CANNOT COPY

A pseudo-deck was encountered among the files being copied. This is an information message; copying continues.

WRITE LOCKOUT <output-unit-mnemonic>

Remove the write lockout and press START to continue.
(B 1990: transmit RUN or GO.)

PACK LABEL BAD <unit-mnemonic>

The label on the output disk pack is invalid. The COLDSTART/DISK program cannot use this disk.

WARNING – <file-id> IS NOT AN IAD FILE ON

<output-unit-mnemonic>

(IAD = Installation Allocated Disk) The input file is an IAD file but the resulting output (name: <file-id>) is not an IAD file.

<output-unit-mnemonic> IS NOT A SYSTEM DISK – CANNOT COLDSTART

The disk to be coldstarted must be a system disk.

COMPARISON ERROR <file-id>

This is an advisory message that is issued when a comparison error is found. (The DATA COMPARISON option must be invoked.) The file named should be physically verified on the output pack.

INVALID MAIN DIRECTORY SECTOR <disk-address> <unit-mnemonic>

An invalid main directory sector has been encountered on the input disk. The entire sector and any linked sectors have been skipped. <disk-address> is a hex value.

INPUT DISK ERROR – SECTOR <nnnnnn> – CURRENT FILE IS <file-id>

OUTPUT DISK ERROR – SECTOR <nnnnnn> – CURRENT FILE IS <file-id>

Retries could not correct read or write errors on the specified files. The file is copied to the output disk, possibly with errors, and should be physically verified. <nnnnnn> is a hex value.

INVALID SUB DIRECTORY SECTOR ON <unit-mnemonic>

INPUT DISK – SECTOR <sector-address>

An invalid subdirectory sector has been encountered on the input disk.

<file-name> MISSING OR DUPLICATE FILE

DISK I/O TIMEOUT – I/O FAILED TO COMPLETE <unit-mnemonic>

DISK I/O DISPATCH DENIED (LOCKED OR MISSING DEVICE) <unit-mnemonic>

PLEASE CORRECT THIS PROBLEM AND PUSH START

THE ENTIRE PROGRAM MUST BE RELOADED AND RESTARTED

COLDSTART ABORTED – THE OUTPUT PACK ON <unit-mnemonic> IS NOT A VALID PACK

SECTION 7

COLDSTART/TAPE

The term "coldstart" refers to the process of loading the basic system software and firmware from a user disk or a library tape to a disk that is to become the system disk. This section describes COLDSTART/TAPE, a utility program that is used when the input software and firmware are on a magnetic tape. Section 6 describes COLDSTART/DISK, a similar program that is used when the input software and firmware are on disk.

PROGRAM OVERVIEW

COLDSTART/TAPE is a stand-alone program that loads the basic B 1000 system software and firmware from a library tape labeled SYSTEM to a head-per-track disk unit or a disk pack, thus creating a system (S) disk. The program is furnished on a cassette tape and loaded through the cassette reader. The library tape may be mounted on any available tape drive; the program locates the tape automatically.

The COLDSTART/TAPE program performs the following functions:

1. Builds and initializes the disk directory and available tables on the new system disk.
2. Loads the system software and firmware files.
3. Builds the name table and makes the appropriate system software and firmware entries.
4. Builds the coldstart variables.
5. Builds the SYSTEM/ODT-QUEUE file.
6. Builds the hardware maintenance log (ELOG).
7. Uses the L register to signal errors and halts and the T register to define them.

Input and Output Media

The input medium is a single reel library tape labeled SYSTEM that contains the system software and firmware files. This tape may be mounted on any available drive; the COLDSTART/TAPE program locates it automatically.

The output medium is the head-per-track disk unit (DKA) if there is one on the system; otherwise, it is the pack on drive 0 (DPA) of the disk pack subsystem. In either case, the target disk must have been initialized as a system disk. (Refer to section 20, PACK/INIT, and/or section 33, SYSTEM/DISK-INIT in this manual for information on disk initialization.)

The system disk created by the COLDSTART/TAPE program is a single system pack configuration that contains no system logfiles except the ELOG. Once the system is running under MCP II control, the number of system drives can be increased with the SD system command and the LOG and ODTL MCP options can be set with the SL system command.

CAUTION

The COLDSTART/TAPE program clears and initializes the existing directory of the target disk. Therefore, all files in that directory are lost following the coldstart.

System Software Files

The COLDSTART/TAPE program must find the following standard system software code files on the input tape:

MCPII
MCPII/MICRO-MCP
GISMO2 or GISMO3
SDL2/INTERP3M
SDL2INTRIN/AGGREGATE
SYSTEM/ODT
SYSTEM/INIT
SYSTEM/COPY
SYSTEM/CONTROLLER
SYSTEM/PANDA

NOTE

If the system tape is larger than a single reel, it is imperative that all the required system software code files reside on the first reel. The COLDSTART/TAPE program only accepts input from a single reel.

Program Options

No options are included in the COLDSTART/TAPE program.

OPERATING INSTRUCTIONS

Coldstart is a two-step process. First, the COLDSTART/TAPE program is loaded to system memory from the cassette tape. Then, the program loads the system software and firmware from the input tape to the output disk. During program execution, the L register specifies the occurrence of errors and halts, and the specific condition causing the problem is indicated by the contents of the T register.

Two sequences are provided for loading the COLDSTART/TAPE program. The first is used for B 1990 systems and the second is for all other B 1000 systems.

Program Loading -- B 1990 Systems

1. Halt the system by entering the MCP command HALT via the ODT and pressing XMT. For dual-processor systems be sure both processors are halted.

NOTE

Instructions for using the B 1990 Soft Panel in this fashion are hereafter presented as follows:

<keyword-1> <keyword-2> ... [XMT]

A looping condition could deny access to the ODT. In such cases, press the INTRPT push button. If the system still does not halt, press the HALT/RUN push button.

2. Place the COLDSTART/TAPE cassette into the cassette reader and ensure that the tape rewinds to BOT.
3. If the output disk is DPA, but drive 0 is not operating, then a TEXT DD entry must be made. The following example makes drive 5 addressable as DPA:

```
TEXT DD 51234067 [XMT]
```

4. Enter CLEAR MTR GO [XMT].

Program Loading -- All Other B 1000 Systems

1. Halt the system using either the INTRPT push button or the its initial position following the halt.) If the system continues to run, press the HALT push button. If the system still continues to run, press the HALT and CLEAR push buttons simultaneously.
2. Place the COLDSTART/TAPE cassette into the cassette drive and ensure that the tape rewinds to BOT.
3. Press (1) the CLEAR push button, (2) the MODE push button to obtain the MTR mode, and (3) the START push button. This causes the bootstrap loader to be read from the cassette into system memory. The system halts with @AAAAAA@ in the L register. If this does not occur, rewind the tape and try again from step 3. (Also, see NOTE 2, following.)
4. Press MODE to return to NORMAL mode, then press START. Cassette reading resumes. Loading is complete when the STATE light goes on.
5. Go to the next subsection (Program Execution).

NOTES

If the tape does not automatically rewind when it is loaded into the cassette, press REWIND. (B 1990: transmit keyword REWIND.) If the tape is wound completely onto the right spindle, remove the cassette and manually rewind it until the clear leader is no longer at the read gap.

A system halt with @000004@ or @0D0053@ in the L register or with a 1 in the PERP register, signifies a cassette error during loading of the program. Restart the procedure from step 1.

Program Execution

After the COLDSTART/TAPE program is loaded from the cassette, program execution begins and continues until it is complete or until a halt occurs. The program finds and loads the nine standard software files (listed previously), which may be in any sequence on the SYSTEM tape. The program also performs all the functions listed earlier. Successful completion is signalled by a halt with the following register values displayed:

```
L = @000011@  
T = @AAAAAA@
```

The SYSTEM tape may also contain other files. These may be loaded following a CLEAR/START operation after the COLDSTART/TAPE operation successfully completes. The following control message is used:

```
ADD & COMPARE =/= FROM SYSTEM (KIND=TAPE)
```

HALTS

A halt or the completion of execution of the COLDSTART/TAPE program is signalled by @000011@ in the L register. The specific condition is denoted by the contents of the T register, as specified in the following paragraphs.

NOTE

On a B 1990 system, type GO and push the TRANSMIT key whenever the following instructions signal to press the START button.

@AAAAAA@

Normal end-of-job. CLEAR/START required.

@0C0001@

Disk I/O error. Press START once to obtain a display of the result descriptor value in the T register. (B 1990: transmit RUN or GO.)

@0C0002@

Tape I/O error. Press START once to obtain a display of the result descriptor value in the T register. (B 1990: transmit RUN or GO.)

@0C0003@

Unexpected data or result descriptor from tape I/O.

@0C0004@

No magnetic tape control found on system.

@0C0005@

No disk (pack or head-per-track) control found on system.

@0C0006@

Disk not initialized in the proper format.

@0C0007@

Attempted to coldstart a disk unit that has not been initialized as a system (S) disk.

@0C0008@

Could not locate tape labeled SYSTEM. Make tape ready and press START. (B 1990: transmit RUN or GO.)

@0C0009@

One or more of the required files is missing from the SYSTEM tape. Pressing START (B 1990: transmitting RUN or GO) causes the T register display to change. The rightmost byte now contains a number that stands for one of the files in the following list. Pressing the START push button repeatedly produces additional numbers if more than one file is missing. When all the missing files have been indicated, the T register display returns to @0C0009@.

File Number	File-identifier
1	MCPII
2	SDL2/INTERP3M
3	GISMO3
4	SYSTEM/INIT
5	MCPII/MICRO = MCP
6	SYSTEM/COPY
7	SDL2INTRIN/AGGREGATE
8	SYSTEM/ODT
9	SYSTEM/CONTROLLER
A	SYSTEM/PANDA
B	GISMO2

@0C000A@

Missing device on I/O dispatch operation.

@0C000B@

Insufficient disk for COLDSTART. This is probably the result of a bad Available or Master Available Table.

@0C000C@

Went beyond the tape mark for a specific file. Try the COLDSTART operation again; if it fails again, a corrupted system tape is indicated.

@0C000D@

Missing tape mark.

@0C000E@

Invalid tape label HDR2 record.

@0C000F@

An I/O failed to complete in a reasonable amount of time. Push START to display the port and channel in the T register.

SECTION 8 CONVERT/BACKUP

The CONVERT/BACKUP program converts B 1000 printer backup files into a format acceptable to the B 9270 Page Printer. The converted files are in the B 6000/B 7000 systems printer backup format.

The program accepts as input B 1000 tape-labeled or disk-labeled printer backup files. These must be files that were created under system software release Mark 9.0 or later; files created prior to the Mark 9.0 release are not acceptable.

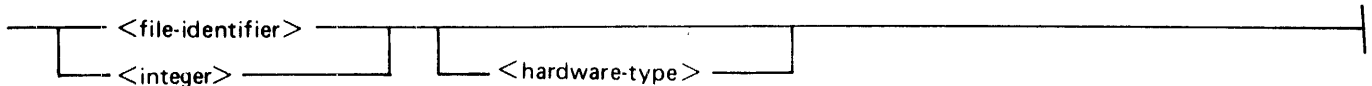
The program produces as output a single B 1000-labeled multifile tape named BACKUP. All the backup files that were accepted by the program as input are on this tape in their converted form, ready for processing by the B 9270 Page Printer. (Because of labeling differences, the tape is not recognized as a valid backup tape by B 6000/B 7000 systems.)

A B 1000 printer backup file with the internal file name B9270NAMES is also created. This file contains the names of all of the files converted during the program run.

OPERATING INSTRUCTIONS

The CONVERT/BACKUP program is executed from the ODT. Input specifications are entered in response to AC or AX messages. The program reads and processes each input specification as it is received. A null input terminates the program.

The input specifications are the input file identifier and an optional origin override. The syntax for the specification string follows:



The <file-identifier> syntax is used for specifying printer backup files by name, especially those with operator-assigned names. The format of the <file-identifier> is the same as that used in MCPPII control instructions and consists of one to three identifiers separated by slashes.

The <integer> syntax specifies by number the printer backup files that have MCPPII-generated names. In this case, BACKUP is the name to which the family name defaults for tape files as well as for disk files.

The optional <hardware-type> is the reserved word TAPE or MTP. If the <hardware-type> is not specified, the default is DISK.

A sample of the CONVERT/BACKUP program follows:

```
EXECUTE CONVERT/BACKUP

CONVERT/BACKUP =626 BOJ.
CONVERT/BACKUP =626 "B927OTAPE/DIRECTORY" = "BACKUP/PRT419"
% CONVERT/BACKUP =626 ENTER SPECS
CONVERT/BACKUP =626 ACCEPT.

626 AX BACKUP/PRT127                % LOOK FOR BACKUP/PRT127

% CONVERT/BACKUP =626 ENTER SPECS  % READY FOR NEXT FILE
CONVERT/BACKUP =626 ACCEPT.

626 AX 418                          % LOOK FOR BACKUP/PRT418

% CONVERT/BACKUP =626 ENTER SPECS
CONVERT/BACKUP =626 ACCEPT.

626 AX 420 TAPE                    % LOOK FOR BACKUP/420 ON TAPE

% CONVERT/BACKUP =626 ENTER SPECS  % READY FOR NEXT FILE
CONVERT/BACKUP =626 ACCEPT.

626 AX                              % TERMINATE THE RUN

CONVERT/BACKUP =626 "BACKUP/PRT419" RELEASED
CONVERT/BACKUP =626 EOJ.
```

ERROR MESSAGES

BACKUP FILE MUST BE CREATED ON RELEASE 9.0 OR LATER

The release level in the control record indicates that the backup file was created in a release earlier than 9.0. Program control returns to ENTER SPECS.

BACKUP FILE MUST BE LABELED

The backup file must be created from a labeled printer file. Program control returns to ENTER SPECS.

FILE NOT BACKUP. RE-ENTER SPECS

The input file is not a B 1000 backup file, so it cannot be converted. Enter the name of a B 1000 backup file to proceed, or make a null entry to terminate the program.

FILE NOT PRESENT. RE-ENTER SPECS

The input file could not be found. Enter the correct file name to proceed, or make a null entry to terminate the program.

THERE ARE NO RECORDS TO PROCESS

An EOF branch was encountered while looking for the control record in the backup file. Program control returns to ENTER SPECS.

SECTION 9 CREATE/ISAM

The CREATE/ISAM program produces a COBOL74-type ISAM file from a data file or from the data portion of an existing ISAM file. If the input is an existing ISAM file, a relative read of the data portion is performed, resulting in a new ISAM file that excludes those records that were marked for deletion.

OPERATING INSTRUCTIONS

The CREATE/ISAM program accepts file specifications from cards, the ODT, or remote terminals. Program switches 2 and 3 are used to specify the input interface.

Program Switch:	2	3	input
	0	0	terminal (default)
	0	1*	ODT
	1*	0	cards
	1*	1*	cards

* means any setting except 0.

The following input string is used to initiate execution:

EXECUTE CREATE/ISAM

The switch settings may be entered with the SWITCH control attribute or, after the job has gone to BOJ, by means of the SW system command. (See *B 1000 Systems System Software Operation Guide, volume 1.*)

Terminal Interface

Unless the default settings for program switches 2 and 3 have been changed to request other than the remote terminal interface, the file screen shown in figure 9-1 is displayed at the terminal following the BOJ message.

```

INPUT FILE SPECIFICATIONS
NAME-PACK ID      [      ] EMPTY IF FILE IS ON DEFAULT PACK
MULTI-FILE ID    [      ]
FILE ID          [      ]
DATA FILE PACK ID [      ] EMPTY IF DATA FILE SPECIFIED ABOVE
OR DATA FILE IS ON DEFAULT PACK

OUTPUT FILE SPECIFICATIONS
NAME-PACK ID      [      ] EMPTY IF FILE IS ON DEFAULT PACK
MULTI-FILE ID    [      ]
FILE ID          [      ]
I/S DATA FILE PACK ID [      ] EMPTY IF FILE IS ON DEFAULT PACK

```

EOJ []

Figure 9-1. File Screen

File Screen Semantics

The file screen is used to enter the desired input and output file names and their locations. The input file must be on disk but can reference one of the following sources:

1. The global (cluster) file of an existing COBOL74-type ISAM file.
2. The data portion of an existing COBOL74-type ISAM file.
3. A sequential file of type data.

DEFAULT PACK refers to (1) the System pack or (2), if the program is under a user code, to the default pack for that user code or (3), if the multi-file-id is a user code, to the default pack for that user code.

The output must be directed to disk. The naming conventions for the newly created ISAM file conform to the following rules:

1. The specified OUTPUT FILE ID becomes the name for the new global file.
2. The data and index files are assigned names as follows:

FILE ID [MYFILENAME]

global file name	MYFILENAME
data file name	00MYFILENA
primary key index file name	01MYFILENA
1st alternate key index file name	02MYFILENA
•	•
•	•
•	•
19th alternate key index file name	20MYFILENA

3. The output ISAM data file must not have the same name as the explicitly or implicitly specified input data file.
4. DISK may be used in the PACK ID fields to mean the system disk. This will override any default packs.

If the information entered through the file screen contains errors, the file screen display reappears with error messages; otherwise, the key screen (figure 9-2) is displayed.

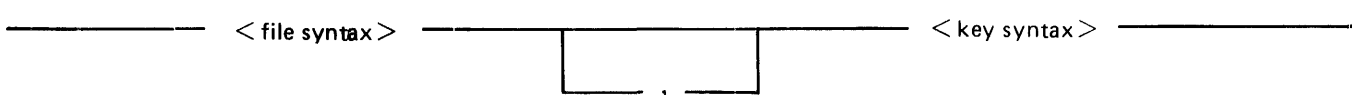
ODT Interface

The ODT interface allows the user to enter the program specifications as AC or AX commands in response to the MCP ACCEPT mechanism. This interface is activated when program switch 3 is set (non-zero).

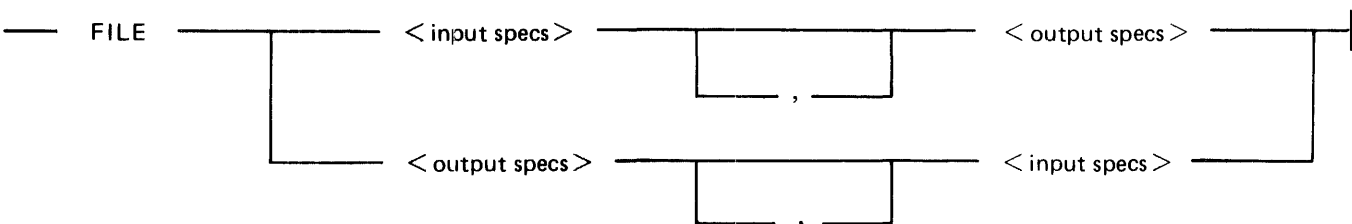
Specifications are entered in the free-field format. The program accepts up to 160 characters per command until a blank or a null string is entered with the AC or AX command. One ISAM file is created with each execution of the CREATE/ISAM program. All limits described in the terminal interface apply in the ODT interface.

Syntax (ODT Interface):

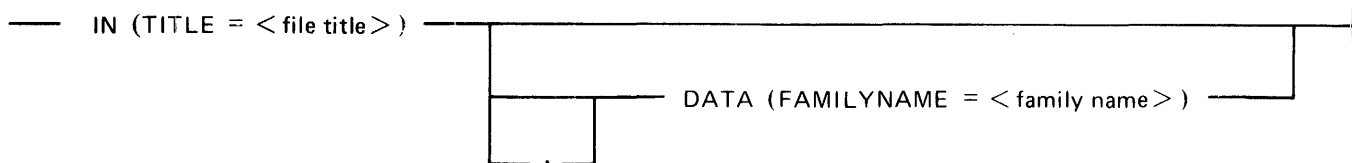
<ODT interface>



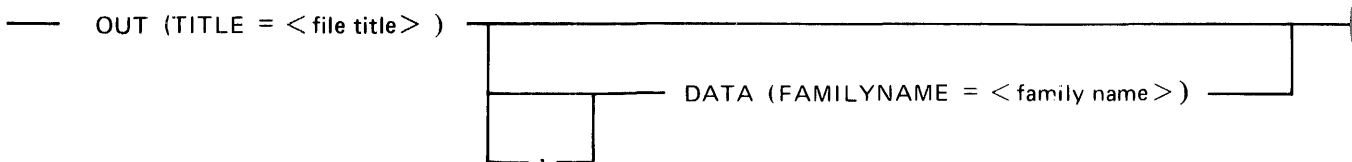
<file syntax>



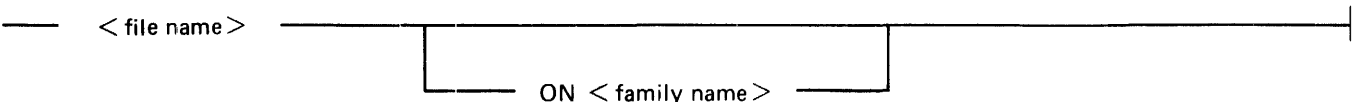
<input specs>



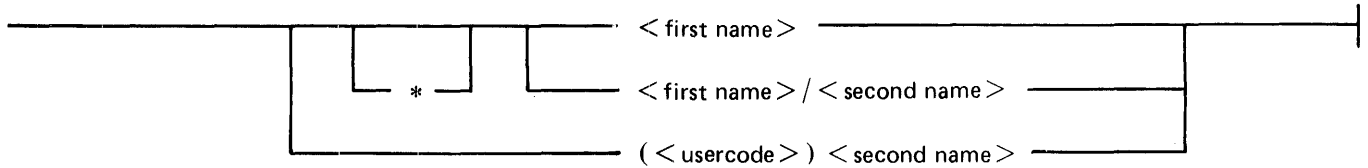
<output specs>



<file title>



<file-name>



Semantics (ODT Interface):

The characters usable in the preceding syntax are upper-case letters, numbers, underscores, and hyphens.

<file name>

Used to reference a file without regard to its medium.

<family name>

One to ten characters. Name of disk or pack. The name DISK can be used to specify the System disk and overrides the default pack associated with a usercode.

<first name>

One to ten characters. When the asterisk syntax is used, the asterisk character counts as one of the ten characters in the <first name>.

<second name>

One to ten characters.

<usercode>

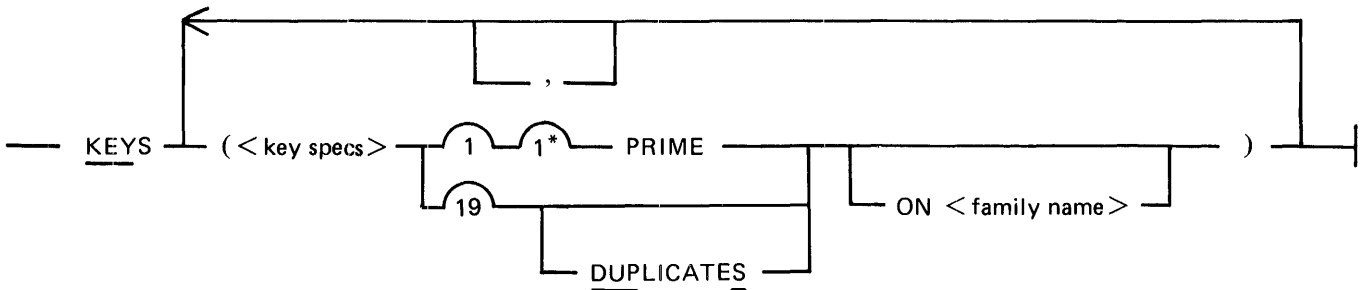
One to eight characters.

* (asterisk)

Used in a <file-name> to override the default usercode and the default pack <family name> associated with that usercode. It is only necessary to use an asterisk in a <file-name> when the program accessing the file is run under a usercode and the <file-name> consists of a <first name> only. Use a <family name> of DISK to override the default pack.

Syntax (<key syntax>):

<key syntax>



<key specs>



Semantics (<key syntax>):

<offset>

Refers to the relative displacement from zero of the first character of the key in bytes.

<length>

Refers to the length of the key in bytes.

NOTE

Keys are alphanumeric; each character is one byte.

ODT Interface Sample Executions

Two sample program control instructions follow:

```
EX CREATE/ISAM SW3 1
    <mix-number>AX FILE IN (TITLE=ACCOUNTS/PAYABLE ON USER),
        OUT (TITLE=ACCOUNTS/PAYABLEX), DATA (FAMILYNAME = USER),
        KEYS (4 5 PRIME) (20 10 DUPS) (32 5 ON USER2)
EX CREATE/ISAM;AX FILE IN (TITLE={usercode}MASTER ON PACKX)
        OUT (TITLE={usercode}NEWMMASTER)
        KEY ( 1 10 PRIME)
```

Card Interface

The card interface allows the user to enter the program specifications through the card reader. This interface is activated when program switch 2 is set (non-zero).

Specifications entered in this interface are free field and the program can accept up to eighty (80) characters of data per card.

All limits described in the terminal interface and the ODT interface apply in the card interface.

The syntax for the card interface is identical to that given for the ODT interface.

OUTPUT MESSAGES

Error messages and informative procedural messages are displayed by the CREATE/ISAM program. When these messages appear is slightly different depending on which interface is being used to enter the input specifications.

Terminal Interface Messages

When all specifications have been accepted by the CREATE/ISAM program, the following message is displayed:

```
BEGIN COPY OF <input-file name> TO ISAMFILE <output-file name>
```

The following message is displayed every 1000 records and when the copy is complete:

```
<number> records copied
```

If errors occur during the copy operation, a message is displayed. Depending on the nature of the error, program execution may be continued or terminated.

ODT Interface Messages

If errors occur in the syntax of the <ODT interface> entries, error messages are displayed after each AC or AX command entry. A blank AC or AX command must be entered before the errors can be corrected. Two successive blank AC or AX commands cause the program to terminate.

The copy operation does not begin until after the syntax is error-free and one blank AC or AX command has been transmitted. Errors that occur during the copy operation are displayed as they occur, with appropriate messages.

Error Conditions

Errors associated with the file screen or the key screen cause the screen to be re-displayed with the user's input in place and self-explanatory error messages displayed at the bottom of the screen.

Errors during the file copy are displayed when they occur. Depending on the error, the program may continue or terminate. For error conditions in this category, the messages displayed, the causes, and the actions to take are as follows:

***** WRITE ERROR 1 – UNEXPECTED END OF FILE, RECORD NUMBER N.**

The MCP returned an end-of-file condition which indicates a system failure of some type. The program terminates.

***** WRITE ERROR 2 – DUPLICATE KEY, RECORD NUMBER N, ENTER "OK" TO IGNORE RECORD AND CONTINUE.**

This condition is typically caused by a duplicate key in a field where duplicates are not allowed. If the error is entered with an AC or AX command, the program hangs, but resumes if AX OK is entered. Otherwise, the program terminates.

***** WRITE ERROR 6 – REQUESTED AREA IS GREATER THAN FILE ALLOWS, RECORD NUMBER N.**

This condition indicates the failure of the program to correctly generate output file specifications. The program terminates.

SECTION 10

CREATE/TABLE

The CREATE/TABLE program is a utility program that generates a translation table to be used by programs to translate from one character set to another. The translation table, in the form of a disk file, translates forward and backward from ASCII 8-bit, ASCII 7-bit, BCL, BCL 8-bit, and B500. All translation is to or from internal 8-bit EBCDIC. In addition, any other type of translation can be specified by the operator entering soft input.

TRANSLATION TABLE

The translation table file contains three records: a translate header, a forward translation table, and a backward translation table. Record sizes vary according to the character set that generates the table. Table 10-1 lists the character sets and the record sizes for each.

Table 10-1. Character Sets and Record Sizes

Character Set	Default Bits Per Record
ASCII 8-bit	2048
ASCII 7-bit	1792
BCL	1536
BCL 8-bit	2048
B500	1536
EBCDIC	2048
Soft input	2048

TRANSLATE HEADER FORMAT

The translate header describes the forward and backward translation tables. Eight fields of information are contained in the beginning of the header.

1. A 24-bit level identifier, which changes for each system software release. It consists of the numbered day and year of the first release of that level, with the last two digits of the year listed first. For example, 79037 represents the year 1979 and the 37th day of the year.
2. An 8-bit forward translation entry size, giving the size of each entry in the second record.
3. An 8-bit backward translation entry size, giving the size of each entry in the third record.
4. A 24-bit forward translation table size for the second record.
5. A 24-bit backward translation table size for the third record.
6. A character table type containing TRAN.
7. A 24-bit field containing the filler character (in hexadecimal, right-justified) associated with the second record. This character is @40@ for EBCDIC.
8. A 24-bit field containing the filler character (in hexadecimal, right-justified) associated with the third record. Following are the filler characters and the associated character sets:

@10@	BCL
@40@	BCL 8-bit
@20@	ASCII 7-bit
@A0@	ASCII 8-bit

OPERATING INSTRUCTIONS

The CREATE/TABLE program is executed through card or disk input. Figure 10-1 illustrates a sample card deck with control and data cards that produces an ASCII 8-bit translation table. Figure 10-2 shows disk file input including the necessary file-equate statement. All data must appear within the disk file.

Program Input

The available options allow an operator to enter as little information as is required to indicate the desired translation table (see figure 10-1) or, through soft input, as much information as is required to create an entirely new table.

Figure 10-3 depicts a sample card deck utilizing all of the available options. The soft input cards provide the translation information for a table that translates from lower-case letters to upper-case letters. Note the use of the IDNT statement to name the table TRANSLATE/MYTEST.

Program input takes the form of statements entered in conjunction with the control cards. The statements are in free-form format. Comments may be entered beginning in column 73.

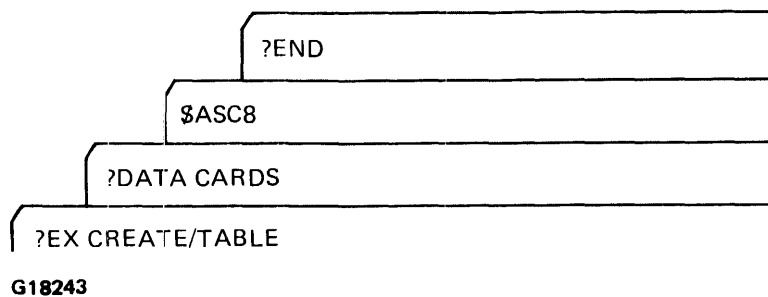


Figure 10-1. Sample Card Deck Execution .

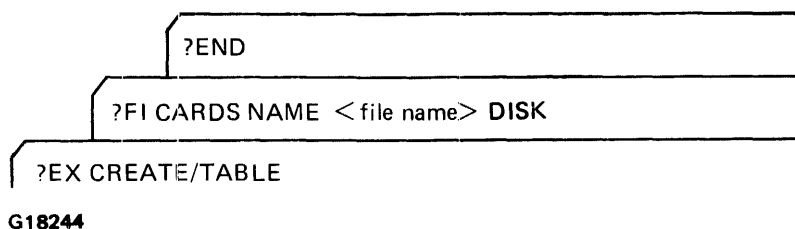


Figure 10-2. Sample Disk File Execution

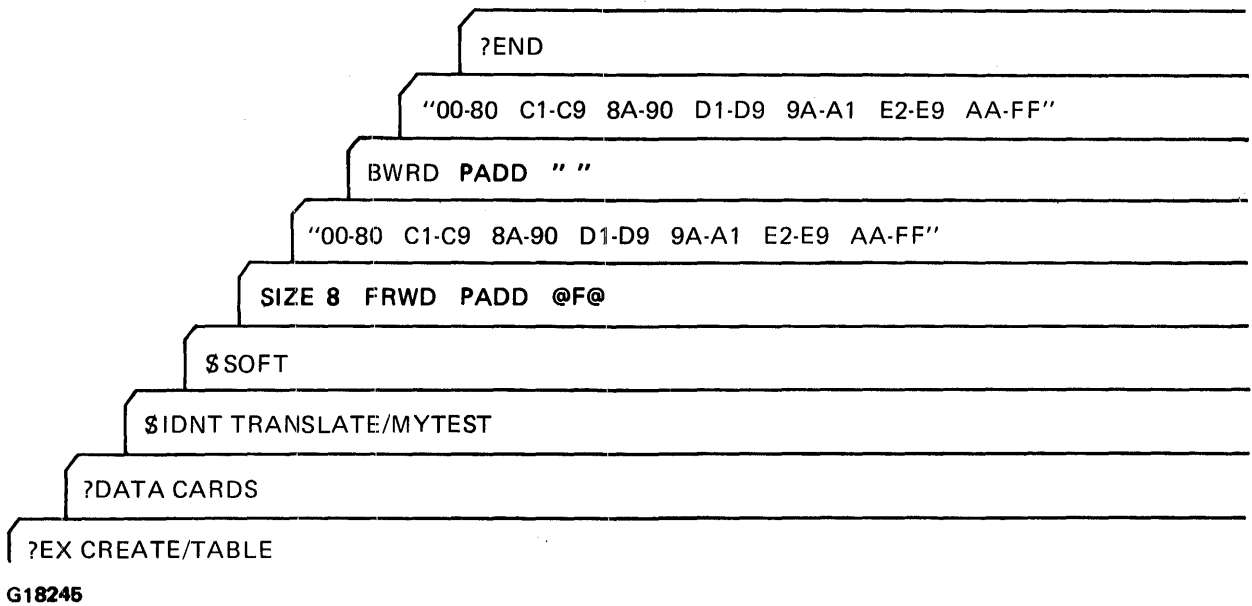
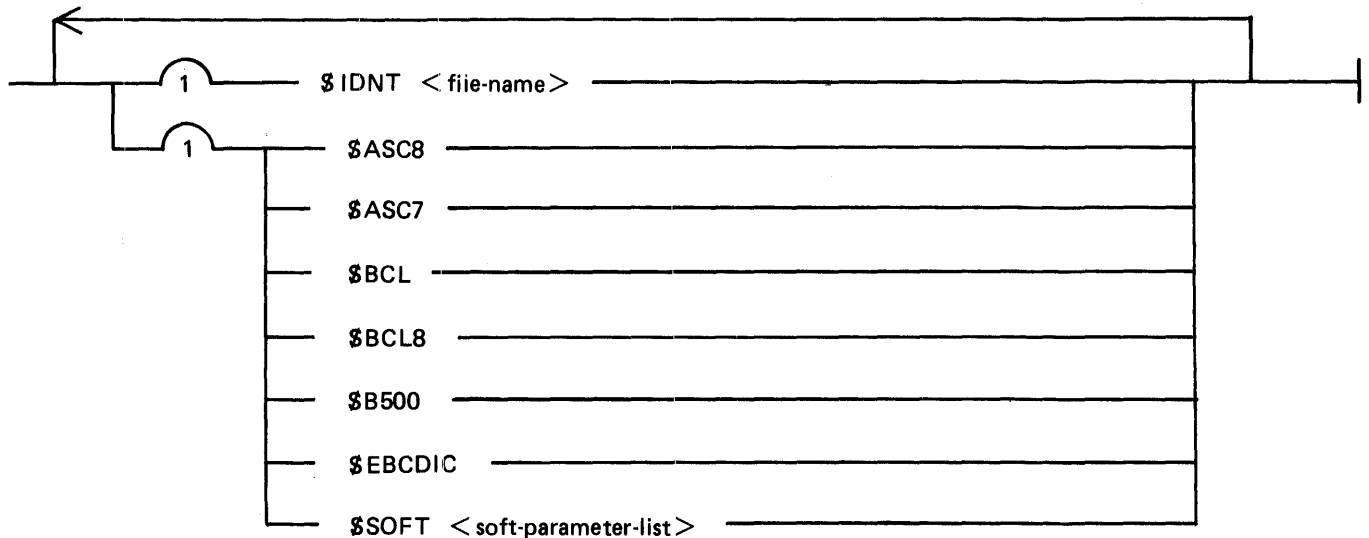


Figure 10-3. Sample Deck Using All Options

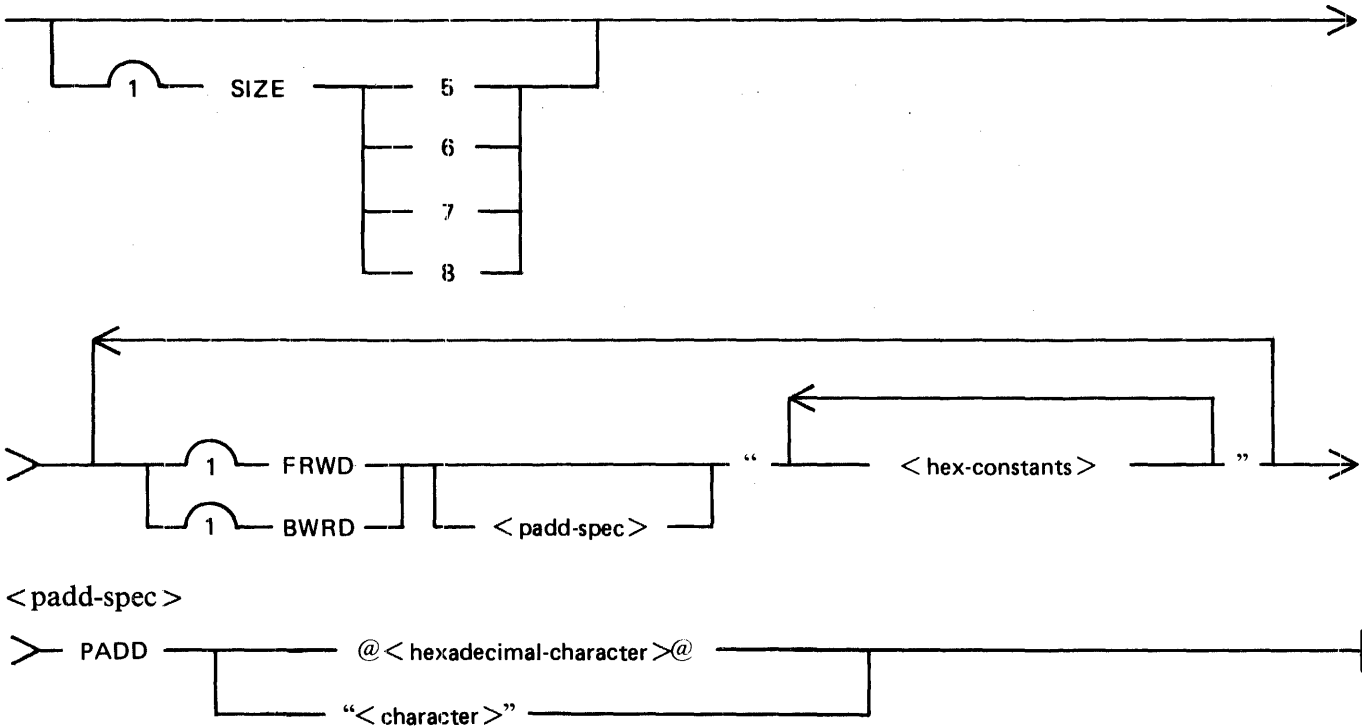
Option Summary

The syntax for the CREATE/TABLE program control statements is shown in the following diagram.

Syntax



<soft-parameter-list>



Semantics

<hexadecimal-character>

This field may be any character 0 through 9, or A through F.

<hex-constants>

This field can be any pair of hexadecimal characters 0 through 9, A through F. For example, 00-80 are a pair of hexadecimal characters. Notice that a single hexadecimal character (0) may be used as part of the hexadecimal pair as well as a double hexadecimal character (80).

<character>

This field can be any character A through Z.

\$ASC8

Produces a translation table for the ASCII 8-bit character set. If the IDNT statement is not used, the name of this table is TRANSLATE.

\$ASC7

Produces a translation table for the ASCII 7-bit character set. If the IDNT statement is not used, the name of this table is TRANSLATE.

\$BCL

Produces a translation table for the BCL character sets. If the IDNT statement is not used, the name of this table is TRANSLATE.

\$BCL8

Produces a translation table for the BCL 8-bit character set. If the IDNT statement is not used, the name of this table is TRANSLATE.

\$B500

Produces a translation table for the B500 character set. If the IDNT statement is not used, the name of this table is TRANSLATE.

\$EBCDIC

Produces a translation table for the EBCDIC character set. If the IDNT statement is not used, the name of this table is TRANSLATE.

PADD

The PADD option specifies the PAD bits for each table when the buffer is larger than the record size.

SIZE

The SIZE option specifies the size, in bits, of each unit of information to be translated to 8 bit on input or from 8 bit on output. The integer following the SIZE keyword can be 5, 6, 7, or 8. The default value is 8.

\$IDNT

The identification option allows the operator to name the translate table. The semantics for the file name are in accordance with the B 1000 file-naming conventions. If this statement is not used, the table is called TRANSLATE, and appears on the system disk as such. The MCP II assumes that translate files are on the system disk with the multiple-file identification TRANSLATE.

\$\$SOFT

The Soft Input option indicates to the program that translation information follows. The translation information must be entered on a separate line after \$\$SOFT. The direction forward (FRWD) or backward (BWRD) is specified and is followed by input enclosed in quotation marks ("). The input consists of pairs of hexadecimal constants listed individually or indicated as lower and upper limits joined by a hyphen. Figure 10-3 illustrates the use of this option.

ERROR MESSAGES

CONTROL INFORMATION NOT FULLY CONTAINED WITHIN COLUMN 72

Any information entered in column 72 and beyond was not recognized by the program.

ERROR IN HEX-CONVERSION

The input was entered incorrectly, perhaps in decimal form, and could not be processed.

INPUT FOR TRANSLATE TABLES NOT PROVIDED. CANNOT CONTINUE.

Either no input statement was provided or the information for the soft input translation table was omitted. An example of an input statement is \$ASC8.

INVALID CONTROL STATEMENT

The control statement contains an invalid control word or incorrect syntax.

INVALID FILE NAME

Ensure that file name in the IDNT statement conforms with B 1000 file-naming conventions.

PARAMETER LIST MUST BE ENCLOSED IN QUOTE SIGNS

Quotation marks were omitted from hexadecimal input.

SOFT STATEMENT DOES NOT CONTAIN ALL REQUIRED ENTRIES

Directional indicators FRWD or BWRD were omitted from the soft input, or hexadecimal sequences were missing.

VALUES ON BOTH SIDES OF HYPHEN ARE EQUAL AND DO NOT DEFINE A SERIES

No translation is requested.

INTERNAL FILES

The following list specifies the function of each file used by the CREATE/TABLE program.

Internal File Name	Function
REPORTFILE	Stores the printer file as it appears in the input for the FRWD translate table.
TRANFILE	A disk file that stores the translate tables.
CARDS	Stores the input file.

SECTION 11

DC/AUDIT

The DC/AUDIT program reads a data communications audit file and produces a readable printed version.

OVERVIEW

Datacomm audit files are created by the network controller or by data communications programs that perform their own data communications input and output functions. Examples of such programs are the Burroughs Network Architecture (BNA) System, the Remote Job Entry (RJE) program, and the HASP program. (The term "network controller" as used in this section refers to any of these programs.)

The DC/AUDIT program reads audit files and produces printouts of their contents. It can be set up to do this while the network controller program is writing the file (on line) or after the writing has ended (off line).

THE AUDIT CAPABILITY

The audit capability is a network controller option that can be set and reset through the ODT.

Writing The Audit File

When the audit option is set, the network controller writes the data communications I/O information to a circular audit file on disk. When the audit file has reached the end of its space, the oldest record is overwritten with the new record. This procedure optimizes disk space while allowing the user to monitor all data communications activity. If the network controller aborts, a record of the most recent data communications activity is retained on disk.

Restrictions

Ordinarily, only one program at a time may write to a given audit file. However, the MCP file-equate feature can be used to give a unique name to the external file name that the network controller associates with the audit file. In this way, for example, the NDL network controller and HASP can each be made to create an audit file of its own data communications activity simultaneously with the other. Then, the DC/AUDIT program may be directed to process the information in the uniquely named disk files.

Structure of the Audit File

The audit file contains 180-character records: 131 characters of text data and 49 characters for the time, I/O descriptor, and control fields needed by the DC/AUDIT program. The network controller breaks each data communications message into 131-character records, creates the 180-character records, and writes them to disk.

Audit File Blocking

Each physical write operation performed by the network controller tends to decrease its response time. The number of physical write operations can be reduced by using a **MODIFY** statement to change the **RECORDS.BLOCK** file attribute of the audit file in the network controller. The number to be used for the **RECORDS.BLOCK** file attribute is obtained by dividing the largest data communications message size by 131 and adding 1 to the quotient. Increasing the **RECORDS.BLOCK** requires more memory but eliminates the need for the network controller to wait for the completion of a physical write that takes place after each logical write operation. Therefore, because the message is written to memory, no time is lost waiting for the write to complete.

The physical write operation occurs after the network controller writes the last message into the block. If, for example, the **RECORDS.BLOCK** is modified to 10, the network controller performs ten logical write operations; then, one physical write operation is performed by the MCP.

Audit File Buffering

If the audit file is not buffered, the network controller frequently must wait for each physical write operation to complete before it can continue processing. The file attribute **BUFFERS** can be modified with the **MODIFY** statement. The recommended number of buffers is 2. (Refer to volume 1 of the *B 1000 Systems System Software Operation Guide* for a complete description of the **MODIFY** statement and the file attributes **BUFFERS** and **RECORDS.BLOCK**.)

DC/AUDIT OPERATING PROCEDURES

In general, the **DC/AUDIT** program is executed with a set of default run-time parameters that control certain aspects of output handling. The user can cause some or all of the default parameters to be overridden by entering the overrides alone or with other desired parameters. Default parameter overrides and other run-time parameters may be entered through the **ODT**, from disk, or from a card reader (80 or 96 column).

Parameters are described in the next major subsection.

The **DC/AUDIT** program is executed from the **ODT** with the following command:

```
EXECUTE DC/AUDIT
```

On beginning execution, the **DC/AUDIT** program looks for a disk file named **DC/PARAMETERS**. (The program ignores this file if program switch 0 is set to a value other than 0.) This disk file is predefined by the user to specify the desired run-time parameters including, if needed, any overrides of the default parameters. If this disk file is found and is error-free, processing begins.

If the file is in error, the appropriate messages are returned and the program then waits for an alternate entry of parameters.

Parameter Entry Alternatives

If program switch 0 equals 0, or if the file **DC/PARAMETERS** is not found, or if there are errors in that file, the following message is displayed on the **ODT**:

```
DEFAULT PARAMETERS?
```

The responses to this message are described next.

<mix-number> AX YES

This response is given to specify the use of the entire set of default parameters that is included in the program. (The default parameters are described along with all the available parameters in a later subsection titled Parameters.)

<mix-number> AX R

This response is given to specify that the program is to read parameters either from cards or from disk.

If you are using this method to enter the parameters from a card file (80-column or 96-column), the card reader should be loaded and ready. The predefined card file CARDS is labeled DC/PARAMETERS.

Example: <mix-number> AX R FILE CARDS NAME DC/PARAMETERS

If you are using this method to enter the disk file, it is assumed that the file (named DC/PARAMETERS) is now on a pack and ready to be read. Simply enter OU <DP?> after your AX R response.

Example: <mix-number> AX R OU DPA

The format rules for card and disk records is given in the next subsection (Parameter Entry Format).

<mix-number> AX <parameter-list>

This response is given if you want to enter parameters directly from the ODT. The <parameter-list> follows the format rules given in Parameter Entry Format, below.

Unless you follow your last entry with END, you get the following prompt after each parameter-list is transmitted and accepted as correct:

ENTER FURTHER PARAMETERS OR END

Parameter Entry Format

In all three methods of parameter entry, the file or entry format is quite rigid. Parameter keywords must be identical to the way they are presented in the Parameters subsection, with no spaces within keywords, at least one space between keywords, no more than 80 characters including spaces per record, and no keyword crossing a record boundary.

Parameter Entry Error Message

The message

INVALID PARAMETER DETECTED

is displayed if any parameter list entered contains an error. Following the error message, the DC/AUDIT program also displays the last group of characters it scanned. This may be of value in locating the error. The entire parameter list must be re-entered if any error occurred.

After the parameter-entry sequence is complete, the DC/AUDIT program attempts to access the audit file. If no audit file is present, the following message is displayed:

FILE DC/AUDITFILE NOT PRESENT

The program automatically goes to ^{end of job}EOJ (EOJ).

Processing

If the audit file is present, the DC/AUDIT program reads record zero and begins its output, following the specifications of the run-time parameters.

ODT Messages

Following is an alphabetized list of the messages that may be displayed on the ODT while the DC/AUDIT program is executing. The event that caused the message and other pertinent information is included with each message.

DEFAULT PARAMETERS?

The system operator must respond to this message by entering R, YES, or the parameters.

FILE DC/AUDITFILE NOT PRESENT

The audit file labeled DC/AUDITFILE is not in the disk directory. The DC/AUDIT program goes to EOJ.

INVALID MESSAGE TYPE

An error was detected on an attempt to dynamically modify one or more parameters. The additional characters that appear following this message are the last characters that were scanned by the DC/AUDIT program before the error was encountered.

INVALID PARAMETER DETECTED

An error was detected while analyzing the parameters. The additional characters printed following this message are the last characters that were scanned by the DC/AUDIT program before the error was encountered.

INVALID RECORD TYPE RECEIVED

A record whose record type was not equal to 1, 2, or 3 was read from the audit file. The DC/AUDIT program produces a dump file of itself, prints a summary report, and goes to EOJ.

NO RECORDS PRESENT IN FILE

The audit file contains no records. The DC/AUDIT program goes to EOJ.

PARAMETER FILE ON DISK NOT VALID

The disk file labeled DC/PARAMETERS is in error. To continue, the parameters must be entered from the card reader or the ODT. The additional characters printed following this message are the last characters scanned by the DC/AUDIT program before the error was encountered.

PARITY ERROR ON DISK FILE DC/PARAMETERS FILE NOT USED

The DC/AUDIT program detected a parity error while trying to read the disk file containing the parameters. To continue, the run-time parameters must be entered from the card reader or the ODT. The additional characters printed following this message are the last characters scanned by the DC/AUDIT program before the error was encountered.

UNEXPECTED "EOF" FOUND ON AUDIT FILE

An end of file was detected on the read of the first record in the audit file. A summary report is printed and the DC/AUDIT program goes to EOJ.

TYPE 2 RECORD OUT OF SEQUENCE

A type 2 record is not followed by a type 1 record or another type 2 record. A summary report is printed and the DC/AUDIT program goes to EOJ.

Termination

The way the DC/AUDIT program completes execution depends on whether it is processing off line (no auditing in process) or on line (auditing in process).

In the off-line mode, it goes to EOJ after it has processed the entire file. The last file record is identified in record zero of the audit file.

In the on-line mode, if one minute passes with no new entries, the DC/AUDIT program interrogates the audit file. If it finds that no other program is using the file (evidence that the generating program has gone to EOJ), it goes to EOJ.

Entry of the system command ST also causes the program to terminate.

Input Commands

The following system commands may be entered through the ODT for the purposes described while the DC/AUDIT program is executing:

<mix-number>AX WY

(Program status.) This causes DC/AUDIT to display a summary accounting of the records processed (read and written).

<mix-number>AX TO

(Display options.) This causes DC/AUDIT to display the current settings of all parameters.

<mix-number>AX <parameter-list >

Entering parameter settings in this way while the program is running changes (sets, resets, or adds to) current parameters.

The TIME parameter is not accepted.

If the HEADER parameter is set, certain other parameter changes will cause DC/AUDIT to skip to the top of a new page when the new parameters take effect.

The LINES=n parameter can be used to add new lines to be checked, but if deletion of lines is desired, the ALL.LINES parameter must be entered first, then the LINES=n parameter with the new line list (not just the lines to be added).

<mix-number>AX ST

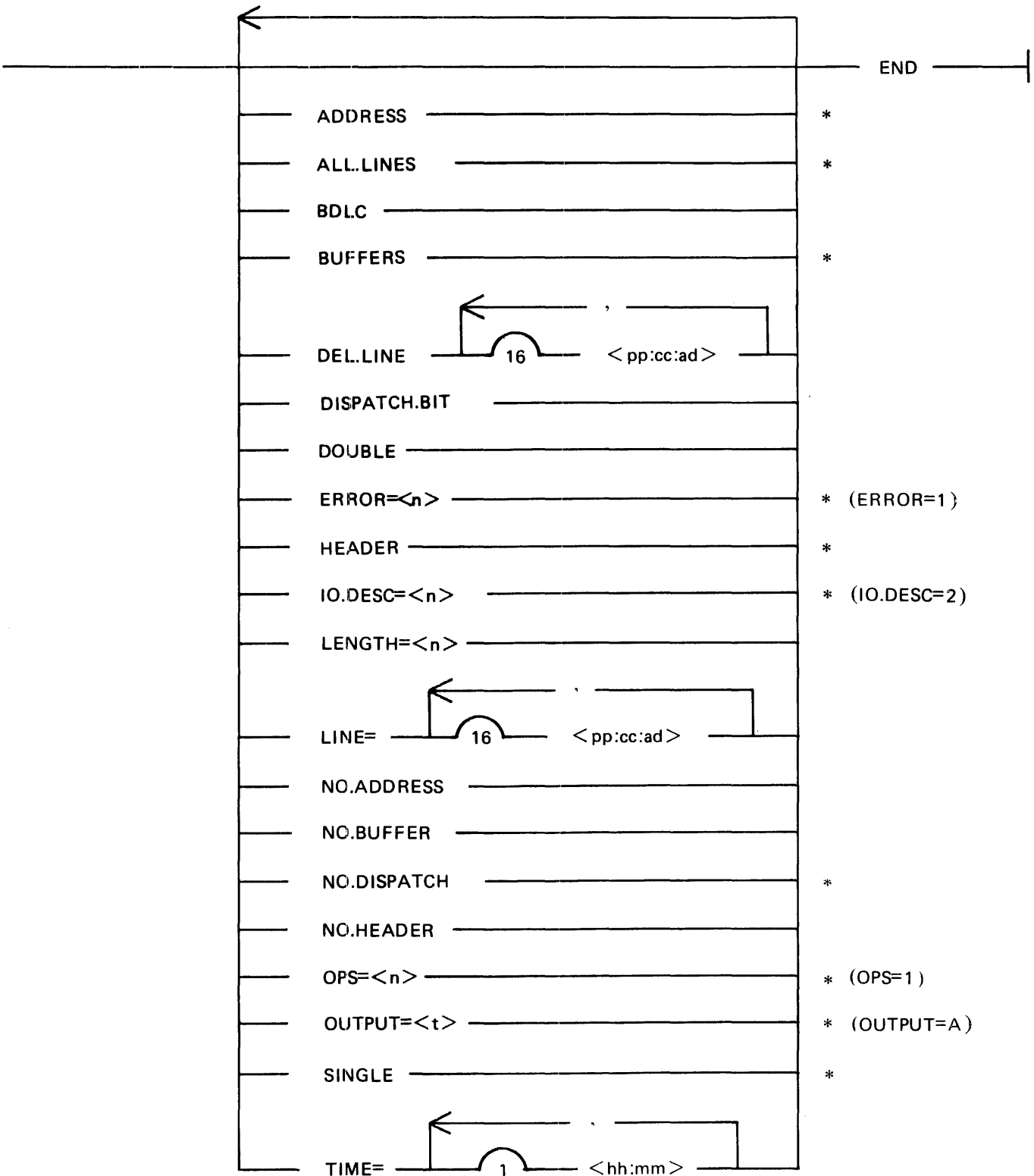
(Stop.) Causes DC/AUDIT to print a summary accounting of records processed and then go to EOJ.

PARAMETERS

The DC/AUDIT program formats the audit output in different ways and performs certain other tasks depending on which of the following parameters are selected for use during execution. (Asterisks (*) refer to default parameters.)

Default parameters may be considered as the baseline set; they deal with the functions most likely to be wanted. Each default parameter may be overridden, however, if the user wants different output forms.

Syntax



* Default parameters.

Semantics:

ADDRESS

Causes printing of the port, channel, and adapter numbers to which the data communications I/O descriptor was dispatched.

This is a default parameter. (NO.ADDRESS is the negation.)

ALL.LINES

Causes printing of all audit file records from all data communications lines.

This is a default parameter. (LINE permits selection; DEL.LINE permits dynamic deletion.)

BDLC

Causes printing of the text data and the result descriptors that are associated with the data communications I/O descriptor. Printing is in hexadecimal format for easy interpretation by BNA and SNA users.

Default: Do not print in hexadecimal format.

BUFFERS

Causes printing of the text data that is associated with the data communications I/O descriptor.

This is a default parameter. (NO.BUFFER is the negation)

DEL.LINE = <pp:cc:aa,pp:cc:aa,...>

This parameter may be entered while the DC/AUDIT program is executing to stop the monitoring of selected lines. Entries are made in the same way as is shown for the LINE parameter later in this list.

Default: Not applicable.

DISPATCH.BIT

Causes the output to specify which of the following two ways was used to dispatch an operation: (1) by network controller dispatch of the I/O descriptor, or (2) by resetting the operation-complete bit of the data communications I/O descriptor.

Default: NO.DISPATCH

DOUBLE

Causes double spacing of the records in the output listing.

Default: SINGLE

ERROR = <n>

The value given to <n> specifies the way records that contain indications of data communications errors are printed. (Note that there are no spaces between the parameter name, the equal sign, and the value.)

ERROR=1 causes printing of all audit records in the audit file. (This is the default condition.)

ERROR=2 causes printing of only the audit records that completed with exceptions.

ERROR=3 causes printing of only the audit records that completed without exceptions.

ERROR=aaaaaa provides a hexadecimal mask (aaaaaa) to use for searching result descriptors for records to print. The mask is converted to a 24-bit binary representation that is used to specify the operation to monitor. The bits are numbered 0 through 23 (high order to low order). If the bit that matches a list item is reset (0), the corresponding operation is not monitored; if the bit is set (1), operation is monitored.

Bit	Description
0	Operation complete.
1	Exception.
2	Reserved.
3	Parity error on read.
4	Memory access error on read.
5	Memory access error on write.
6	Timeout.
7	Break received during a write.
8	No ending control character received.
9	Linking terminated.
10	Reserved.
11	Loss of data set ready.
12	Loss of carrier.
13-24	Reserved.

Default: ERROR = 1.

HEADER

Causes column headings to be printed at the top of each page.

This is a default parameter. (NO.HEADER is the negation.)

IO.DESC = <n>

The value given to <n> specifies the printing format of the data communications I/O descriptor field. (Note that there are no spaces between the parameter name, the equal sign, and the value.)

IO.DESC=1 causes the descriptor to be printed in hexadecimal format with a blank between each of its fields.

IO.DESC=2 causes the descriptor result field to be interpreted as a mnemonic and printed. If more than one exception condition has occurred, only the first exception is printed.

Default: IO.DESC=2

LENGTH = <n>

<n> signifies a decimal number one to four characters long. The value given to <n> specifies the printing of the length of the buffer in bytes. (Note that there are no spaces between the parameter name, the equal sign, and the value.)

Example: LENGTH=400.

Default: Length is the entire buffer.

LINE = <pp:cc:aa,pp:cc:aa,...>

Defines specific data communications lines to be audited. In the term LINE = <pp:cc:aa,pp:cc:aa, ... >, pp identifies the multiline or single-line control port, cc identifies the data communications line adapter channel (00 for multiline controls), and aa identifies the data communications line adapter address (00 for single-line controls) Each value must be a pair of decimal digits, the colon characters (:) are required, and multiple entries may be entered separated by commas. No embedded spaces are permitted.

Up to 16 lines may be specified. A second record is needed to specify more than 8 lines.

To add to the list of lines under audit, the LINE parameter is entered again with the additional lines specified. To change the list of lines that are under audit, all lines are reestablished with the ALL.LINES parameter. Then, the LINE parameter may be used to enter a revised line list.

Examples:

```
LINE = 01:00:01, 01:00:14,07:14:00  
LINE = 03:00:03
```

Default: ALL.LINES

NO.ADDRESS

Suppresses printing of the address to which the data communications I/O descriptor was dispatched.

Default: ADDRESS

NO.BUFFERS

Suppresses the printing of the contents of the data communications buffer.

Default: BUFFERS

NO.DISPATCH

Suppresses printing of the manner in which the data communications I/O descriptor was dispatched.

This is a default parameter. (DISPATCH.BIT is the negation.)

NO.HEADER

Suppresses the printing of a heading at the top of each page.

Default: HEADER

OPS = <N>

The parameter OPS = n controls the printing of data communications operation codes (op codes). Note that there are no spaces between the parameter name, the equal sign, and the value.

OPS = 1 causes the printing of all op codes. This is the default condition.

OPS = 2 causes the printing of the Read and Write op codes only.

OPS=aa provides an 8-bit mask (specified by the hexadecimal values aa) that is used to specify which op codes to monitor and print. The bits are numbered 0 through 7, high order (left) to low order. If the bit value is 0, the corresponding op code is not printed; if it is 1, the op code is printed. Op codes are assigned to bits as follows:

Bit	Op Code
0	Read
1	Write
2	Break
3	Test
4	Stop
5-7	Reserved

Examples:

OPS=00 = 0000 0000 – Monitor no ops.

OPS=E8 = 1110 1000 – Monitor Read, Write, Break, and Stop.

OPS=C0 1100 0000 – Monitor Read and Write.

Default: OPS=1.

OUTPUT= <t>

<t> can take the value A or H.

OUTPUT=A specifies the use of a translation routine that

- (1) Converts special line control character codes into the appropriate names, and prints them enclosed in broken brackets (< >).
- (2) Converts non-graphics into the equivalent hexadecimal characters for each four-bit group and prints them enclosed in square brackets.
- (3) Converts graphics into the equivalent characters and prints them without enclosures.

Example: @01C1C2C33503@ is printed <SOH>ABC[35]<ETX>

OUTPUT=H prints the hexadecimal code.

Example: @01C1C2C33503@ is printed 01C1C2C33503

Default: OUTPUT=A

SINGLE

The SINGLE parameter causes single spacing of the printed output.

Default: DOUBLE

TIME= <hh:mm[,hh:mm]>

The TIME parameter allows a start time and, optionally, an end time to be specified for printing from the audit file. Any record entered before the start time or after the end time is ignored. Time refers to the time of creation of an audit record. Either TIME= <hh:mm> or TIME= <hh:mm, hh:mm> may be used (hh = hours, mm = minutes).

TIME = <hh:mm> is used to specify start time only. Output is expected to continue indefinitely or as long as the DC/AUDIT program is executing. TIME = <hh:mm, hh:mm> is used to specify start time and end time.

If TIME is not specified, output begins with the oldest record in the audit file and continues as long as the DC/AUDIT program is executing.

PRINTER OUTPUT FORMATS

Any of four basic line printer formats may be used by the DC/AUDIT program to print audit files, depending on parameter selection. The formats depend on how the I/O result descriptor is interpreted and printed and whether or not data read from the data communications operation is printed.

Format	I/O Descriptor	Data
1	mnemonic	printed
2	mnemonic	not printed
3	hexadecimal representation	printed
4	hexadecimal representation	not printed

When an I/O descriptor record is being printed in any of these formats, columns 12 through 18 on the listing indicate the type of I/O descriptor. There are four basic types of I/O descriptors, with a number of variants possible for each:

Descriptor	Variants
READ	RD.NTIM (Read, no time out) RD.APOL (Read, autopoll) RD.NEOT (Read, no EOT) READ
WRITE	WR.APOL (Write autopoll) WR.NEOT (Write, no EOT) WRITE
STOP	DISCONN (Disconnect) BREAK STOP
TEST	TESTRNG (Test and wait for ring) TESTDSR (Test and wait for DSR) TEST

For READ and WRITE, if more than one variant is set, only one is indicated on the listing. The priorities of read and write variants are (high, left; low, right)

RD.NTIM, RD.APOL, WR.APOL, RD.NEOT, WR.NEOT, READ, WRITE

When logging a WR.APOL operation, a slash character (/) in the autopoll string indicates that the autopoll operation completed at the address immediately preceding the slash character. If no slash character is printed, the operation completed on the last address printed.

If a TEST I/O descriptor is processed, an additional field is printed (ADAPTER.ID = nn) indicating the type of adapter to which the descriptor was issued. The possible values of nn and their meanings follow:

- 00 No adapter present.
- 01 Standard direct connect asynchronous, or standard synchronous for leased data communications lines, or standard asynchronous for leased data communications lines.
- 04 Teletype direct asynchronous or Teletype asynchronous for leased data communications lines.
- 05 EBCDIC bisynchronous for leased data communications lines.
- 07 EBCDIC bisynchronous wideband for leased data communications lines.
- 10 Automatic calling unit (ACU).
- 11 Standard synchronous for switched data communications lines.
- 14 Teletype asynchronous for switched data communications lines.
- 15 EBCDIC bisynchronous for switched data communications lines.
- 17 EBCDIC bisynchronous wideband for switched data communications lines.

The data communications I/O descriptor is described in the *B 1860/B 1870 Systems Reference Manual*.

The four output formats are described in the paragraphs that follow and are illustrated in figures 11-1 through 11-4.

Output Format 1

Output format 1 is generated when the parameters IO.DESC=2 and BUFFERS are specified.

If the HEADER parameter is specified, the header format of each page is as shown in the upper part of figure 11-1. Records from the audit file are printed in the format shown in the lower part of figure 11-1. Each field in a record is defined in the text that follows the figure.

```
***** B 1000 DC/AUDIT <date time> *****  
  
TIME   OP           BUFFER           RESULTS  
  
      Header Format  
  
hh:mm:ss.t aaaaaaa data pp:cc:aa bbbbb LENGTH=nnnn ccccc  
  
      Record Format
```

Figure 11-1. Header and Record Formats, Output Format 1

hh:mm:ss.t

Time in hours, minutes, seconds, and tenths. For a read, time represents read completion time. For all other operations, time represents start time.

aaaaaaa

Type of I/O operation.

data

Data written or read. If OUTPUT=A, data consists of graphics, special line control characters with their names enclosed in broken brackets (<>), and non-graphics with each 4-bit group in hexadecimal representation enclosed in brackets ([]). The characters "[" and "]" are treated as non-graphics.

Example: <STX>[EC]ABC12<ETX>

If OUTPUT=H, then each 4-bit group of data is in hexadecimal representation.

Example: 01ECC1C2C3F1F2203

pp:cc:aa

Port, channel, and adapter to which the I/O descriptor was issued. (Included if ADDRESS parameter is set.)

bbbb

Appears if the DISPATCH.BIT parameter is set. If I/O was initiated by resetting the I/O complete bit, RESET is printed. If I/O was initiated by a dispatch communicate to the MCP, DISPATCH is printed.

LENGTH=nnnn

The length of the data read or written.

cccc

The mnemonic interpretation of the manner in which the I/O operation completed. Refer to the subsection of this manual titled I/O Completions for a complete description of the completions.

Output Format 2

Output format 2 is generated when default parameters IO.DESC=2 and NO.BUFFERS are both specified. If the HEADER parameter is specified, the header format of each page is as shown in the upper part of figure 11-2. Records from the audit file are printed in the format shown in the lower part of figure 11-2.

The field descriptions given for output format 1 are valid for output format 2.

***** B 1000 DC/AUDIT <date and time> *****

TIME OP RESULTS

Header Format

hh:mm:ss.t aaaaaaa pp:cc:aa bbbbb LENGTH=nnnn ccccc

Record Format

Figure 11-2. Header and Record Formats, Output Format 2

Output Format 3

Output format 3 is generated when the parameters IO.DESC=1 and BUFFERS are both specified. If the HEADER parameter is specified, the header format of each page is as shown in the upper part of figure 11-3. Records from the audit file are printed in the format shown in the lower part of figure 11-3.

Field descriptions for output format 3 follow the figure.

```
***** B 1000 DC/AUDIT <date time> *****  
TIME   OP     BUFFER           DESCRIPTOR  
  
      Header Format  
  
hh:mm:ss.t  aaaaa data AE=xxxxxx RS=xxxxxx LK=xxxxxx OP=xxxxxx SA=xxxxx  
              EA=xxxxxx MS=xxxxxx xxxxxx xxxxxx xxxxxx xxxxxx xx  
              LENGTH=nnnn  
              --->hexadecimal representation of I/O descriptor<--
```

Record Format

Figure 11-3. Header and Record Formats, Output Format 3

Each item or field is described as follows:

hh:mm:ss.t

Time in hours, minutes, seconds, and tenths. For a read, time represents read completion time; for all other operations, time represents start time.

aaaaaaa

Type of I/O operation.

data

Data written or read. If OUTPUT=A, data consists of graphics, special line control characters with their names enclosed in broken brackets (<>), and non-graphics with each 4-bit group in hexadecimal representation enclosed in brackets ([]). The characters "[" and "]" are treated as non-graphics.

AE

The actual ending address of the descriptor.

RS

The result descriptor.

LK

The link address to the next descriptor.

OP

The operation code.

SA

The start address of the descriptor.

EA

The ending address of the descriptor.

MS

The MCP work area.

LENGTH = nnnn

The length of the data read or written. If the ADDRESS parameter and/or the DISPATCH.BIT parameter is specified, the port, channel, adapter, and either the word RESET or DISPATCHED precedes LENGTH = nnnn.

Output Format 4

Output format 4 is generated when the parameters IO.DESC = 1 and NO.BUFFERS are both specified. If the HEADER parameter is specified, the header format of each page is as shown in the upper part of figure 11-4. Records from the audit file are printed in the format shown in the lower part of figure 11-4. The field descriptions given for output format 3 are valid for output format 4.

```

***** B 1000 DC/AUDIT <date time> *****
TIME                OP                DESCRIPTOR

      Header Format

hh:mm:ss.t  aaaaaa  pp:cc:aa  bbbbb  LENGTH=nnnn  AE=xxxxxxx  RS=xxxxxxx
LK=xxxxxxx  OP=xxxxxxx
SA=xxxxxxx  EA=xxxxxxx
MS=
-> hexadecimal <-
      value of the I/O
      descriptor

      Record Format

```

Figure 11-4. Header and Record Formats, Output Format 4

I/O COMPLETIONS

The result descriptor of each I/O descriptor is a 24-bit field that contains pertinent information regarding the manner in which the descriptor was terminated. The result descriptor is scanned from left to right and the scan is stopped when a true condition is detected. Following is a list, in scan sequence, of the bits at result descriptor locations 0 through 12, location 0 leftmost.

Bit Location	Value of cccc Field of Output Formats 1 and 2
0	COMPLETE or NOT COMPLETE
3	PARITY
4	MEMORY ACCESS ERR
5	MEMORY PARITY
6	TIMEOUT
7	BREAK
8	NO END
9	END CHAINING
11	LOSS OF DSR
12	LOSS OF CAR or LOSS OF CTS

The I/O result conditions are false if the bit is equal to zero, and true if the bit is equal to one. A description of each I/O completion follows.

BREAK

BREAK indicates that a break operation occurred during a write operation. The write operation is canceled and a new operation is started.

COMPLETE

COMPLETE indicates the end of either a **TEST**, **STOP**, **WRITE**, or **BREAK/DISCONNECT** data communications operation.

END CHAINING

END CHAINING indicates that one of the following conditions exists during a polling sequence:

1. If a link on a negative response is indicated and either a positive response is detected or an error occurs on a negative response, linking is terminated. This applies to a read operation.
2. If a link on a positive response is indicated and either a negative response is detected or an error occurs on a positive response, linking is terminated. This applies to a write operation.

If any errors are detected during a read operation, the linking ends and an interrupt is returned. If errors are not detected during a read operation, a link to the write operation occurs and an **ACK** is sent, followed by a link back to the read operation.

END CHAINING is the normal completion for a **READ AUTOPOLL** data communications operation.

LOSS OF CAR

LOSS OF CAR indicates that the data communications line adapter has detected a loss of carrier during a read operation due to either the carrier signal dropping or missing receive data. If bit 6 is set, a receive time-out has occurred. If bit 6 is not set, the adapter has detected a carrier drop and recovered if data is not lost. Data loss results in an incorrect **BCC**.

LOSS OF CTS

LOSS OF CTS indicates that the line adapter has detected a loss of clear-to-send during a write operation due to either the request-to-send signal dropping or a faulty data set. If bit 6 is set, the loss of clear-to-send has occurred during the initial time-out period of the **WRITE** operation.

LOSS OF DSR

LOSS OF DSR indicates that the data communications adapter has detected a loss of data-set-ready during a read operation. The time-out bit (number 6) is also set.

MEMORY ACCESS ERR

MEMORY ACCESS ERR indicates that the system could not respond to the single line control or multi-line control service request during a read operation. This is a result of either misassigning priorities of the network controller, excessive network speed, causing system overload, or a hardware/software malfunction.

MEMORY PARITY

MEMORY PARITY indicates that an incorrect parity was received from system during a write operation.

NO END

NO END indicates that an ending control character was expected but none was received during either a read or write operation. Normal message termination occurs, including the BCC.

NOT COMPLETE

NOT COMPLETE indicates the end of a WRITE AUTOPOLL data communications operation.

PARITY

PARITY indicates that a character parity error or a BCC error was detected during a read operation.

TIMEOUT

TIMEOUT indicates one of the following:

1. Data was not received during a specified period of time during a read operation.
2. The data communications line adapter detected either loss of carrier or loss of data-set-ready during a read operation or during the initial time-out period during a write operation.

DC/AUDIT MNEMONIC CHARACTERS

The valid special control characters and their codes are listed in table 11-1.

Table 11-1. Special Control Characters and Codes

Name	Code	Definition	Name	Code	Definition
NUL	00	Null	RS	1E	Record separator
SOH	01	Start of heading	US	1F	Unit separator
STX	02	Start of text	DS	20	Digit select
ETX	03	End of text	SOS	21	Start of significance
PF	04	Punch off	BYP	24	Bypass
HT	05	Horizontal tab	LF	25	Line feed
LC	06	Lower case	ETB	26	End of transmission block
DEL	07	Delete	ESC	27	Escape
SMM	0A	Start of manual message	SM	2A	Set mode
VT	0B	Vertical tab	CU2	2B	Customer use 2
FF	0C	Form feed	ENQ	2D	Enquiry
CR	0D	Carriage return	ACK	2E	Acknowledge
SO	0E	Shift out	BEL	2F	Bell
SI	0F	Shift in	SYN	32	Synchronous idle
DLE	10	Data link escape	PN	34	Punch on
DC1	11	Device control 1	UC	36	Upper case
DC2	12	Device control 2	EOT	37	End of transmission
DC3	13	Device control 3	CU3	3B	Customer use 3
RES	14	Restore	DC4	3C	Device Control 4
NL	15	New line	NAK	3D	Negative acknowledge
BS	16	Backspace	SUB	3F	Substitute
IL	17	Idle	POL	97	Poll
CAN	18	Cancel	SEL	98	Select
EM	19	End of medium	SEQ	99	Sequence
CC	1A	Cursor control	FSL	A2	Fast select
FS	1C	Field separator	BSL	A3	Broadcast select
GS	1D	Group separator			

SECTION 12

DISK/ALLOCATOR

The DISK/ALLOCATOR program is a normal-state utility program used to create Installation Allocated Disk (IAD) files. IAD files are assigned to an absolute disk address and have operator-specified physical attributes. The capability of designating a file's physical location and its attributes is useful for recovering files after a coldstart operation, for allocating files to specific drives of multiple system disk configuration, and for aligning areas of a file on disk cylinder boundaries.

OPERATING INSTRUCTIONS

The following paragraphs describe the operating instructions.

Execution

The DISK/ALLOCATOR program is executed either from the ODT or from cards. By default, the program expects to receive command specification input through AC or AX commands. When switch 0 = 1, DISK/ALLOCATOR accepts all command specification input from a card file labeled CARDS.

Command Specifications

Command specifications describe the location and the attributes of the file being created. All command input specifications are entered in free-form format as keyword/value pairs. Multiple command specifications can be entered in a single input message or card with each keyword/value pair separated by at least one comma (,) or blank character.

If an error is detected in any input specification, the file is not allocated, and the DISK/ALLOCATOR program scans the input stream for another TITLE specifier. All keywords encountered before the next TITLE specifier are ignored.

The equal sign (=) character must be included between each keyword and its associated value. Numeric values can be specified in either decimal or hexadecimal notation. Hexadecimal values must be enclosed in at sign (@) characters and must not include the hexadecimal mode indicator (4) preceding the number.

Valid attributes and their meanings are described in table 12-1.

Table 12-1. DISK/ALLOCATOR Attribute Keywords and Default Values

Keyword	Description	Default
TITLE	File-identifier	--
RECSIZE	Record size	180
BLOCKSIZE	Records-per-block	1
AREASIZE	Blocks-per-area	100
AREAS	Number of areas	25
LASTRECORD	EOF pointer	0
FAMILYINDEX	Disk drive or EU	0
AREANBR	Area number	--
AREAADDRESS	Absolute disk address	--

The TITLE attribute must be specified first for each file being allocated. The attribute keywords RECSIZE, AREASIZE, AREAS, and LASTRECORD are optional and can follow the TITLE keyword in any order, but must precede the attribute keywords FAMILYINDEX, AREANBR, and AREAADDRESS. When the optional keywords are not specified, their default values are assumed.

Once the general attributes of the file are established, each area and its absolute disk address are then specified through the keywords FAMILYINDEX, AREANBR, and AREAADDRESS. It is not necessary to allocate all areas of the file at once.

FAMILYINDEX specifies the electronic unit (EU) number for head-per-track disks, or the disk drive for multiple system packs/cartridges. The default is DRIVE/EU 0. Once FAMILYINDEX is specified, it must be specified for all of the following pairs of AREANBR and AREAADDRESS.

AREANBR and AREAADDRESS must be specified in pairs, with AREANBR appearing first. AREANBR specifies the file area number and can have a value of 1 to 105, inclusive. AREAADDRESS is the absolute disk address of the area. The AREAADDRESS is encoded as the rightmost 24 bits of a standard address field. The remaining high-order bits locate the disk and are not included in the AREAADDRESS used by the DISK/ALLOCATOR program.

The DISK/ALLOCATOR program does not actually allocate any areas of a file until it receives another TITLE specifier, which indicates that another file is to be allocated, or until it receives a blank AC or AX command from the ODT or an EOF from the card file. When the described disk file is allocated, the following message is displayed at the ODT:

<title> ALLOCATED

ALLOCATED FILE STRUCTURE

When a file is allocated, DISK/ALLOCATOR constructs a disk file header. Constructing this header requires one to three contiguous disk sectors, depending on the number of areas specified for the file (1 to 25 areas require one sector, 26 to 65 areas require two sectors, and 66 to 105 areas require three sectors). The DFH is allocated automatically by the MCPPII at the lowest available disk address. Thus, the location of the disk file header is not determined by the DISK/ALLOCATOR program.

All file attributes are determined by the keyword attribute specifications processed by DISK/ALLOCATOR.

SAMPLE EXECUTION

```
EX DISK/ALLOCATOR
  DISK/ALLOCATOR = 1234 BOJ....
  % DISK/ALLOCATOR = 1234 ENTER SPECS
1234AXTITLE = A/B/C
  %DISK ALLOCATOR = 1234 ENTER SPECS
1234AXRECSIZE=90,BLOCKSIZE=2,AREASIZE=100,AREAS=25
  % DISK/ALLOCATOR = 1234 ENTER SPECS
1234AXLASTRECORD=250
  % DISK/ALLOCATOR = 1234 ENTER SPECS
1234AXAREANBR = 1,AREAADDRESS=@003FCO@
  % DISK/ALLOCATOR = 1234 ENTER SPECS
1234AXAREANBR = 2,AREAADDRESS = @DDOC@
  % DISK/ALLOCATOR = 1234 ENTER SPECS
1234AX
  % DISK/ALLOCATOR = 1234 A/B/C ALLOCATED
  DISK/ALLOCATOR = 1234 EOJ....
```

ERROR MESSAGES

INVALID <keyword> -- <value>

The specified <value> is invalid for the <keyword>.

INVALID TOKEN -- <string>

The specified <string> is not a valid <keyword>.

EQUAL SIGN EXPECTED

A <keyword> was not followed by an equal sign.

VALUE EXPECTED AFTER =

No value was specified following the <keyword> and equal sign.

UNEXPECTED TOKEN -- <keyword> -- TITLE EXPECTED

The keyword TITLE was expected, but some other token was encountered. This message can occur if an error is detected in an input message that causes the program to scan for another TITLE keyword to continue processing.

CANNOT SET FAMILYINDEX TO <integer>

The FAMILYINDEX cannot be set to the specified <value>.

CANNOT ALLOCATE AREA -- NEED AREANBBR

An AREAADDRESS was specified without first specifying the AREANBR.

CANNOT ALLOCATE AREA <integer>

The requested disk area could not be allocated. An associated MCP II error message describes the reason for the failure.

LASTRECORD OF <integer> IS NOT WITHIN ASSIGNED FILE AREA

The value specified for LASTRECORD is beyond the maximum file size.

CANNOT SET LASTRECORD TO <integer>

The value specified for LASTRECORD is invalid.

SECTION 13

DISK/DUMP

The DISK/DUMP program is a stand-alone cassette utility program that copies data sector-by-sector from one disk to another. The dump is a sequential copy of the entire input disk. System and user disk packs as well as user disk cartridges are allowed to be copied, but the output disk is required to be of the same or larger capacity as the input disk. System disk cartridges are not allowed.

The DISK/DUMP program first checks the pack labels for validity and then performs a sector-by-sector copy of the data. Neither disk may have any removed (XDed) sectors. The user may instead wish to use one of the other full disk copy programs such as SYSTEM/DISK-DUMP or STANDALONE/DISK-DUMP, which perform a file-by-file copy. These programs produce a "squashed" output pack and can recover more of a corrupted pack. The SYSTEM/DISK-DUMP program can also perform a sector-by-sector full disk copy under MCP control allowing other programs to execute in a multi-programming environment.

All program operation instructions are entered from the Operator Display Terminal (ODT).

Two sets of procedures for program loading follow. The first is for B 1990 systems and the second is for all other B 1000 systems.

Program loading on B 1990 Systems

To load the program on B 1990 systems:

1. Halt the system by entering the MCP command HALT through the ODT and pressing XMT. For dual-processor systems, be sure both processors are halted.

NOTE

Instructions for using the B 1990 Soft Panel in this fashion are hereafter presented in either of the following forms:

Transmit <keyword-1> <keyword-2> ...

<keyword-1> <keyword-2> ... [XMT]

A looping condition could deny access to the ODT. In such cases, press the INTRPT push button. If the system still does not halt, press the HALT/RUN push button.

2. Place the cassette into the cassette reader and ensure that the tape rewinds to BOT.
3. Transmit CLEAR MTR GO.

The cassette automatically rewinds on B 1990 systems.

Program loading on other B 1000 Systems

To load the program on all other B 1000 systems:

1. Halt the system using either the INTRPT push button or the console INTERRUPT switch. If the INTERRUPT switch is used, return it to its initial position following the halt. If the system continues to run, press the HALT push button. If the system still continues to run, press the HALT and CLEAR push buttons simultaneously. For dual-processor systems be sure both processors are halted.
2. Place the cassette into the cassette drive and ensure that the tape rewinds to BOT.
3. Press (1) the CLEAR push button, (2) the MODE push button to obtain the MTR mode, and (3) the START push button.
4. After the bootstrap loader is read from the cassette tape, the processor halts. The L register must contain @AAAAAA@ at this time; if not, the cassette must be rewound and the procedure restarted from step 3, preceding.
5. Press the MODE push button to return to the RUN mode, then press the START push button. Reading of the cassette resumes. When the program is completely loaded, program operation begins.
6. Rewind the cassette tape by pressing REWIND. (Rewind is automatic on B 1900 systems.)

NOTES

If the tape does not automatically rewind when it is loaded into the cassette drive, press the REWIND push button. If the tape is wound completely onto the right spindle, remove the cassette and manually rewind it until the clear leader is no longer at the read gap.

A system halt with @000004@ or @0D0053@ in the L register or with a 1 in the PERP register, signifies a cassette error during program loading. Restart the procedure from step 1.

OPERATING INSTRUCTIONS

After the program is loaded from the cassette, the following messages are displayed on the Operator Display Terminal (ODT):

```
DISK DUMP - <mark-and-patch-level> <compile-date/time>  
ENTER INPUT DRIVE - <DC? or DP?>
```

After a valid response, the following messages are displayed in sequence:

```
ENTER OUTPUT DRIVE - <DC? or DP?>  
IS VERIFICATION REQUIRED? - <YES or NO>
```

After valid responses, the dump begins. Unrecoverable read or write errors will result in the termination of the dump. After the copy phase is complete and if verification has been requested, a compare phase will start with the message:

```
VERIFICATION BEGINS
```

After successful completion, the following message is displayed:

```
DUMP COMPLETE <input drive> <serial number>
      TO <output drive> <serial number>
ENTER INPUT DRIVE - <DC? or DP?>
```

At this point additional disks may be copied without reloading the program from the cassette or, if a blank is entered, the DISK/DUMP program terminates and the following message is displayed:

```
END DISK/DUMP
```

STATUS COMMAND

Entering a STATUS or ST command during the copy phase causes the following message to be displayed:

```
COPY APPROXIMATELY <percentage> % COMPLETE
```

Entering a STATUS or ST command during the verify phase causes the following message to be displayed:

```
VERIFICATION APPROXIMATELY <percentage> % COMPLETE
```

ERROR MESSAGES

NOTE

If a B 1990 system halts with an unrecoverable exception, one of the error messages listed below is displayed on page 2 of the ODT. The Maintenance Card switches the ODT display to page 1 to allow command line input. The operator should look to page 2 for the program error message. For those halts that indicate "press the START push button", GO may be transmitted from the command line of page 1. This is equivalent to pushing START on other B 1000 systems.

DISK ERROR - RESULT IN 'T'

The T register contains the result descriptor for the I/O operation. Pressing START causes 100 more retries.

DISK NOT READY <drive>

PLEASE CORRECT THIS PROBLEM AND THEN PUSH START

Make the disk ready and press the START push button.

I/O ERROR <drive> <disk address>

A data parity error, timeout, address parity error, extended result descriptor error, or address error has occurred and retrying the operation failed to correct the problem. An input error is retried 5 times. An output error is retried 5 times, rewritten, and then reread 5 more times.

INPUT SIZE LARGER THAN OUTPUT

This function is not allowed.

INVALID RESPONSE – TRY AGAIN

Self explanatory.

NO CART OR PACKS ON SYSTEM

Mount the disks, make them ready, and re-execute the program.

PACK LABEL BAD <drive>

Input disk: A copy will not be allowed.

Output disk: Initialize the disk before copying.

REMOVED SECTORS ON <drive>

Program terminates. Copying to or from a cartridge or pack with any removed sectors could produce an invalid output disk.

TEMP TABLE FILLED <input-drive>

Ready this input disk under MCP control to return the temporary table entries to the available table and try again.

WARNING – SERIAL NUM. MISMATCH – MULTI PACK FILE NOT USABLE

The input and output disk serial numbers do not agree; this makes the multi-pack files on the output disk unusable.

WRITE LOCKOUT <output-drive>

Reset the WRITE LOCKOUT button on the output drive, and then press the START push button.

SECTION 14

DISKETTE/COPY

The DISKETTE/COPY program is a normal-state utility program that copies data files to or from an industry-compatible mini-disk (diskette or floppy disk). Copying files between two diskettes is not allowed.

All communication with a diskette is accomplished through the DISKETTE/COPY utility program.

Each diskette can contain up to 19 files and can hold a maximum of 246,272 bytes of information. Each diskette has 74 addressable tracks, numbered 0 through 73. Each track contains 26 sectors, numbered 1 through 26, and each sector can hold a maximum of 128 bytes of information. Diskette track number zero is reserved for the volume identifier and the file directory. A diskette address consists of five digits: the first two digits are the track number, the third digit is always zero, and the last two digits are the sector number of the record addressed.

OPERATING INSTRUCTIONS

The DISKETTE/COPY program can be executed either from the ODT or from a card reader. Program switch settings determine the source of input specifications.

PROGRAM SWITCHES

The DISKETTE/COPY program recognizes program switches 0 and 2. When switch 0 = 1, DISKETTE/COPY reads its input specifications from the card file labeled SPEC; otherwise, input specifications are entered through the ODT. When switch 2 = 1, the DISKETTE/COPY program stops processing when a data error is encountered and requests that the operator enter an AX OK command and then continues. Otherwise, the program deletes the sector in error, displays an error message and continues processing.

INPUT SPECIFICATIONS

DISKETTE/COPY obtains its input specifications from AC or AX commands entered through the ODT if program switch 0 = 0, or from a card file labeled SPEC if program switch 0 = 1.

The DISKETTE/COPY program can perform five distinct actions using diskettes. The input commands can request each of these actions, except AUTOLOAD. The actions are summarized in the following list.

AUTOLOAD

Automatically load all PSR pseudo reader files on a diskette to the system disk when the diskette is readied. This task cannot be requested with input commands.

COPY

Copies files to or from diskettes.

KA

Generates a directory listing of all files on a diskette. This listing is in the standard KA format.

PURGE

Purges all files on a diskette, converting it to a scratch diskette.

RELABEL

Relabels the diskette (change its volume identifier).

Detailed descriptions of the commands and the AUTOLOAD facility follow.

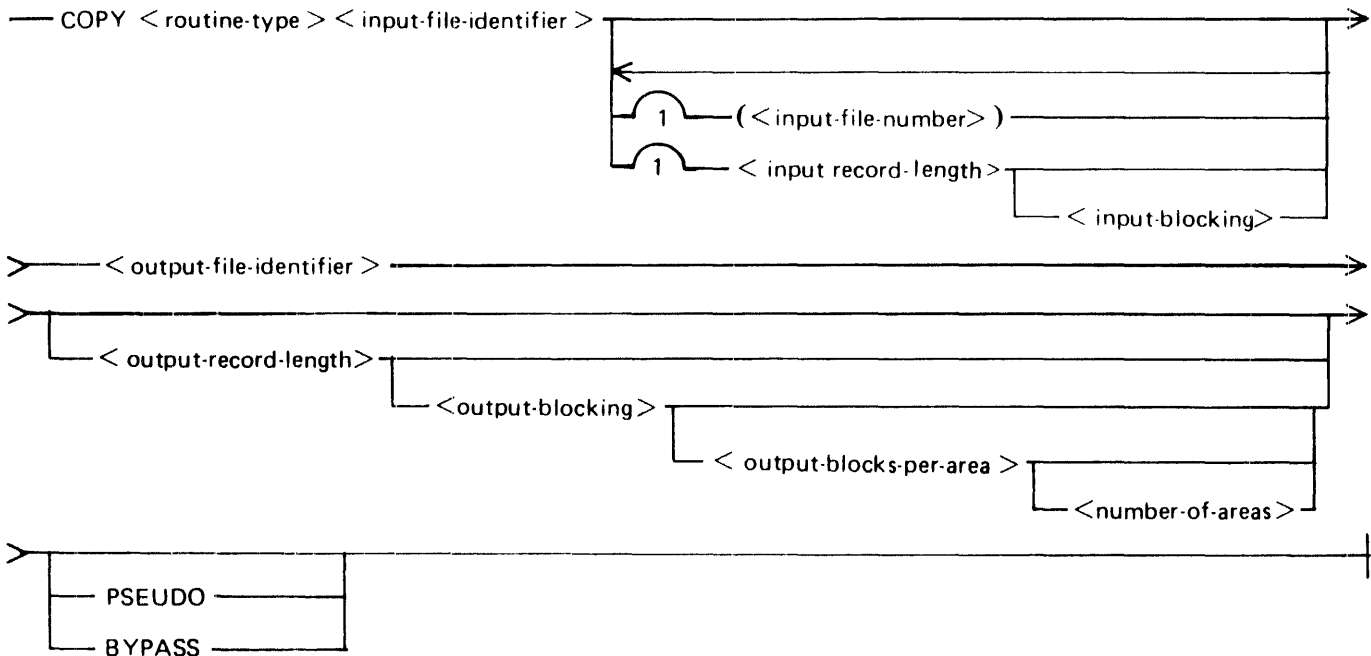
AUTOLOAD Facility

The DISKETTE/COPY program automatically loads all PSR pseudo-reader files contained on a diskette to the system disk when the diskette is placed on-line. PSR pseudo-reader files have the letters PSR as the first three characters of their eight-character file-id. When the diskette is readied, the MCP II searches the diskette directory for PSR file-ids. If one or more PSR files are found, then DISKETTE/COPY is executed by the MCP, which then automatically copies the PSR files to the system disk as pseudo-reader decks, without operator intervention.

COPY Command

A description of the COPY command syntax follows.

Syntax:



Routine Type Specifier

The <routine-type> is specified in the same manner as the shorthand notation for <routine-type> in the DMPALL utility program and must designate a diskette unit as one (and only one) of the two units. Abbreviations for the allowable devices follow:

Hardware Device	Abbreviation
Card	CRD
96-col. card	C96
Magnetic tape	MTP
Paper tape	PPT
Disk	DSK
Diskette	FDx (x indicates the drive)

For example, to copy from cards to diskette unit 0 (FDA), the <routine-type> is CRDFDA.

Input File Specifications

The format of <input-file-id> is the same as that used in MCPPII control instructions. A diskette <file-identifier> can be up to eight characters in length and must be specified as either a single-name identifier or a two-name identifier with the following format:

<volume-id>/<file-id>

The <volume-id> can be up to six characters in length, and must agree with the volume-id of the designated diskette. If the <volume-id> is omitted, IBMIRD or a blank <volume-id> is assumed, depending on the actual volume-id of the designated diskette. If <volume-id> is omitted and the volume-id on the diskette is neither blank nor IBMIRD, then an error message is displayed. A diskette with a volume-id that matches the volume-id specified in the COPY command must be loaded, or processing of the specification must be terminated. If a null string is entered through an AC or AX command to the DISKETTE/COPY program, processing of the current specification stop and the following prompt is issued:

PLEASE ENTER COPY PARAMETERS

The <input-file-number> is an integer (enclosed in parentheses) used when there is more than one input file on the diskette with the same name. In such a case, the <input-file-number> specifies which of these files to use as input (the first of such duplicate file names is considered number one). Specifying the <input-file-number> causes DISKETTE/COPY to ignore the BYPASS flags.

The <input-record-length> specifies the input file record size in bytes. If omitted, a record length of 80 is assumed except for disk files, where the actual record length of the file is used. The maximum record size permitted on a diskette input file is 128 bytes.

The <input-blocking-factor> is the second numeric entry, and specifies the number of logical records in a block (physical record). If omitted, a blocking factor of one is assumed, except for disk files, where the actual blocking factor is used. The maximum blocking factor permitted on a diskette input file is one.

Output File Specifications

The format of the <output-file-identifier> is the same as that of the <input-file-identifier>.

The <output-record-length> specifies the output file record size in bytes. If omitted, a record length of 80 bytes is assumed, unless either the input or output file is a disk file (FDxDSK or DSKFDx). When a disk file is used, the default <output-record-length> is the same as the <input-record-length>. The maximum record size permitted on a diskette output file is 128 bytes, and the input records being written to the diskette file are truncated if they are longer than 128 bytes.

The <output-blocking-factor> is the second numeric entry following <output-file-identifier>. If omitted, a blocking factor of one is assumed. The maximum blocking factor permitted on a diskette output file is one.

The <output-blocks-per-area> value is the third numeric entry following the <output-file-id>. This entry is applicable only to non-diskette disk files. If omitted, 100 blocks per area are assumed by default.

The <output-areas> value specifies the maximum number of disk areas allowed. This entry is applicable only to non-diskette disk files. If omitted, 25 areas are assumed by default.

BYPASS Option

The BYPASS option creates files on the diskette with duplicate identifiers and sets the BYPASS flag for the file. If there are duplicate diskette file identifiers, only one of these files can be NOT BYPASS.

PSEUDO Option

The PSEUDO option can be used following the <output-file-identifier> and in place of all other output file specifications. When used, the input file from the diskette is copied to system disk as a pseudo-reader file. The <output-file-id> is required for the syntax, but is ignored when the pseudo-reader file is created. The file is assigned a disk-file-id of DECK/<#>. Truncation or padding of the input records to 96 bytes is performed where necessary.

KA Command

The DISKETTE/COPY program can provide a line printer listing of the file-ids and file characteristics of all files on a designated diskette. A description of the syntax of the KA command follows.

Syntax:

— KA <unit-mnemonic> —————|

The KA listing includes the diskette addresses of each file's extent and lists the following information about the diskette:

DATE/TIME

The DATE/TIME fields represent the current system date/time.

V.IDENTIFIER

The V.IDENTIFIER field represents the volume identifier.

ACCESSIBILITY

The ACCESSIBILITY field must be blank if the DISKETTE/COPY program is to access the diskette. A non-blank field means that the diskette is inaccessible.

RESERVED

The RESERVED field is to be blank (non-reserved), and is ignored by the DISKETTE/COPY program.

STANDARD.LABEL

The STANDARD.LABEL field must be a W. This indicates a standard label format. If this field contains any other value, the DISKETTE/COPY program cannot read the diskette.

The KA command also lists the following information about each file (referred to as a "data set") on the diskette:

FILE.ID

The FILE.ID field is the diskette file identifier.

BLOCK.SIZE

The BLOCK.SIZE field is the record size of the file.

H.BOE

The H.BOE field is the beginning-of-extent. This is the address of the first sector of the file.

H.EOE

The H.EOE field is the end-of-extent. This is the address of the last sector reserved for the file and indicates the end of the disk area allocated to the file.

H.EOD

The H.EOD field is the end-of-data. This indicates where the data ends, yet actually points to the next available sector (EOF + 1). For a file that has filled its extent, this is equal to H.EOE + 1. If H.EOD is equal to H.BOE, then the file is actually an available area define. On a diskette that contains less than 19 files and has available space, there is always an extra file present which describes the available space. This dummy file-id is DATA if it is the only file present (scratch disk), or DATA <nn> where <nn> is the sector number of the track 0 sector in which the file header is located.

WRITE.PROTECT

A letter P in this field protects the file from being overwritten. The DISKETTE/COPY program creates only non-protected output files, and the WRITE.PROTECT field contains a blank.

INTERCHANGE

The INTERCHANGE field must be blank, indicating that the file can be used for data interchange.

MULTI.VOL

A blank character in the MULTI.VOL field indicates that the file is fully contained on the diskette; a letter C indicates the file continues on another diskette; a letter L indicates that this is the last diskette on which a continued file resides.

MV.SEQ#

The MV.SEQ# field contains an optional multi-volume sequence number that specifies the sequence of diskettes in a multi-volume file. If used, sequence numbers must be consecutive starting at 01 (to a maximum of 99).

CREATED

The CREATED field is the date the file was created, in YYMMDD format.

EXPIRES

The EXPIRES field is the date the file expires, in YYMMDD format. The DISKETTE/COPY program assigns a value of the creation date plus seven days. Files are never automatically purged, regardless of the value in the EXPIRES field.

BYPASS

A blank character in the BYPASS field indicates that the file is accessible to the DISKETTE/COPY program without using the <input-file-number>; The letter B in this field specifies that the <input-file-number> must be used to access the file.

PURGE Command

The DISKETTE/COPY program can purge all files contained on a diskette, converting it to a scratch diskette.

Syntax:

—PURGE <unit-mnemonic> —————

Only the file that describes the available space on the diskette remains. Its name is DATA, its extent is 01001 to 73026, and its end-of-data pointer is 01001.

RELABEL Command

The DISKETTE/COPY program can change the <volume-id> of a diskette by using the RELABEL command.

Syntax:

—RL FDx <volume-id> —————

The <volume-id> can have a maximum length of six characters.

The data files contained on the diskette are not affected by the RELABEL command.

SAMPLE INPUT SPECIFICATIONS

Sample input specifications with a description of the action performed for each are described in the following paragraphs.

COPY CRDFDA CARDS <volume-id>/CRDIMG

The card file named CARDS is copied to the diskette on drive A. The <volume-id> of the diskette must match the specified <volume-id>.

COPY FDADSK INPUT (2) 80 1 USER/DISKETTE/INPUT 80 9 100 25

The second file with the file-id INPUT on the diskette is copied to the disk USER and is named DISKETTE/INPUT. The diskette file record size is 80. When the file is copied to disk, it is blocked 9 records to a block, and the blocks-per-area and number-of-areas for the output file are specified.

COPY MTPFDB TAPEFILE 180 10 FDFILE 128 1 BYPASS

The BYPASS keyword allows the file-id FDFILE to be assigned, even when other files with that name exist on the diskette.

COPY FDBDSK PSRDECK DUMMY PSEUDO

The diskette file named PSRDECK is copied to disk as a pseudo deck. A dummy output-file-id is required, even though the system assigns a pseudo-reader file-id to the file. Because the diskette file-id begins with PSR, the file was also automatically copied as a pseudo-deck when the diskette was readied.

KA VOLID1

The diskette with the volume-id VOLID1 is analyzed, and a KA listing is generated.

PURGE FDA

Creates a scratch diskette on drive A.

RL FDB VOLID2

Changes the label of the diskette to VOLID2. Data on the diskette is not affected.

ERROR MESSAGES

NO INPUT FILE SPECIFIED

The <input-file-identifier> is missing from the COPY specifications.

NO OUTPUT FILE SPECIFIED

The <output-file-identifier> is missing from the COPY specifications.

INVALID COPY FILE HARDWARE TYPE <hardware-type>

The <hardware-type> is not a valid device abbreviation.

FLEXIDISK MUST BE SPECIFIED AS ONE OF THE COPY MEDIA

INVALID FILE NAME

INVALID PARAMETER <parameter>

MISSING COPY STATEMENT

The input request was not recognized.

MISSING FILE SPECIFICATION

No file identifiers were entered in the COPY statement.

MEMORY PARITY ERROR ON DISKETTE I/O

System had a memory parity error while doing the diskette I/O.

UNEXPECTED NOT READY CONDITION ON <diskette-unit-id>
The diskette was not ready.

<error = type> ERROR ON <unit> TRACK = <track> SECTOR = <sector>
SECTOR BYPASSED.

An error (designated by <error-type>) was encountered on the diskette at the specified sector. If input, the copy continues, but the data in the sector is lost. If output, the sector is marked as deleted (if possible).

TIME OUT ON <unit mnemonic>
The diskette unit had a timeout error. The copy is terminated.

TRACK SEEK ERROR ON <unit-mnemonic>
The diskette had a track seek error. The copy is terminated.

UNCORRECTABLE WRITE FAULT ON <unit-mnemonic>
The diskette had a write fault error. The copy is terminated.

UNDIAGNOSED ERROR ON <unit-mnemonic>
The diskette had an unknown error. The copy is terminated.

YOUR SUPPLIED RECORD SIZE DOES NOT MATCH LABEL RECORD SIZE
The record size on the diskette is not the same as specified.

DISKETTE BLOCKING FACTOR MUST BE 1

INVALID BEGINNING OF EXTENT ON <unit> <h.boe>
The beginning of extent is not valid.

INVALID END OF DATA ON <unit> <h.eod>
The end-of-data is not valid.

BEGINNING OF EXTENT <h.boe> EXCEEDS END OF DATA <h.eod> ON <unit>

END OF DATA <h.eod> EXCEEDS END OF EXTENT <h.eoe> ON <unit>

INVALID HARDWARE TYPE FOR PSEUDO DECK
PSEUDO TYPE PERMITTED ON OUTPUT DISK FILE ONLY
The output <hardware-type> must be disk (DSK) if specifying PSEUDO.

MAXIMUM DISKETTE RECORD SIZE EXCEEDED
The maximum record size for diskette files is 128 characters.

WARNING: DISKETTE DEFAULTS ON OUTPUT ARE RECORD SIZE OF
128 CHARACTERS BLOCKED 1
This is a warning message only. The record size of the input file is greater than 128 bytes. Records are truncated to 128 bytes when copied to the diskette.

WARNING: INVALID HEADER ON <unit-mnemonic>
This is a warning message only. One of the files on the diskette has a bad header.

UNABLE TO READ VOLUME LABEL ON <unit-mnemonic>

UNABLE TO READ HEADER IN SECTOR <sector> OF <unit-mnemonic>
The file whose header is in the specified sector is inaccessible.

UNABLE TO WRITE ENDING HEADER IN SECTOR <sector> of <unit>
The header could not be written, so the copy is terminated.

DUPLICATE FILE FOUND ON UNIT <unit-mnemonic> -- NO COPY PERFORMED
The diskette had duplicate file identifiers with the BYPASS flag not set on the duplicate files.

NO FILE <file-id> FOUND ON FLEXIDISK <volume-id>
The file requested is not present on the diskette.

OVERLAPPING HEADER (BOE/EOE BOUNDARIES) ON FLEXIDISK LABELED <volume-id>

VOLUME ACCESSIBILITY ON <unit-mnemonic> IS PROHIBITED
The ACCESSIBILITY flag in the volume header is not blank.

VOLUME LABEL ON <unit-mnemonic> IS NOT STANDARD
The STANDARD.LABEL field does not contain a W.

VOLUME SEQUENCE NUMBER MISMATCH, FOUND <sequence> LOOKING FOR <sequence>
For multi-volume diskette files, the MV.SEQ# must be sequential.

NO SPACE AVAILABLE ON <unit-mnemonic> FOR FILE LABELED <file-id>
There is insufficient space on the diskette for the file.

NO DIRECTORY SPACE AVAILABLE FOR FILE <file-id> ON UNIT <unit>
The diskette already has 19 (the maximum allowed) files.

INVALID FD SPECIFICATION <parameter> ON <type> REQUEST
The request identified by <type> contains an invalid FD specification.

MISSING FD SPECIFICATION ON <type> REQUEST
The request identified by <type> does not contain an FD specification.

UNABLE TO WRITE VOLUME LABEL ON <unit-mnemonic>
The RELABEL procedure cannot write the new volume label.

MISSING RIGHT PAREN IN DUPLICATE FILE SPECIFICATION
The <input-file-number> must be enclosed in parentheses.

DUPLICATE FILE NUMBER REQUESTED NOT FOUND ON UNIT <unit mnemonic>
The file requested is not present on the diskette.

DISKETTE <volume-id> FOR FILE LABELED <file-id> NOT FOUND. LOAD AND <MIX>
AX <UNIT MNEMONIC>
The requested diskette cannot be found. Ready the diskette and enter <mix-number> AX <unit-mnemonic> to attempt to resume the copy. Enter a null AC or AX command to cause the DISKETTE/COPY program to discontinue from attempting to process the current input specification. Enter a new COPY PARAMETER to request a different diskette.

SECTION 15

DISKPACK/INTERCHANG

One Burroughs standard structure of interchange media, as defined for disk packs, is compatible across several of the Burroughs product lines.

The DISKPACK/INTERCHANG program copies and translates data to or from an Interchange (I) pack. Interchange packs allow data to be exchanged between Burroughs systems.

The DISKPACK/INTERCHANG program has two basic modes of operation:

1. The forward mode, which creates a disk pack in standard Interchange format from a B 1000 user disk pack.
2. The reverse mode, which creates a B 1000 user disk pack.

OPERATING INSTRUCTIONS

If the forward mode is desired, mount the input B 1000 user disk pack in one disk pack drive, and mount a disk pack initialized as INTERCHANGE in another disk pack drive.

If the reverse mode is desired, mount the previously written Interchange disk pack in one disk pack drive, and mount a newly initialized or purged B 1000 user disk in another disk pack drive.

Execute the DISKPACK/INTERCHANG program in the appropriate manner. (Refer to the SYSTEM/DISK-DUMP program for execution instructions.) Messages from the DISKPACK/INTERCHANG program are displayed on the operator display terminal (ODT). Responses are also entered through the ODT (with AX or AC messages for DISKPACK/INTERCHANG). The messages displayed and the responses expected are as follows:

WHICH PACK (SOURCE OR DESTINATION) IS B1700 FORMAT?

An S response causes the forward mode to be selected, which creates an Interchange disk pack.

A D response causes the reverse mode to be selected, which creates a B 1000 user disk.

ENTER SOURCE PACK DRIVE (A,B,C,ETC.)

Enter the unit-mnemonic of the input disk drive.

ENTER DESTINATION PACK DRIVE (A,B,C,ETC.)

Enter the unit-mnemonic of the output disk drive.

Neither drive specified can be a system drive, nor can the source and destination unit-mnemonics specify the same drive.

ERROR MESSAGES

Due to the differences between the B 1000 user disk format and the standard Interchange format, it is not possible to copy certain files. As these incompatibilities are encountered during conversion, they are indicated by one of the following error messages. The program continues to copy the remaining valid files.

FILE <file-id> CONTAINS MULTI PACK LINKS, NOT COPIED
Files containing multi-pack areas are not copied.

FILE <file-id> CONTAINS > 105 AREAS, NOT COPIED
Files with more than 105 disk areas are not copied (reverse mode only).

FILE <file-id> EXCEEDS 6 CHARACTERS, NOT COPIED
Files with file-ids longer than six characters are not copied (forward mode only).

FILE <file-id> IS MULTI-FILE NAME, NOT COPIED
Files with family identifiers are not copied (forward mode only).

FILE <file-id> IS VARIABLE LENGTH, NOT COPIED
Files with variable length records are not copied (forward mode only).

FILE <file-id> WAS NOT CLOSED WHEN LAST USED, NOT COPIED.
Files with a user count greater than zero are not copied (forward mode only).

The following messages can appear due to disk I/O or format errors:

DISK ERROR AT <sector-address> ON FILE <file-name>
An irrecoverable I/O error occurred on the input or output disk pack. The file containing the error is not copied.

TOO MANY ERRORS AT SECTOR <sector-address>, SECTOR REMOVED
An irrecoverable I/O error occurred on the output pack. The sector in error is removed from the Master Available Table on the output pack, and processing continues.

DISK ERROR AT <sector-address> ON <input-or-output> DRIVE
An irrecoverable I/O error occurred on the specified pack. The DISKPACK/INTERCHANG program terminates immediately.

COULD NOT FIND SOURCE PACK
Indicates that the specified pack is (1) not on the proper drive, (2) not ready, or (3) in-use.

COULD NOT FIND DESTINATION PACK
Indicates that the specified pack is (1) not on the proper drive, (2) not ready, or (3) in-use.

DESTINATION PACK NOT AN INITIALIZED PACK
The destination pack is not a scratch pack (reverse mode only) or has not been properly initialized.

ERROR IN PACK LABEL, CAN'T CONTINUE
Indicates that one of the following has occurred:

The source and destination packs are reversed. (The copy attempt is in the wrong direction.)

The source pack is not a B 1000 user disk or a disk written in the standard Interchange format.

NOT ENOUGH DISK SPACE
Enough space could not be found on the output pack for the current file or structure. Either the output pack is smaller than the input pack, the output pack has too many removed (XDed) sectors, or it has a bad available table. A larger pack must be used or the output pack must be re-initialized.

SECTION 16

DMPALL

The DMPALL program is a generalized media-conversion utility program that operates under MCPPII control. The DMPALL program has four distinct functions.

1. Printing/punching the directory of Load Dump tapes.
2. Printing the contents of files.
3. Reproducing data from one hardware device to another.
4. Concatenating files to produce a single disk file.

OPERATING INSTRUCTIONS

The DMPALL program is executed from the ODT or from a card reader. If the ODT is used, input specifications are entered by means of MCP ACCEPT mechanism (AC or AX commands). If program switch 0 or 3 = 1, input specifications are accepted from a card file labeled SPEC. Each input specification is processed as it is received; after the requested action is completed, the next input specification is read and processed.

Program Switches

Program switches are used to specify the manner in which the DMPALL program executes. The switches used and their functions are described in table 16-1.

Table 16-1. DMPALL Program Switches

Switch	Value	Meaning
0 or 3	0	Specifications accepted only from the ODT.
	1	Specifications accepted only from cards.
1	0	Continue processing to the end.
	1	Terminate processing the current specification and proceed to the next specification.
2	0	Number of syntax errors encountered are not displayed at any time.
	1	Number of syntax errors are displayed at EOJ.
4	0	ISAM data files are listed as relative files and causes no block control bytes to be listed.
	1	ISAM data files are listed as sequential files and causes block control bytes to be listed.
7	0	Continue with next specification.
	1	Terminate DMPALL after completion of the next executed specification.
8	0	Interpret card decks.
	1	Do not interpret cards.
6 or 9	0	Terminate DMPALL if an error is encountered in input files.
	1	Ignore input file errors (for example, parity errors).

When the DMPALL program detects that switch 1 is set, it terminates the specification being processed and then resets switch 1 by zipping a <mix=number> SWITCH 1=0 message; thus, processing of only the current specification is terminated.

Input to the DMPALL Program

The DMPALL program, by default, issues prompts asking for input specifications. The responses are entered through the ODT.

Specifications can, as an option, be entered from a file labeled SPEC if the DMPALL program is executed with switch 0 or 3 set to 1. By default, this file is defined as a card file. The DMPALL card control deck has the following format:

```
? EXECUTE DMPALL SWITCH 0=1
? DATA SPEC
  <specification cards>
? END
```

Changing the external file identifier of the file SPEC with a FILE statement, as shown below, has the same effect as setting switch 0 or 3.

```
? EXECUTE DMPALL
? FILE SPEC NAME= SPECS;
? DATA SPECS
  <specification cards>
? END
```

There can be more than one record in the specification file; however, a single specification string can extend beyond one record by specifying a hyphen with a leading and trailing blank character (-). The file containing the specifications is loaded to disk by the DMPALL program (to free the reader for other programs); each specification is then processed from disk.

All specifications are entered in free-form format and can be up to 96 characters in length. Options and keywords within any one specification are separated by either a space or a comma, or any combination thereof. A semicolon terminates any specification string, after which comments can be entered.

Sample Input Specification Strings

These strings are presented in advance of the command descriptions to give an idea of how the commands are used. The first string is used to list a disk file on pack USERA in alphanumeric format, beginning with the fifty-first record:

```
LIST USERA/PAYROLL/ A SKIP 50
```

The following specification string is used to start printing a tape file with the first record that has the letters C U B E beginning in byte 37 of the record (a byte contains one character):

```
LIST ABC MTP SEARCH 37 "CUBE"
```

The following specification string is used to print in hexadecimal format only the records 11 through 20 that have a hexadecimal @FF@ beginning in byte 1.

```
LIST ABC H SKIP 10 INCL 10 SELECT 1 @FF@
```

To print (double-spaced) only the records whose second byte is greater than hexadecimal @C0@, any of the following three specifications can be used (note the different <argument> types and their effect on the <start> and <length> specifications):

```
LIST2 ABC KEY 2 1 GTR @C0@
LIST2 ABC KEY 3 2 GTR @(4)C0@
LST2 ABC KEY 9 8 > @(1)11000000@
```

The following specification string is used to copy a card file labeled CARDS to a disk file labeled SOURCE having 80-byte records blocked 9:

```
COPY CRDDSK CARDS SOURCE 80 9
```

The following specification string is used to copy an ASCII tape file to an EBCDIC disk file:

```
PFM TAPE TO DISK TAPEFILE "ASCII" DISKFILE
```

The following specification string is used to copy a disk file, exclude all records that have CUBE beginning in byte 37, and change the blocking factor:

```
PERFORM DSKDSK ABC 80 1 DEF 80 9 EXC 37 "CUBE"
```

Sample Executions

To copy a card file labeled XXX to a disk file labeled DSKFILE, and then to list the resulting disk file, the following execution deck can be used:

```
?EXECUTE DMPALL SWITCH 0=1
?DATA SPEC
COPY CRDDSK XXX 80 1 DSKFILE 80 1
LIST DSKFILE A
?DATA XXX
  <data cards>
?END
```

Tape Parity Modifications

To change the parity for an input or output tape file from ODD to EVEN, use one of the following FILE statements with the EXECUTE, MODIFY, or DYNAMIC system commands:

```
Input files: FILE INPFILE EVEN;
Output files: FILE OUTPFILE EVEN;
```

Examples:

```
EXECUTE DMPALL;FILE INPFILE EVEN;
MODIFY DMPALL;FILE OUTPFILE EVEN;
DYNAMIC 1223 FILE INPFILE EVEN;
```

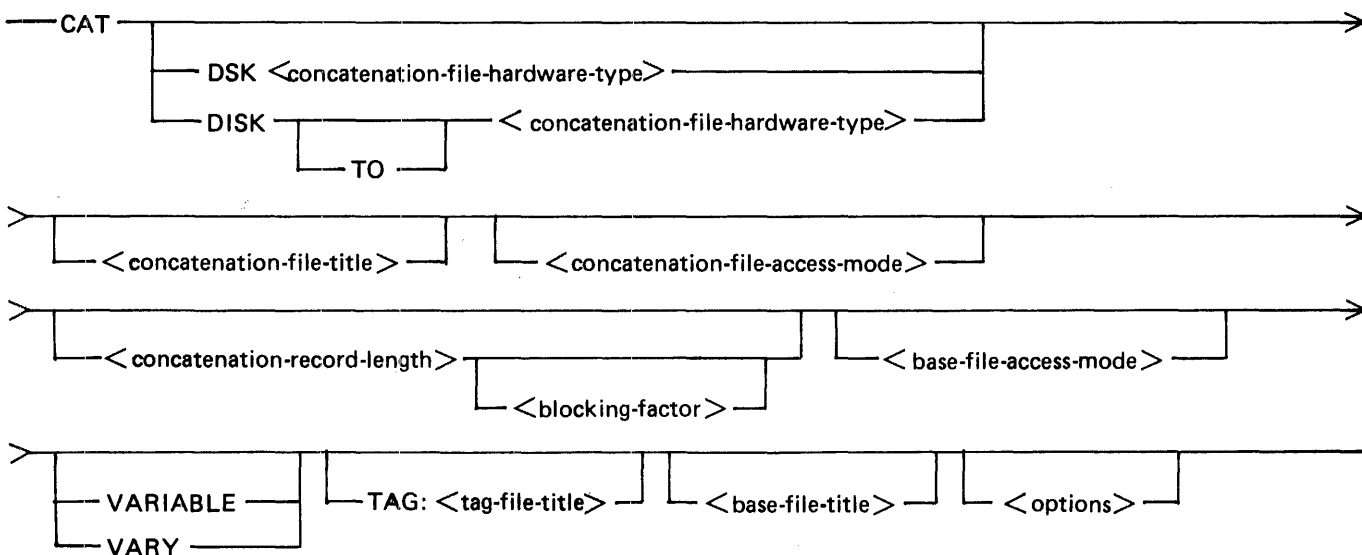
COMMANDS

The DMPALL program commands are CAT (concatenation), COPY (or PERFORM), LIST, and PD (print directory). These commands and their functions are described in detail in the following pages. The various specifications referred to in the command semantics are all grouped under Specifications, following the PD command.

CAT (Concatenation) Command

The CAT command concatenates files. Records from any card, magnetic tape, paper tape, or disk file can be copied to the end of a specified disk file, called the base file. The base file is opened, and the specified concatenation file records are added to the base file; when the base file is closed, the end-of-file (EOF) pointer is updated to include the new records. If the concatenated file size exceeds the size limit of the base file, and if both the concatenation and the base files are disk files, then the DMPALL program copies the base file, enlarging it, and then adds the concatenation file. Likewise, if the base file is crunched, the DMPALL program copies the base file before performing the concatenation. If both files are not disk files, the operator must ensure that the concatenated file does not exceed the file size limit of the base file.

Syntax:



Semantics:

base-file-title

This specifies the name of the file (base file) to append the <concatenation-file-title> file. Refer to <file-title> Specification for the syntax and semantics of <base-file-title>.

blocking-factor

This field specifies the number of records per block for the file and must be numeric. If <blocking-factor> is not specified, it defaults to 1. For disk and labeled B 1000/B 6000/B 7000 tape files, when both <concatenation-record-length> and <blocking> are not specified, the value of <blocking-factor> defaults to the value contained in the label record.

concatenation-file-access-mode

This field specifies the recording mode of the concatenation file. Refer to <access-mode> Specification for the syntax and semantics.

concatenation-file-hardware-type

This field specifies the hardware type of the concatenation file. Refer to <hardware-type> Specification for the syntax and semantics.

concatenation-record-length

This field specifies the size of each record, in bytes, and must be numeric. If <concatenation-record-length> is not specified, it defaults to 80 bytes, except for disk and labeled B 1000/B 6000/B 7000 tape files, where <concatenation-record-length> defaults to the record size of the file in the label record.

concatenation-file-title

This field specifies the name of file in which to append to the end of the <base-file-title> file. Refer to <file-title> Specification for the syntax and semantics of <concatenation-file-title>.

DISK

The DISK keyword specifies that the hardware type of the input file is a disk file.

DSK

The DSK key symbol specifies that the hardware type of the input input is a disk file. If DSK is specified, then there can be no blank character between the DSK key symbol and the <concatenation-file -hardware-type> specification.

options

Refer to <options> Specification for the syntax and semantics.

TAG:

The TAG: key symbol specifies that the file to be listed is the tag file of <tag-file-title>. This applies to the B 1000 ANSI 68 COBOL RPGII index (\$ TAG) files. The tag file must be created with the B 1000 ANSI 68 COBOL or RPGII naming conventions. The tag file specified is read sequentially, and the key field of each tag record is used to access the associated indexed file record.

tag-file-title

Refer to <file-title> Specification for the syntax and semantics.

TO

The TO keyword is optional and specifies that the <concatenation-file hardware-type> specification follows.

VARIABLE

The VARIABLE keyword specifies that the input file is a variable-length record file.

VARY

The VARY keyword specifies that the input file is a variable-length record file.

Pragmatics:

When concatenating files, the base file must be a disk file. The concatenation file can have any input hardware device; LIST (printer) is not a valid input device.

The <concatenation -record-length> and <blocking-factor> options are specified in the same manner as the <record-size> and <blocking> options in the LIST command. If the record length of the concatenation file is larger than the record length of the base file, then the concatenation file records are truncated when they are added to the base file. The <base-file-title> file must name an existing disk file.

A record size and blocking factor cannot be specified for the base file. These values are taken from the disk file header of the base file.

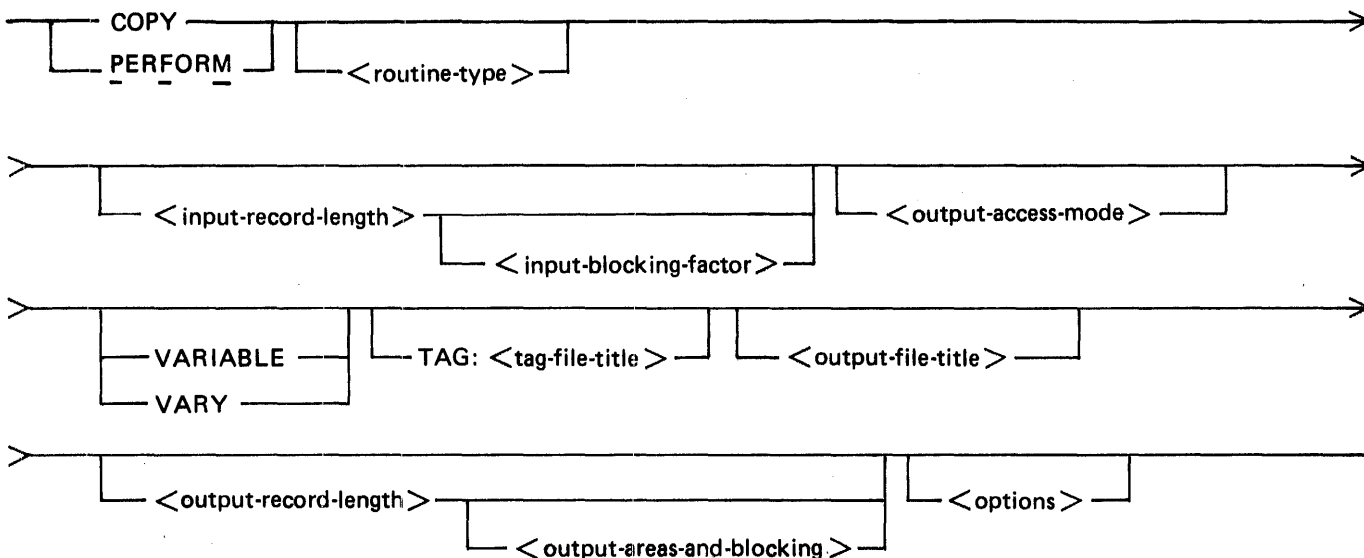
Examples:

```
CAT DSKDSK HISTADDS 180 1 HISTORY
CAT DISK TO TAPE APPENDIX 90 20 MANUAL SKIP 4
```

COPY (or PERFORM) Command

Records can be copied from any card, magnetic tape, paper tape, or disk file to any output file desired. Various file attributes, including record sizes, blocking factors, and file-identifiers can be changed during the copy process. It is also possible to select certain records from the input file to be copied based on specified conditions.

Syntax:



Semantics:

"NONE"

The "NONE" keyword specifies that the file to list is an unlabeled tape file. The double quotation mark (") characters are required.

input-access-mode

This field specifies the recording mode of the input file. Refer to <access-mode> Specification for the syntax and semantics.

input-blocking-factor

This field specifies the number of records per block for the file and must be numeric. If <input-blocking-factor> is not specified, it defaults to 1. For disk and labeled B 1000/B 6000/B 7000 tape files, when both <input-record-length> and <input-blocking-factor> are not specified, the value of <blocking-factor> defaults to the value contained in the label record.

input-file-title

This field specifies the name of the input file to copy. Refer to <file-title> Specification for the syntax and semantics.

input-record-length

This field specifies the size of each record, in bytes, and must be numeric. If <input-record-length> is not specified it defaults to 80 bytes, except for disk and labeled B 1000/B 6000/B 7000 tape files, where <input-record-length> defaults to the record size of the input file in the label record.

options

This field specifies the DMPALL options to skip or include records, to exclude, search, or select records based on specified conditions, or specify that the input file is a variable-length record file. Refer to the <options> Specification for the syntax and semantics.

output-access-mode

This field specifies the recording mode of the output file. Refer to <access-mode> Specification for the syntax and semantics.

output-areas-and-blocking

This field specifies the blocking factor, blocks per area, and number of areas for the output file. Refer <output-areas-and-blocking> Specification for the syntax and semantics.

output-file-title

This field specifies the name of the output file. Refer to <file-title> Specification for the syntax and semantics.

output-record-length

This field specifies the length in bytes for the output file. If omitted, the record length defaults to 80 bytes, unless both the input and output files are either disk files or labeled B 1000/B 6000/B 7000 tape files. In this case, <output-record-length> is assumed to be the same as the <input-record-length>.

routine-type

This field specifies the hardware type of the input file and output files. Refer to <routine-type> Specification for the syntax and semantics.

TAG:

The TAG: key symbol specifies that the file to be created is a tag file with the name <tag-file-title>. This applies to the B 1000 ANSI 68 COBOL RPGII index (\$ TAG) files. The tag file must be created with the B 1000 ANSI 68 COBOL or RPGII naming conventions. The tag file specified is read sequentially, and the key field of each tag record is used to access the associated indexed file record.

tag-file-title

This field specifies the name of the tag file to be created. Refer to <file-title> Specification for the syntax and semantics.

VARIABLE

The VARIABLE keyword specifies that the input tape or disk file is a variable-length record file. If the input file is a disk file, the DMPALL program checks the disk file header to verify that the file consists of variable-length records; if not, the VARIABLE option is ignored.

VARY

The VARY keyword specifies that the input tape or disk file is a variable-length record file. If the input file is a disk file, the DMPALL program checks the disk file header to verify that the file consists of variable-length records; if not, the VARY option is ignored.

Pragmatics:

When a partial or selective copy (SKIP, INCLUDE, SELECT, or SEARCH options) is specified for an input file with a file type of PRINTER BACKUP, PUNCH BACKUP, PSEUDO-DECK, RELATIVE, or INDEX-SEQUENTIAL (ISAM), the resulting output is a file of type DATA. These file types have special meaning to the operating system and cannot be partially or selectively copied.

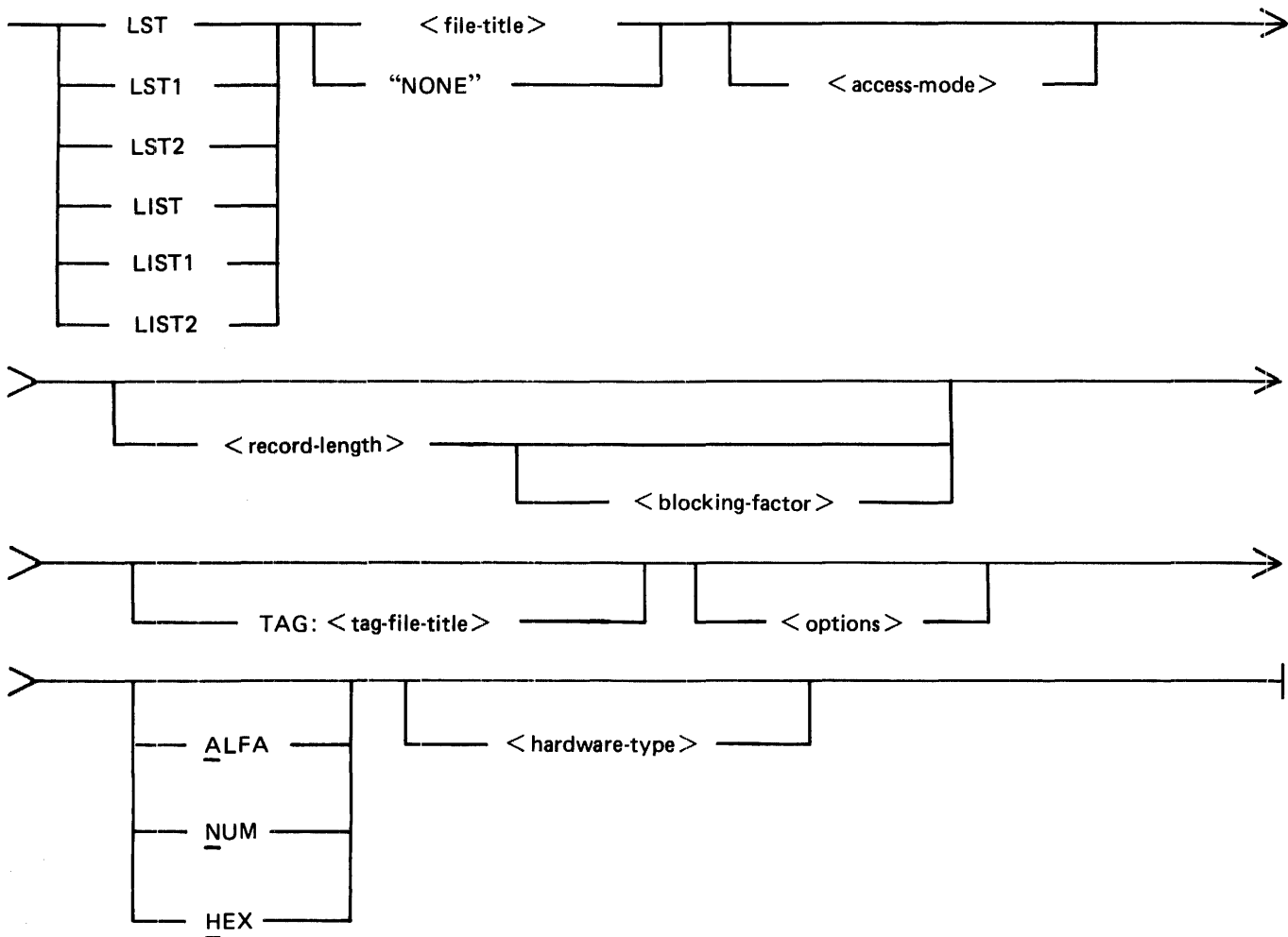
LIST Command

A line printer listing of files in hexadecimal, alphanumeric, or combined form can be generated from card, magnetic tape, cassette, paper tape, or disk files. Options allow for selective inclusion or exclusion of records in the listing, based on operator-specified conditions.

The output listing contains the file identifier, record size, blocking factor, and the current date and time. In addition, a listing of a disk file contains the end-of-file pointer value in the heading. For each record printed, the current record count is printed in the left-hand margin.

The page format may be controlled with the file attributes LINEFORMAT, PAGESIZE, and LOWERMARGIN = 1.

Syntax:



Semantics:

LIST, LIST1, LST, LST1

The keysymbols LIST, LIST1, LST and LST1 print the file, single-spaced.

LIST2, LST2

The keysymbols LIST2 and LST2 print the file, double-spaced.

"NONE"

The "NONE" keyword specifies that the file to be listed is an unlabeled tape file. The double quotation mark (") characters are required.

file-title

This field specifies the name of the file to be listed. Refer to <file-title> Specifications for syntax and semantics.

record-length

This field specifies the size of each record, in bytes, and must be numeric. If <record-length> is not specified, it defaults to 80 bytes, except for disk and labeled B 1000/B 6000/B 7000 tape files, where <record-length> defaults to the record size of the file in the label record.

blocking-factor

This field specifies the number of records per block for the file and must be numeric. If <blocking-factor> is not specified, it defaults to 1. For disk and labeled B 1000/B 6000/B 7000 tape files, when both <record-length> and <blocking-factor> are not specified, the value of <blocking-factor> defaults to the value contained in the label record.

TAG:

The TAG: keysymbol specifies that the file to be listed is the tag file of <file-title>. This applies to the B 1000 ANSI 68 COBOL RPGII index (\$ TAG) files. The tag file must be created with the B 1000 ANSI 68 COBOL or RPGII naming conventions. The tag file specified is read sequentially, and the key field of each tag record is used to access the associated indexed file record.

ALFA, HEX, or NUM

The keysymbols ALFA, HEX, and NUM specify the output format type. ALFA causes the output to be alphanumeric. HEX and NUM cause the output to be hexadecimal. If no output format type is specified, the file is listed in both alphanumeric and hexadecimal format.

hardware-type

If <hardware-type> is not specified, the default is to DISK. Refer to <hardware-type> Specification for the syntax and semantics.

options

Refer to <options> Specification for a description of the list options available with the DMPALL program.

PD (Print Directory) Command

The PD command lists or punches the directory of a library tape created by the SYSTEM/LOAD-DUMP program. Use the DIR command to list the directory of a SYSTEM/COPY generated tape.

Syntax:



Semantics:

PUNCH

The PUNCH keyword causes the tape directory to be punched, one directory entry per card. If the PUNCH keyword is omitted, the tape directory is listed on the line printer.

tape-identifier

This field specifies the tape name of the SYSTEM/LOAD-DUMP generated tape.

Examples:

```
PD SYSTEM
PD PUNCH BACKUPTAPE
```

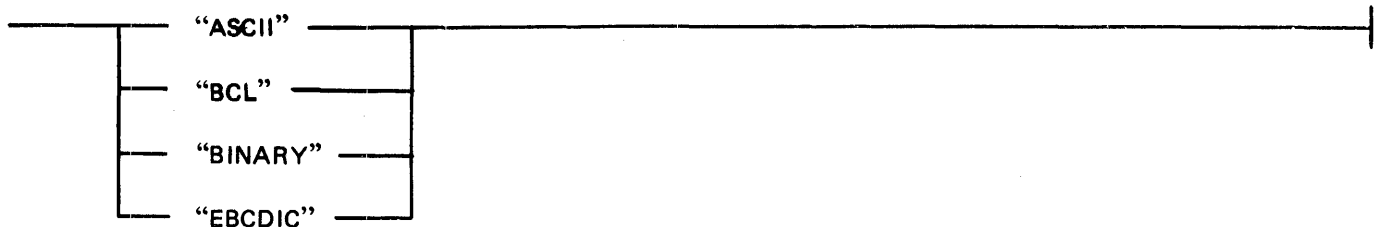
Specifications

Following are descriptions of all the specifications referenced in the discussions of the commands.

< access-mode > Specification

The < access-mode > Specification identifies the recording mode of the file. If the < access-mode > Specification is omitted, "EBCDIC" is assumed as the default. If the input file is a labeled tape file, the < access-mode > Specification defaults to the access mode that is included in the label record.

Syntax:



Semantics:

"ASCII"

The "ASCII" keyword specifies that the file is in the ASCII recording mode.

"BCL"

The "BCL" keyword specifies that the file is in the BCL recording mode.

"BINARY"

The "BINARY" keyword specifies that the file is in the BINARY recording mode.

"EBCDIC"

The "EBCDIC" keyword specifies that the file is in the EBCDIC (default) recording mode. The recording mode defaults to the recording mode contained in the label record of the input file.

Examples:

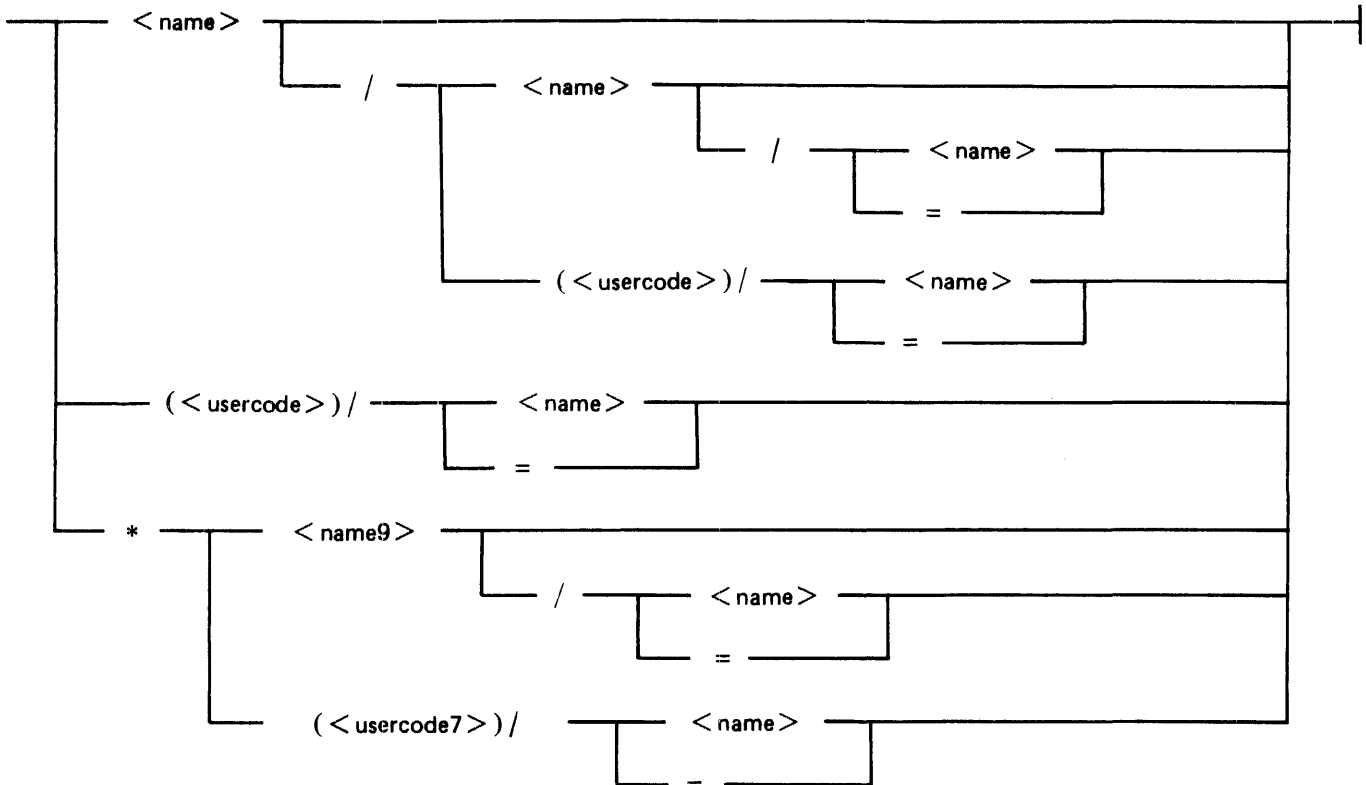
"EBCDIC"

"BCL"

<file-id> Specification

The <file-id> Specification identifies the name of the file to be listed.

Syntax:



Semantics:

name

This field can be any valid 10-character name.

name9

This field can be any valid 9-character name.

(usercode)

This field can be any valid 8-character usercode name.

(usercode7)

This field can be any valid 7-character usercode name.

*

The asterisk (*) character specifies that the file is located on the system disk.

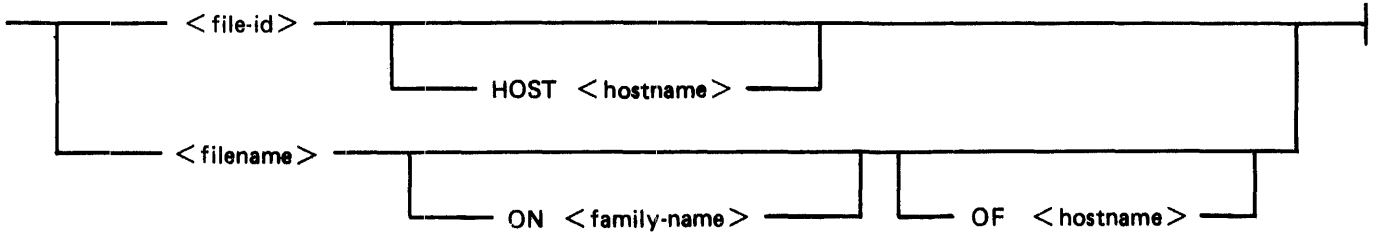
=

The equal sign (=) character specifies that all files with the same first-part name are to be listed. For example, if TEST/A and TEST/B are file names and TEST/= is specified, then both the TEST/A and TEST/B files are listed.

<file-title> Specification

The <file-title> Specification identifies the name of the file to be listed.

Syntax:



Semantics:

file-id

Refer to <file-id> Specification.

filename

Refer to <file-name> Specification.

HOST

The HOST keyword specifies that the file is located on the host system specified by <hostname>.

ON

The ON keyword specifies that the file is located on the disk specified by <family-name>.

OF

The OF keyword specifies that the file is located on the host system specified by <hostname>.

family-name

This field specifies the name of the disk that contains the file.

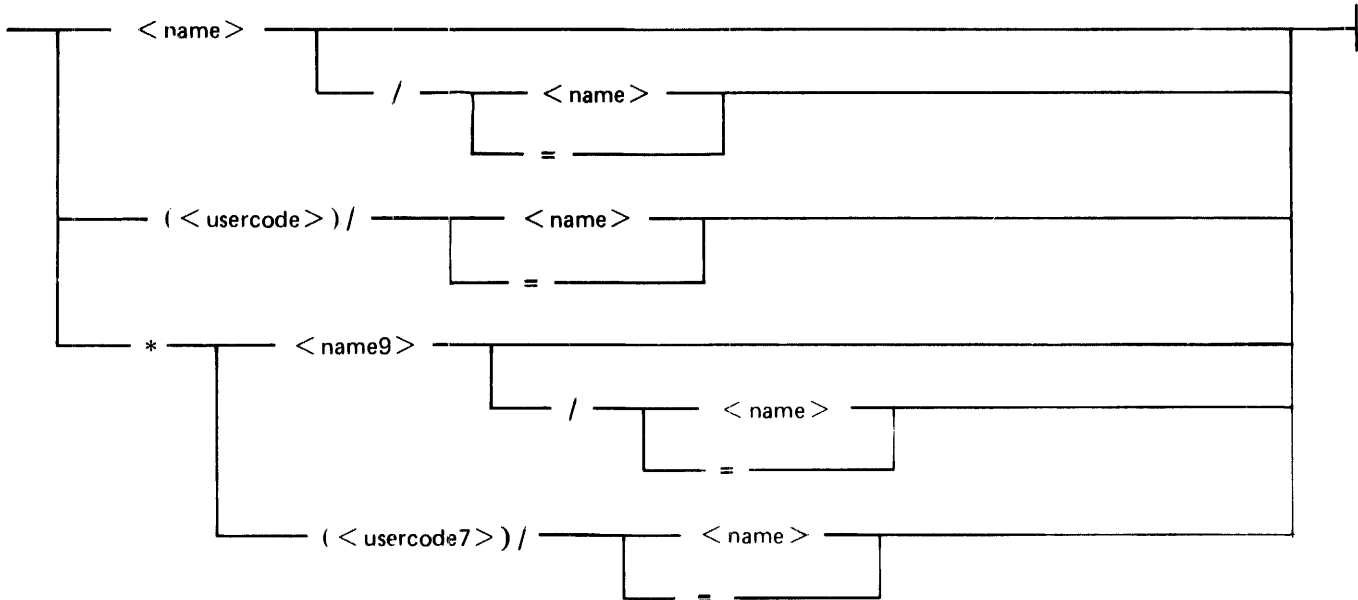
hostname

This field specifies the name of the remote host system where the file is located.

<file-name> Specification

The <file-name> Specification specifies the name of the file to be listed.

Syntax:



Semantics:

name

This field can be any valid 10-character name.

name9

This field can be any valid 9-character name.

(usercode)

This field can be any valid 8-character usercode name.

(usercode7)

This field can be any valid 7-character usercode name.

The asterisk (*) character specifies that the file is located on the system disk.

=

The equal sign (=) character specifies that all files with the same first-part name are to be listed. For example, if TEST/A and TEST/B are file names and TEST/= is specified, then both the TEST/A and TEST/B files are listed.

<hardware-type> Specification

The <hardware-type> Specification is used in the CAT and LIST commands and specifies media type for the input and output files.

Syntax:

<u>BINARY</u>
<u>CASS</u>
<u>CARD</u>
<u>CARD96</u>
<u>DISK</u>
MPF
MTP
MT7
MT9
MULTI
PAPER
PPT
TAPE
TAPE7
TAPE9

Semantics:

See Semantics for <routine-type> Specification.

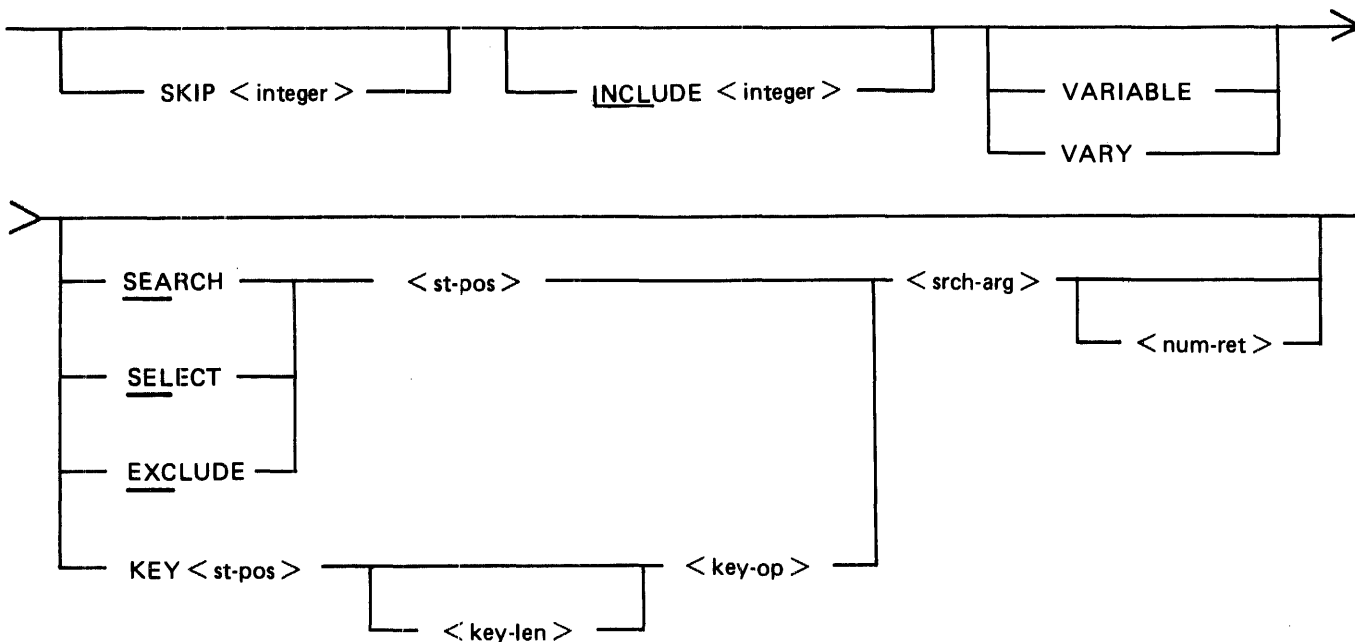
Examples:

DSK
DISK
MTP
TAPE9

<options> Specification

The options available with the DMPALL program are EXCLUDE, INCLUDE, KEY, SEARCH, SELECT, SKIP, and VARIABLE.

Syntax:



Semantics:

EXCLUDE

The EXCLUDE keyword specifies that only records that do not satisfy the entered parameters (within the range specified by the SKIP and INCLUDE options) be printed.

INCLUDE

The INCLUDE keyword specifies that the following <integer> is the number of records to be read. The default is to read until the end-of-file record is read.

KEY

The KEY keyword functions in a manner similar to the SELECT option; however, the parameters can specify more detailed selection criteria.

key-len

This field specifies the size of the compare field in the record for use with the KEY option. The maximum <key-len> that can be specified is 20. The units (bytes, digits, or bits) associated with the <key-len> are determined by the <srch-arg> type.

key-op

This field specifies the type of comparison to be performed between the compare field in the record and the <srch-arg> for use with the KEY option. The <key-op> field can be one of the following:

<key-op>	<Meaning>
GTR or >	Greater than
GEQ or >=	Greater than or equal
EQL or =	Equal
LEQ or <=	Less than or equal
LSS or <	Less than
NEQ	Not equal

num-ret

This field specifies the maximum number of records in which to stop the selective process. Care must be taken when using both the INCLUDE and <num-ret> options together, as the list operation terminates when either of the two options is satisfied.

SEARCH

The SEARCH keyword specifies that printing is to begin with the first record that satisfies the entered parameters, within the range specified by the SKIP and INCLUDE options.

srch-arg

This field specifies the data string against which the specified field in each record is compared. The field type is determined by the <srch-arg> type.

Alphanumeric data is delimited by either blanks or quotation marks. Hexadecimal data is delimited by at signs (@). Digit data and binary data are also delimited by @ signs but with mode indicators in parentheses after the first @. The mode indicator for digit data is 4; for binary data, it is 1. Some forms of <srch-arg> follow:

Argument Type	Field type	Sample Argument
Alphanumeric	Byte	"ABCDEF 1234"
Hexadecimal	Byte	@0123ABCDEF@
Digit	Digit	@(4)01@
Binary	Binary	@(1)11001@

SELECT

The SELECT keyword specifies that only records that satisfy the entered parameters (within the range specified by the SKIP and INCLUDE options) be printed.

SKIP

The SKIP keyword specifies that the following <integer> equals the number of records to skip before reading is to begin. Record 1 is the first record in the input file.

P A S S
@D7C1E202@
AF83C5C
5F078B8
B60F171
7C1E202

st-pos

This field specifies the starting position of the field to be compared in the input record. The first position in the record is relative position one. The numeric <st-pos> entry specifies either a byte, digit, or bit starting position, depending on the <srch-arg> type.

VARIABLE

The VARIABLE keyword specifies that the input tape or disk file has variable-length records. If the input file is a disk file, the DMPALL program checks the disk file header to verify that the file consists of variable-length records; if not, the VARIABLE keyword is ignored. If the CAT command is specified, then the base and concatenation files are variable-length record files. If the COPY command is specified, then the input and output files are variable-length record files.

VARY

The VARY keyword specifies that the input tape or disk file has variable-length records. If the input file is a disk file, the DMPALL program checks the disk file header to verify that the file consists of variable-length records; if not, the VARY keyword is ignored. If the CAT command is specified, then the base and concatenation files are variable-length record files. If the COPY command is specified, then the input and output files are variable-length record files.

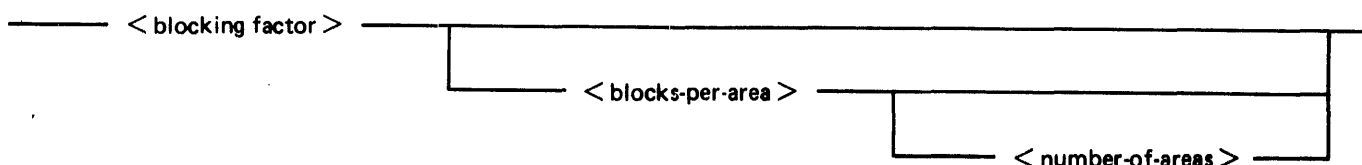
Examples:

```
LIST MASTERFILE ON USER SKIP 10 INCLUDE 250 ALFA DISK
COPY DISK TO TAPE MASTERFILE 90 20 EXTRACT 180 10 SELECT 1 "AAA"
COPY DISK TAPE TESTFILE TEST KEY 1 5 EQL "11111"
LIST TESTFILE ALFA
COPY DSKDSK SOURCE 80 9 NEWSOURCE 90 20
```

<output-areas-and-blocking> Specification

The <output-areas-and-blocking> Specification describes the blocking factor, blocks per area, and the number of areas for the output file.

Syntax:



Semantics:

blocking-factor

This field specifies the number of records per block for the file. If omitted, a blocking factor of one is assumed. If both the input and output files are either disk files or labeled B 1000/B 6000/B 7t000 tape files and the <output-record-size> is omitted, the <output-blocking-factor> is assumed to be the same as the <input-blocking-factor>.

blocks-per-area

This field specifies the number of blocks per area for disk files only. If omitted, 100 blocks per area is assumed. If both the input and output files are disk files and both the <output-record-length> and <output-blocking-factor> are omitted, the <blocks-per-area> is assumed to be the same as the blocks per area of the input file.

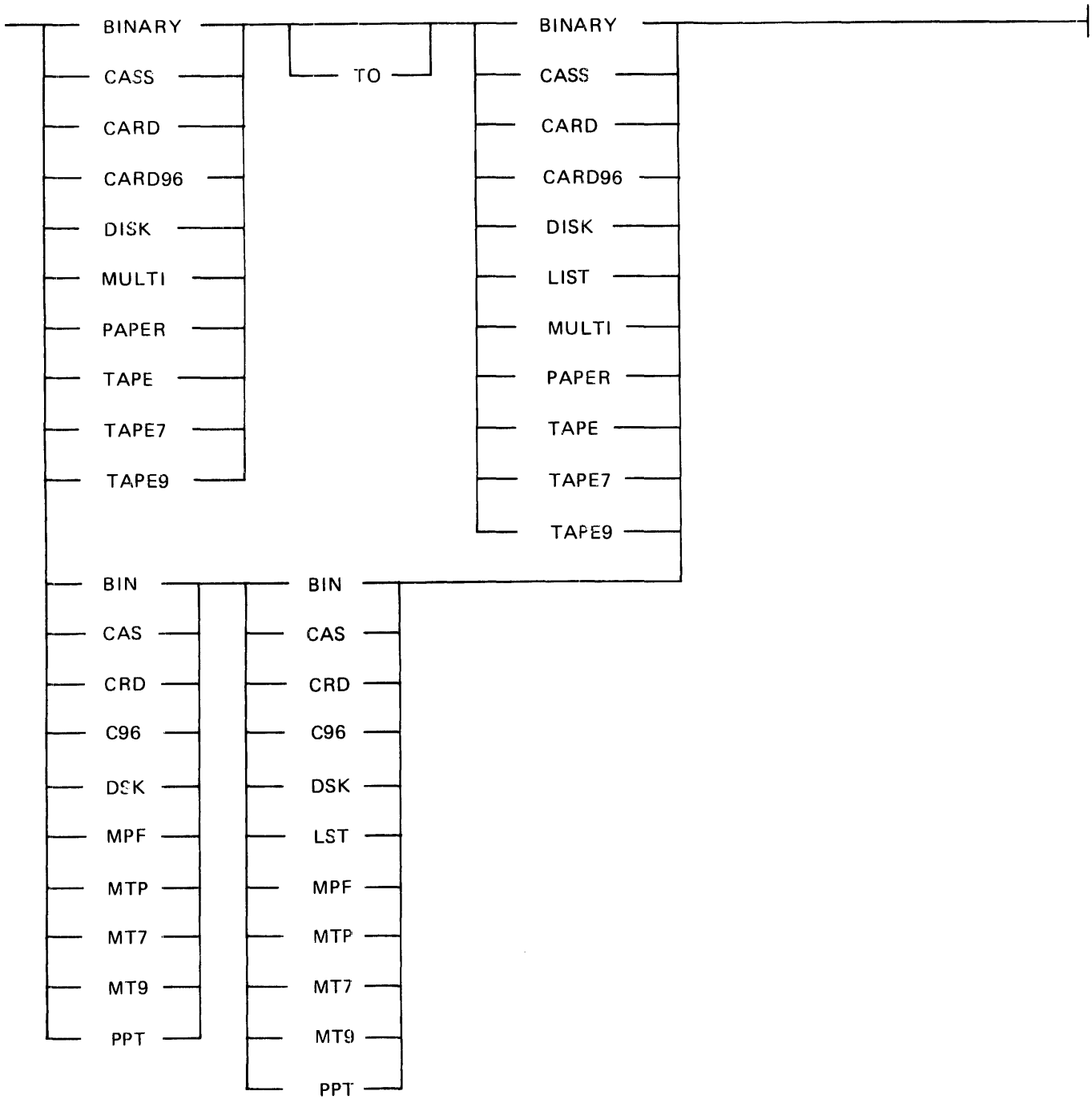
number-of-areas

This field specifies the number of disk areas for disk files only. If omitted, <number-of-areas> defaults to 25. If both the input and output files are disk files and the <output-record-length>, <output-blocking-factor>, and <blocks-per-area> are omitted, <number-of-areas> is assumed to be the same as the number of areas assigned to the input file.

<routine-type> Specification

The media types for the input and output files for the COPY (PERFORM) command are selected from the <routine-type> Specification. The medium specified first in the string pertains to the input file; the second pertains to the output file.

Syntax:



Semantics:

BIN	The BIN (or BINARY) keyword specifies that the hardware type is a binary card file. The card file is assumed to be 80-column and a card with END-OF-DECK punched in columns 1 through 11 must be the last card in the deck.
BINARY	See BIN.
CAS, CASS	The CAS (or CASS) keyword specifies that the hardware type is a cassette file.
CASS	See CAS.
CARD	The CARD (or CRD) keyword specifies that the hardware type is a card reader file.
CARD96	The CARD96 (or CRD96) keyword specifies that the hardware type is a 96-column card reader file.
CRD	See CARD.
CRD96	See CARD96.
DISK	The DISK (or DSK) keyword specifies that the hardware type is a disk file.
DSK	See DISK.
LIST	The LIST (or LST) keyword may be specified for an output. It causes the file, without heading or record numbers, to be listed on the line printer.
LST	See LIST.
MPF	The MPF (or MULTI) keyword specifies that the hardware type is a multi-pack disk file.
MTP	The MTP (or TAPE) keyword specifies that the hardware type is a magnetic tape file.
MT7	The MT7 (or TAPE7) keyword specifies that the hardware type is a 7-track magnetic tape file.
MT9	The MT9 (or TAPE9) keyword specifies that the hardware type is a 9-track magnetic tape file.
MULTI	See MPF.
PAPER	The PAPER (or PPT) keyword specifies that the hardware type is a paper tape file.
PPT	See PAPER.
TAPE	See MTP.
TAPE7	See MT7.
TAPE9	See MT9.

Pragmatics:

Notice that all keywords for <routine-type> are of two forms, a three-character short form and a greater-than-three-character long form.

The short form must be entered in concatenated pairs only (no spaces). Examples:

CRDMTP
DSKMTP

The long forms must be entered as pairs either separated by at least one space, or by the word TO with at least one space before and after it.

Examples:

CARD TO TAPE
CARD TAPE
DISK TO TAPE9
DISK DISK

If the <routine-type> option is omitted, both media (input and output) are assumed to be DISK.

<hardware-type> Specification

The <hardware-type> Specification is used in the CAT and LIST commands and specifies media type for the input and output files.

Syntax:

BINARY

FILES

Output and internal files are described.

Output Files

The following paragraphs describe the output files generated by the DMPALL program.

Disk Files

The DMPALL program alters copied disk files when requested and in the manner specified in the input specification. If there are no alterations requested, the disk file header retains the original creation date and file attributes.

Tape Files

The DMPALL program can write one file to a tape. This file is written to the tape in the format of a data file. This tape file is not a library tape as in the case of tape files created by the SYSTEM/COPY program.

Printer Listings

The LIST command includes heading information describing the file being printed, in addition to the data. A COPY command, which specifies LIST as the output device, generates a listing that contains only data (no heading information).

Card Files

The DMPALL program punches card files when CARD or C96 is specified as the output medium during a COPY command. Only data is punched. Records longer than 80 bytes are truncated when punched to an 80-column CARD, and records longer than 96 bytes are truncated when punched to a 96-column card (C96).

Paper Tape Files

Only data (no headers) is punched to paper tape files.

Internal Files

The internal files used by the DMPALL program follow:

Internal File Name	External File Name	Function
INPFILE	INPUT	Input file used when processing the CAT, COPY, and LIST commands.
OUTPFILE	OUTPUT	Output file used when processing the CAT and COPY commands.
BASEFILE	BASEFILE	A disk file used when processing the CAT command and is the file that the concatenation file is appended to.
LISTFILE	LIST	Output printer file used when processing the COPY or LIST command.
SPEC	SPEC	Input card file used to process DMPALL commands when program switch 0 = 1.
TAGFILE	TAGFILE	Input or output disk file of a tag file used for RPG (\$TAG) indexed files or ANSI 1968 COBOL index-sequential files. Used when TAG: key symbol is specified in the COPY or LIST commands.
NULFILE	NULFILE	This file is a dummy file used to check whether or not a disk pack is on-line.
DSKCRDS	DSKCRDS	This file is a workfile used to hold the actual file names when a LIST command is specified for a family files, such as, LIST B/= ON A or LIST A/B/=.
DISKFILE	DISK/SPEC	This file is a workfile that contains the specifications read in from the SPEC file.

ERROR MESSAGES

The following are the error messages generated by the DMPALL program.

<input-file-title> FILE NOT ON DISK

The input file is not in the disk directory.

MULTIFILE-ID OR PACK-ID NUMERIC – USE QUOTES AROUND IT

File titles that are numeric must be enclosed in double quotation mark (") characters.

INVALID HARDWARE TYPE OR OUTPUT FORMAT

The mnemonic for the hardware type or output format is incorrect, please re-enter.

NUMERIC VALUE MUST FOLLOW <option>

The SKIP, INCLUDE, SEARCH, SELECT, EXCLUDE, or KEY option did not contain a numeric parameter.

<command> ITEM SHOULD NOT APPEAR BEFORE LIST OR PERFORM

Commands to the DMPALL program must start with either PD, PERFORM, PFM, COPY, CPY, CAT, LIST, LST, LIST1, LST1, LIST2, or LST2.

<areas> AREAS EXCEEDS 105 MAXIMUM

A value of 105 for <areas> is the maximum number of output areas that can be specified.

SEARCH FIELD OUTSIDE RECORD

The search string starting position plus the string length is greater than the input record size. That is, <start-pos> + <length> is greater than <record-size>.

ILLEGAL HARDWARE NAME

The hardware name is not correct, please re-enter.

MISSING OR INVALID SECOND HARDWARE NAME <input-item>

The first hardware type was found, and the second hardware name was invalid; please re-enter.

<input-error> IS NOT VALID AFTER OUTPUT <file-title>

The hardware type was either invalid or treated as the input file title, or an invalid entry was detected after the second file title.

NO SPECS

A blank card was read from the specification deck.

<output-file-title> IS ON A SYSTEM DRIVE AND CANNOT BE A MULTI-PACK FILE

Output multipack files cannot be specified to the system disk, please re-enter.

<input-file-title> IS A SYSTEM LOG AND CANNOT BE COPIED OR LISTED

The system log file cannot be copied or listed. Use the LOGCONVERT utility to convert, or enter the LN system command to get a listing of the log activity.

INVALID RECORD LENGTH <record-length>

The maximum <record-length> allowed is 8000 bytes (4000 bytes if either the TAG: key symbol or ASCII conversion is specified).

NO RECORDS PROCESSED – END OF FILE REACHED

The input file is empty, has been skipped past end-of-file record, or no record satisfying the parameters specified for the <options> were found. If a copy operation was specified, the output file contains zero records.

EQUAL SIGN IS NOT VALID IN THIS CONTEXT

For the command entered, files must be specified to the DMPALL program one at a time.

<search-argument> IS NOT A VALID KEY ARGUMENT

<search-argument> had hexadecimal, digit, or bit data type specified and the data contains invalid characters for that type or the <search-argument> is not terminated with an at sign (@) character.

PARITY ERROR – REC.NUM: <record-number>

Indicates that the data read contained a parity error; processing continues.

<tag-file-title> I/O ERROR <record-number>

An input/output error occurred; processing continues.

INVALID KEY LENGTH <key-length>

Key <length> must be between 1 and 20, inclusive.

INVALID KEY OPERATOR <op>

The <key-op> field was not a valid entry.

INVALID TAG FILE <tag-file-title>

The <tag-file-title> file was not created with ANSI 1968 COBOL or RPGII (\$ TAG) naming conventions.

INVALID INDEX FILE <input-file-title>

The <input-file-title> file is not a disk file.

WARNING – <output-file-identifier> IS NOT AN IAD FILE

This is a warning message indicating that the input file to the COPY command was an Installation Allocated Disk (IAD) file. The output file created is not an IAD file.

BASE FILE IS NOT A DISK FILE

When using the CAT command to concatenate files, the <base-file-title> file (the second file named) must be a disk file.

PARAMETERS CANNOT BE SPECIFIED FOR BASE FILE

The parameters cannot be specified for the base file; please re-enter.

CONCATENATION ALLOWED ONLY ON DATA FILES

<output-file-title> IS NOT A DATA FILE.

The output file of a CAT command must be a data file. The concatenation file can be a non-data file; however, the resultant file is a data file.

<input-file-title> IS NOT A VARIABLE LENGTH DATA FILE

The input file is not a variable length record data file.

<output-file-title> FILE NOT ON DISK

The output file was not on disk, please load and re-enter.

<file-title> IS TOO LONG FOR FILE-NAME

The file title specified was too long, please re-enter.

PACK: <input-disk-name> NOT FOUND

The disk pack specified is not on-line, put on-line and re-enter.

DISK DIRECTORY READ PROBLEMS

An error was encountered while reading the disk directory. The SYSTEM/PANDA utility may be used to check the integrity of the disk pack.

MISSING HOSTNAME

The name of the host system was not specified in the HOST option.

<hostname> : TOO LONG FOR HOST_NAME

The value entered for <hostname> was greater than 17 characters. Please correct and re-enter.

EQUAL SIGN NOT ALLOWED IN FILE-ID FOR HOST FILES

The LIST command does not allow listing of a family of files from a remote host system. Please list one at a time.

HOSTNAME NOT ALLOWED ON OUT-FILE FOR "CAT" OPTION

The HOST option is not allowed for output file when the CAT command is specified.

PLEASE USE SYSTEM/COPY TO TRANSFER NON-DATA FILES ON BNA HOST

The DMPALL program does not transfer non-data files to a remote BNA system. Use the SYSTEM/COPY program with the HOSTNAME option.

NUMBER OF AREAS MUST BE 1 FOR CODE, INTERPRETER OR INTRINSIC FILES

The COPY command specified more than one area for the output file when the input file was a code, interpreter, or intrinsic file. This is a restriction on B 1000 operation.

CODE, INTERPRETER OR INTRINSIC FILES MUST HAVE A RECORD SIZE OF 180 BYTES

The COPY command specified a different value than 180 for the record length of the output file when the input file was a code, interpreter, or intrinsic file. This is a B 1000 operating system restriction.

OUTPUT BLOCKS PER AREA TOO SMALL

The value specified for <blocks-per-area> was too small; please increase and re-enter.

SKIP, INCLUDE AND OTHER OPTIONS NOT ALLOWED WITH CODE, INTERPRETER, OR INTRINSIC FILES

<options> cannot be specified for code, interpreter, or intrinsic files. This is a B 1000 operating system restriction.

INCOMPLETE FILE TITLE

The file title entered was not complete; please re-enter.

HOSTNAME NOT ALLOWED WITH TAG OR PUNCH FILENAMES

The HOST option does not allow the TAG option or output files of <hardware-type> CARD.

INVALID FILE TITLE

The file title entered was not correct; please re-enter.

INCONSISTENT FILE TITLES

The output file name syntax must be the same as the input file name syntax.

SECTION 17

FOREIGN/TAPECOPY

The FOREIGN/TAPECOPY program is designed to copy or list any 7-or 9-track magnetic tape. The program can copy the input tape to disk, line printer, or tape. Disk and line printer output files are in the standard B 1000 format and tape files are an exact copy of the input file. Data on the tape that is bounded by the specified delimiters is copied to a separate file. The operator is allowed to specify the tape format by using mnemonics to designate tape marks, header information, data, and labels. Input specifications can be entered from an ODT, remote terminal, or a card file.

OPERATING INSTRUCTIONS

The FOREIGN/TAPECOPY program uses dynamic memory for both input and output tape buffers. This allows the program to use only as much memory as required. When executing the FOREIGN/TAPECOPY program, enough memory for an input buffer plus an output buffer must be specified. Refer to the MEMORY instruction in volume 1 of the B 1000 Systems System Software Operation Guide. When specifying output blocking, add the size of one record. The input sizes are determined by the size of the first block on the input tape. If the amount of memory specified is not large enough, the program terminates with an appropriate message.

Table 17-1 lists the program switch settings and their meanings.

Table 17-1. FOREIGN/TAPECOPY Program Switches

Switch	Value	Meaning
0	0	Input from ODT
	1	Input from cards
7	0	No action
	1-15	Message displayed for each file copied
8	0	No action
	1-15	Program dump at abnormal terminate

Executing the FOREIGN/TAPECOPY program with switch 0=1 directs the program to a card file (CARDS) which must contain the input specifications. See Input Specifications in this section. A maximum of 20 input cards is allowed.

Executing the FOREIGN/TAPECOPY program with switch 0=0 results in the display of the following message:

ENTER INPUT SPECIFICATIONS OR "HELP"

The operator should enter either a string of information representing the input specifications, or the HELP command. A maximum of 1600 characters is allowed. Entering the HELP command causes the following messages to be displayed. The operator should respond with the appropriate information. Refer to Input Specifications in this section for details concerning input parameters. A blank or null response terminates the program at this point.

INPUT TAPE MNEMONIC? <MTx>

OUTPUT DEVICE? <DISK, PRINTER, TAPE>

DEFAULT SPECIFICATIONS? <YES, NO>

If the operator wishes to override the default values, the following input messages are displayed.

ENTER TAPE FORMAT? V <D, EOF, H, L, TM, V>

INPUT TRANSLATION? <ASCII8, ASCII7, EBCDIC, BCL, file name>

VARIABLE LENGTH RECORDS? <YES, NO>

OUTPUT BLOCKING?

BLOCKING FOR VARIABLE RECORDS: <MAX BLOCK LENGTH>

INPUT-OUTPUT BLOCKING: <characters per record> <records per block>

ENTER OPTIONS?

<PURGE <format list>,
SKIP(number of files),
INC(number of files),
@xx@,
FIRST, COPY,
COPYALL>

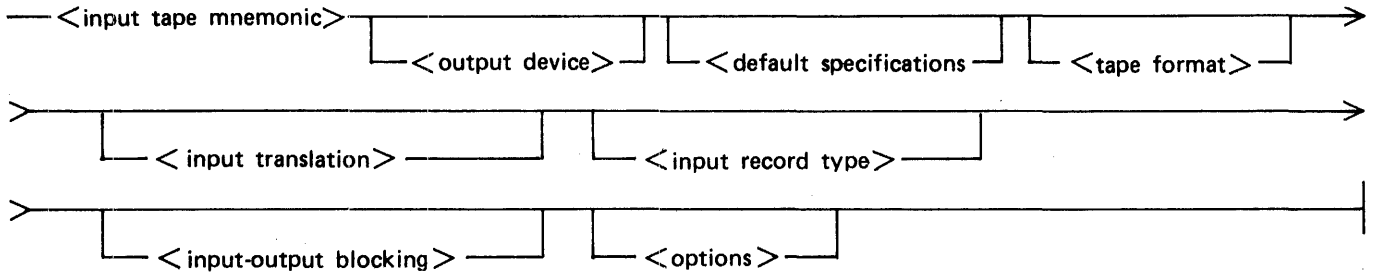
NOTE

A blank or null response, in most cases, is equivalent to a no answer or no-action answer.

If the output device is tape, only the relevant specifications are displayed.

INPUT SPECIFICATIONS

Syntax:



The input tape mnemonic is the only required specification. The remaining specifications are optional. If used, they must be in the sequence shown in the syntax diagram, with each separated from the next by at least one space.

Only columns 1 through 72 of a card file are scanned for specifications. Comments or sequence numbers may be entered in columns 73 through 80.

<input tape mnemonic>

— MT <unit> —

The input device must be a system-defined unit mnemonic for any 7-or 9-track tape, (for example, MTA). If the input device name is entered incorrectly, the operator can re-enter the appropriate device name. If the input device name is a valid device name but not present, the program hangs and waits for available hardware. The programmatic check of that device is not made until the copying actually begins. There is no default value for the input device name.

<output device>

— DISK —
— PRINTER —
— TAPE —

The output can be directed to disk or line printer using the words DISK, PRINTER, or PRINT. The default output device is DISK. All output files have the name OUTPUT as the first part of the file title. The file title for disk files begin with FILE0001 and are numbered sequentially.

Each file written to disk becomes a separate file. For printers, one backup print file is created with BOF and EOF messages separating the individual files. The print files contain a header consisting of blocking factor, file title, and input tape mnemonic for each file copied. The file title for a print file is FILES.

The output can also be directed to 9-track tape to create an exact copy of the 9-track input tape. Tape output is allowed only with the COPY or COPYALL options. It is not valid for 7-track tapes.

<default specification>

DEFAULT

The word DEF or DEFAULT is used to specify the default specification. This enables the operator to copy an EBCDIC tape with fixed-length records to the output device. The output records are unblocked with a new file being created for data that exists between single tapemarks. The FOREIGN/TAPECOPY program stops copying when it encounters a double tapemark. If the <default specification> is used, no other input specifications are required. If TAPE is specified as the output device along with DEFAULT, the COPY option is assumed with default blocking. PRINT and DISK output devices assume no options.

<tape format>

< tape format list >

<tape format list>

< tape parts list >

< number of repeats > / <tape parts list> /

<number of repeats>

<integer>

The FOREIGN/TAPECOPY program assumes that all files are separated by single tapemarks and that the end-of-reel marker is a double tapemark. Because all tapes do not follow this form, the operator may specify the format of the tape to be copied. The following mnemonics are used to explain the tape format:

Mnemonic	Meaning
D	Data portion of the tape
EOF	End-of-file information
H	Header information
L	Label
TM	Tape mark
V	Volume label

Any mnemonic can be optionally followed by <number of blocks> to specify the number of blocks to be included in that portion of the format. A zero block count is not valid. The FOREIGN/TAPECOPY program must rely on tapemarks to delimit information. The alternative to this is to specify the number of blocks for each portion of the file (specified by a mnemonic) between tapemarks. This causes the number of blocks specified to be placed in a separate file.

The FOREIGN/TAPECOPY program stops processing when it detects a double tapemark or a single tapemark immediately following an end-of-tape mark. If a tape does not conform to this standard, it is up to the operator to specify the physical limit of the input tape. This can be accomplished by specifying <number of repeats>. If <number of repeats> is specified, double tapemarks will not terminate the copy. If <number of repeats> is too large, the integrity of the data copied is unreliable; it is possible that the tape could run past the double tapemark and off the reel. It is suggested that the operator try his format with the FIRST option to a line printer to check for the correct operation. For example, a tape with one file would have a tape format as follows.

V H H TM D TM EOF EOF TM TM

Such a format results in the following files.

FILE0001	(V H H)
FILE0002	(D)
FILE0003	(EOF EOF)

The V H H is placed in the same file because a block count was not specified. FOREIGN/TAPECOPY copies the blocks until it reaches a tape mark.

Slashes (/) are used to enclose the repetitive parts of a tape format. A blank must delimit the repeat <tape parts list> from the preceding <tape parts list>. No space is allowed between the <number of repeats> and the slash.

For example, the same tape with n files would have the following tape format:

V /H H TM D TM EOF EOF TM/

This would result in the following files:

FILE0001	(V H H)
FILE0002	(D)
FILE0003	(EOF EOF)
FILE0004	(H H)
FILE0005	(D)
.	.
.	.
.	.
FILEnnnn	(EOF EOF)

If <number of repeats> was specified for the above format, the result would be:

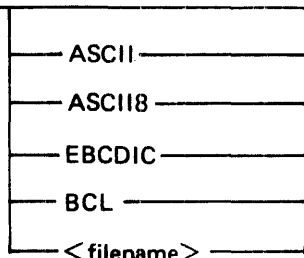
V 5/ H H TM D TM EOF EOF TM /

```

FILE0001 (V H H)
FILE0002 (D)
FILE0003 (EOF EOF)
FILE0004 (H H)
FILE0005 (D)
FILE0006 (EOF EOF)
FILE0007 (H H)
FILE0008 (D)
.
.
FILE00014 (D)
FILE00015 (EOF EOF)
    
```

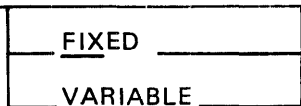
A maximum of 40 different tape format tokens can be used with any one run.

<input translation>



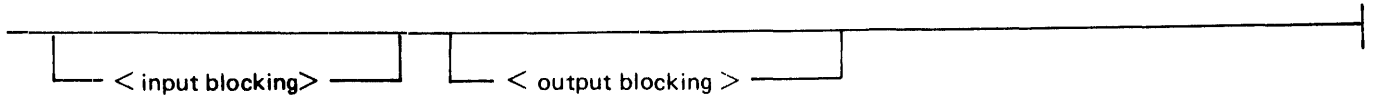
The input tape can be translated from BCL, ASCII-7, or ASCII-8 format to EBCDIC format. The default is EBCDIC, but the operator can specify translation from ASCII-7 and ASCII-8 for 9-track tapes, BCL for 7-track tape, or provide a unique translate table. The user-supplied translate table must appear as a system file title. All translate tables are to be created with the CREATE/TABLE program. The FOREIGN/TAPECOPY program looks for the translate table named TRANSLATE/EBCDIC, TRANSLATE/ASCII7, TRANSLATE/ASCII8, TRANSLATE/BCL, or <translate file name>. The second record of that file is used for translation. If the FOREIGN/TAPECOPY program is executed with a usercode, the name of the usercode overrides TRANSLATE as the multifile-id in the translate table file name.

<input record type>

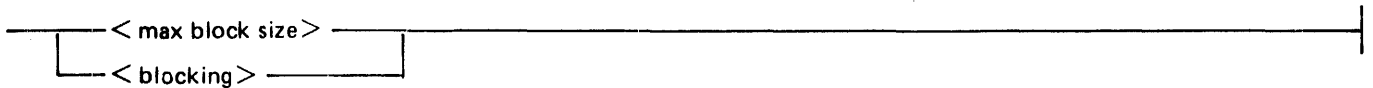


An input file can have fixed or variable length records. This can be specified by FIXED, FIX, VARIABLE, or VAR. The default is fixed length records for input. Output files are type FIXED only. The input record type is used for all files on the tape.

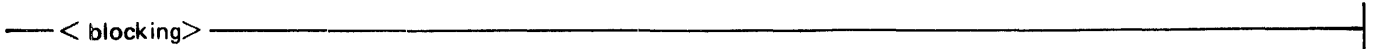
<input-output blocking>



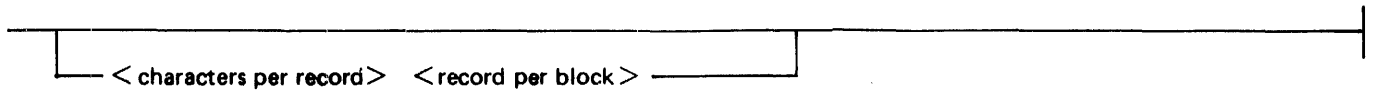
<input blocking>



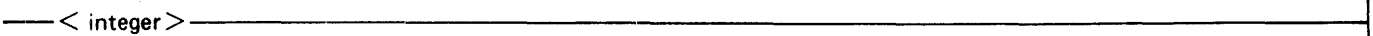
<output blocking>



<blocking>



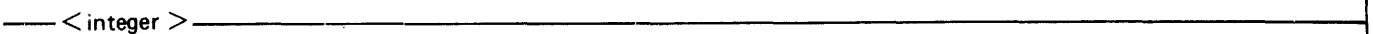
<max block size>



<characters per record>



<records per block>



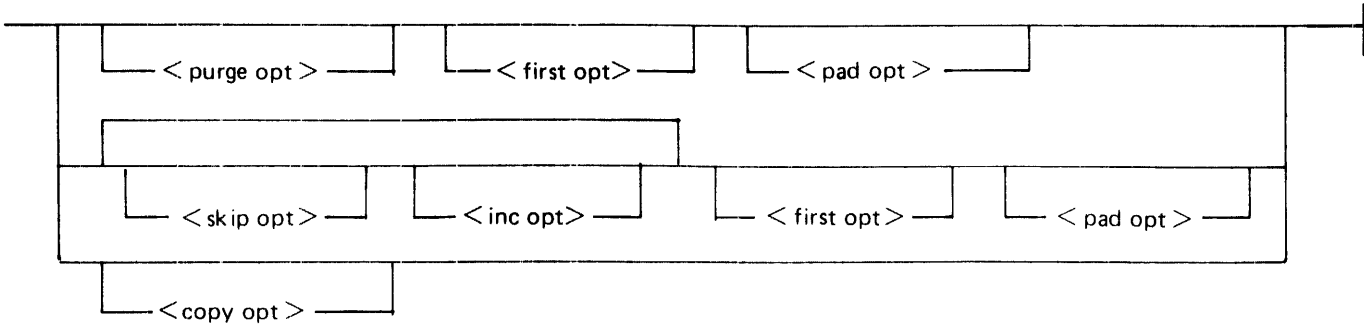
The blocking factors are used to determine block and record sizes on the input tape and output file. If no blocking specifications are provided, the default is unblocked, fixed length records. The first block of each file determines the length of the records. Blocking can be specified in characters per record and records per block. For instance, 80 2 is interpreted as 80 characters per record and 2 records per block. If a blocking factor is specified, it is applied to all files copied from that tape. If only one blocking specification is entered, it is applied to both the input and output files.

A maximum block size is required for variable length input blocks. The blocks, which can contain several variable or fixed-length records, is written to the output device as fixed-length records with the record size as the maximum block size specified. The records are padded either with zeros or the pad character, which is specified as an option.

If the blocksize is too large or too small for a block of data, the appropriate error message is printed. Refer to Error Messages in this section.

When using the COPY or COPYALL option, a maximum block size is used to determine the input buffer size only. If a block is shorter, it is written using its actual length.

<options>



<purge opt>

— PURGE <tape parts list> —————|

<skip opt>

— SKIP < number of files > —————|

<inc opt>

— INC < number of files > —————|

<first opt>

— FIRST —————|

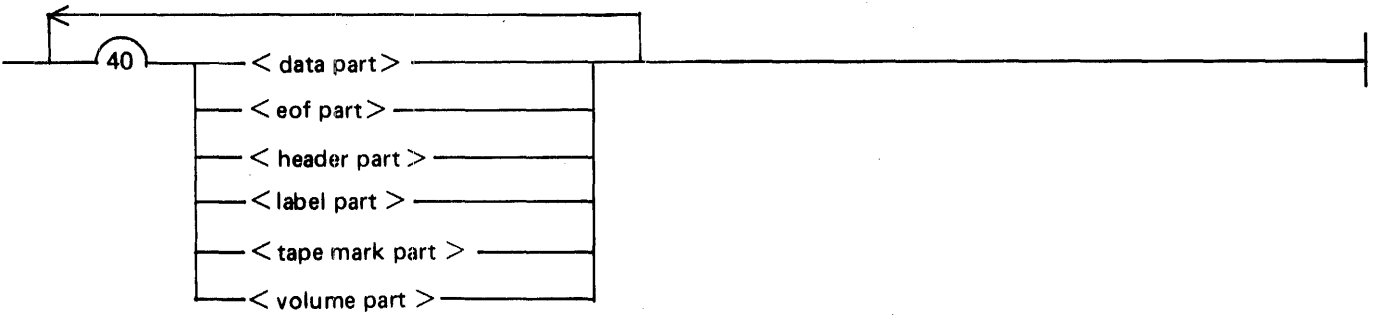
<pad opt>

— @ < EBCDIC character > < EBCDIC character > @ —————|

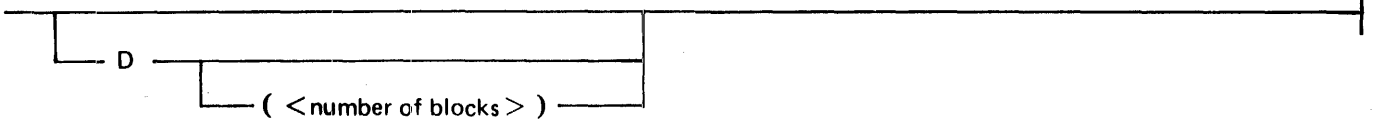
<copy opt>

— COPY —————|
 — COPYALL —————|

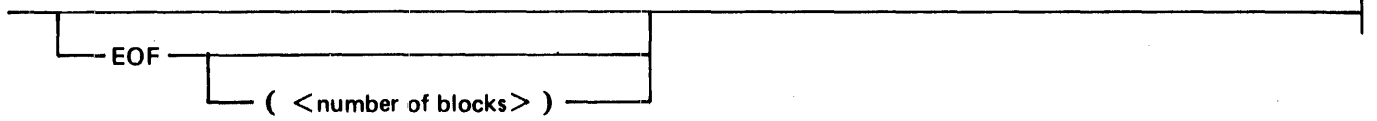
< tape parts list >



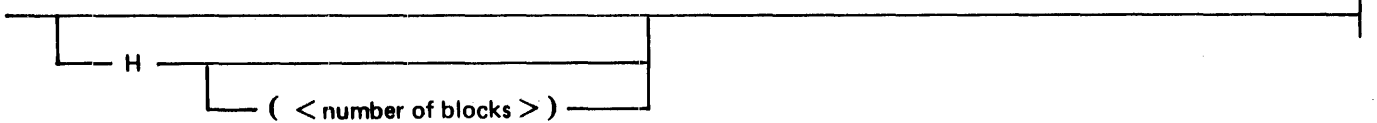
< data part >



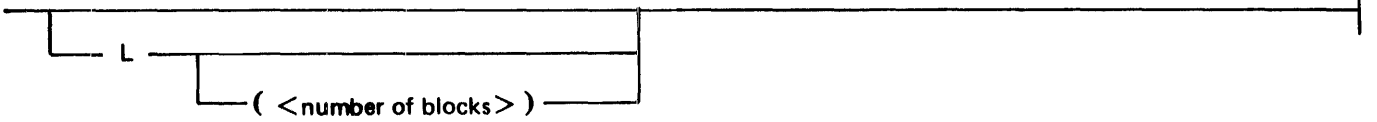
< eof part >



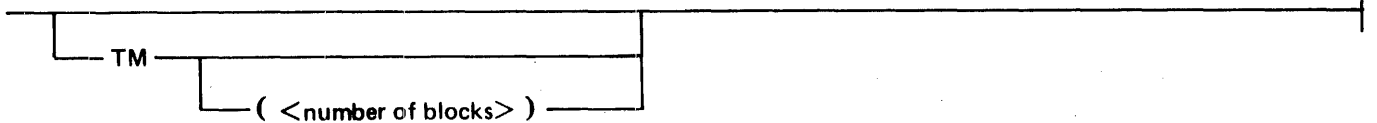
< header part >



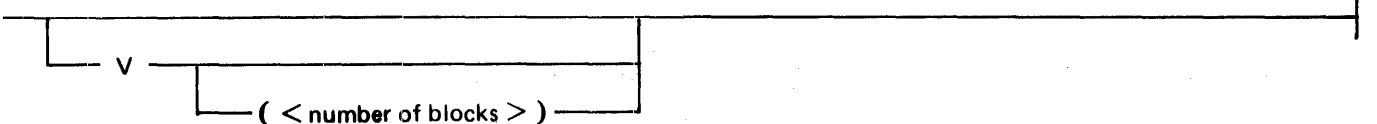
< label part >



< tape mark part >



< volume part >



<number of files>

—<integer>—

<file name>

—<packname> / <multifile-id> / <file-id>—

<packname>

—<name>—

<multifile-id>

—<name>
—<usercode>—

<file-id>

—<name>—

<usercode>

—(<name>)—

<number of blocks>

—<integer>—

<name>

—<letter> <name>—

<options>

PURGE <format list>

The PURGE option can be used with an operator-specified tape format to specify portions of the tape that are to be excluded from copying. Any portion may be excluded except tapemarks, which are not copied but are used only as a means of delimiting files. The PURGE option cannot be used in conjunction with the SKIP or INC options.

Example:

```
PURGE V  
PURGE H V EOF
```

SKIP <number of files>

The SKIP option is used to bypass the specified number of files before copying a file to the output device. A file is defined to be that data existing between single tapemarks. The SKIP option must be used with the INC option to select the required portions of the input tape. This option cannot be used with a tape format. A maximum of 100 SKIP and INC options can be specified in a given run.

INC <number of files>

The INC option specifies the number of files to be copied to the output device. A file is defined to be that data existing between single tapemarks. The INC option is used in conjunction with the SKIP option to select the specified portions of the input tape. This option cannot be used with a tape format. Only files with an INC option are copied if either the SKIP or INC option is used. In other words, the FOREIGN/TAPECOPY program stops processing at the end of the last SKIP or INC option.

FIRST

The FIRST option allows the operator to print only the first block of each file of the input tape. This option is used to check the contents of the input tape against the contents of the intended copy and can be used only when the output device is a line printer.

PAD

If padding is necessary, the FOREIGN/TAPECOPY program automatically pads with @00@. The PAD option specifies characters other than @00@ for padding. The operator-selected character must be specified in the form @<EBCDIC character><EBCDIC character>@, where <EBCDIC character> is a 4-bit hexadecimal number. The operator-selected character can also be specified for use in filling variable length blocks for fixed output. For example, if a blank is desired, the appropriate input would be @40@.

COPY

The COPY option provides an exact duplicate of the input tape. Translation, blocking, tape formats, or other options are not allowed. Double tapemarks terminate the copy. The input and output tape densities specified must match.

COPYALL

The COPYALL option provides an exact duplicate of the input tape. Translation, blocking, tape formats, or other options are not allowed, and the physical end of either the input or output tape terminates the copy. Input and output tape densities must match.

ERROR MESSAGES

UNKNOWN ERROR.

An error occurred for which no error message is provided. Submit an FTR.

INPUT TAPE MNEMONIC REQUIRED.

INVALID INPUT TAPE MNEMONIC.

Input mnemonic must be a system-defined unit mnemonic for a 7-or 9-track tape.

INVALID OUTPUT DEVICE MNEMONIC.

Output device must be DISK, PRINTER, or TAPE.

PLEASE ANSWER "YES" OR "NO".

A YES or NO answer was expected.

INVALID TRANSLATION SPECIFIED.

A standard translation type or operator translation file must be specified.

NUMERIC VALUE REQUIRED FOR BLOCKING.

TOO MANY BLOCKING PARAMETERS SPECIFIED

A maximum of four blocking parameters, two for input and two for output, can be specified.

INVALID OPTION SPECIFIED.

MISMATCH OF PARENTHESES.

INVALID TAPE FORMAT MNEMONIC.

Valid tape format mnemonics are D, EOF, H, L, TM, and V.

TAPE FORMAT EXCEEDS 60 MNEMONICS.

NUMERIC VALUE REQUIRED FOR "SKIP" AND "INC" OPTIONS.

INVALID PADDING CHARACTER SPECIFIED.

The pad character used must represent a 4-bit hexadecimal digit.

DOUBLE TAPEMARKS ENCOUNTERED.

Copying ends if a double tapemark is found.

NO OTHER INPUT SPECS ALLOWED WITH "DEFAULT" OPTION.

INPUT STRING EXCEEDED MAXIMUM LENGTH.

Card input must not exceed 20 cards.

INPUT TAPE NOT READY, PLEASE READY UNIT.

PARITY ERROR ON INPUT TAPE, PLEASE ENTER "YES" TO CONTINUE.

PROGRAM TERMINATING.

Program ending after fatal error.

TIMEOUT ERROR.

BLOCK SIZE TOO SMALL FOR ACTUAL DATA. (ONE TIME WARNING)

<MAXIMUM BLOCK-SIZE> REQUIRED FOR VARIABLE LENGTH RECORDS.

INPUT TAPE BLOCK LENGTH EXCEEDS 65528 BITS. ENTER "YES" TO TRUNCATE BLOCK AND CONTINUE.

DYNAMIC MEMORY TOO SMALL. RE-EXECUTE PROGRAM WITH MORE DYNAMIC MEMORY.

BLOCK SIZE TOO LARGE FOR ACTUAL DATA. (ONE TIME WARNING)

A "TAPE FORMAT" MAY NOT BE SPECIFIED WITH "SKIP" OR "INC" OPTIONS. "SKIP" OR "INC" OPTION IGNORED.

A "TAPE FORMAT" MUST BE SPECIFIED WITH A "PURGE" OPTION. "PURGE" OPTION IGNORED.

INVALID INPUT/OUTPUT BLOCKING SPECIFIED.

Input/output blocking factors must be non-zero.

PROGRAM EXCEEDED MAX NUMBER OF ELEMENTS IN TAPE FORMAT.

Increase number of elements on tape format.

TAPE MARK ENCOUNTERED BEFORE NUMBER OF BLOCKS SPECIFIED WERE COPIED.

END OF TAPE ENCOUNTERED. MAY LOSE ENDING LABELS.

Program will terminate at end of tape.

FIRST BLOCK OF FILE HAS A LENGTH OF ZERO.

Cannot open a file with a zero-length record.

USER SPECIFIED TAPE MARK NOT FOUND ON INPUT TAPE.

User specified input tape format does not match input tape format.

SIZE OF BLOCK EXCEEDS MAXIMUM LIMIT OF 65528 BITS.

"FIRST" OPTION ALLOWED WITH "PRINTER" OUTPUT ONLY.

END OF INPUT TAPE ENCOUNTERED.

Copying has stopped at end of input tape.

END OF OUTPUT TAPE ENCOUNTERED.

Copying has stopped at the end of the output tape.

MUST SPECIFY "COPY" OPTIONS WITH OUTPUT DEVICE "TAPE".

OUTPUT DEVICE "TAPE" MUST BE SPECIFIED WITH "COPY" OPTIONS.

PARITY ERROR ON OUTPUT TAPE, PLEASE ENTER "YES" TO CONTINUE.

INPUT AND OUTPUT TAPE DENSITIES MUST MATCH FOR "COPY".

PARITY ERROR ON 7-TRACK TAPE. CHANGING PARITY AND RETRYING.

Parity error encountered. Parity will be changed and processing continued.

EXACT COPY NOT ALLOWED ON 7-TRACK TAPES.

Internal Files

The following is a list of the name and function of each file used by the FOREIGN/TAPECOPY program.

Internal File Name	Description
CARDS	A card file containing input specifications
OUTPUTFILE	Modified to appropriate output device
INPUTFILE	A tape file containing input data
TRANSFILE	Translate file

SECTION 18

INITIALIZE/ANALYZER

The INITIALIZE/ANALYZER program is a normal-state utility program that creates a line printer listing of the addresses of disk sectors that were relocated or removed during initialization or reconfiguration. The disk pack or disk cartridge that is to be analyzed must have been initialized by a Mark 8.0 or later version of the SYSTEM/DISK-INIT or PACK/INIT programs. The INITIALIZE/ANALYZER program can analyze a disk while other programs are accessing the same disk.

OPERATING INSTRUCTIONS

The INITIALIZE/ANALYZER utility program is executed from the ODT. Several disks can be analyzed during a single execution of the program, one after another. One printer backup file is created that describes each disk analyzed.

Upon execution, the INITIALIZE/ANALYZER program issues the following prompt message on the operator display terminal (ODT):

ENTER INPUT DRIVE

The operator responds by entering an AC or AX command with the unit mnemonic that names the drive containing the disk to be analyzed. Unit mnemonics accepted by the INITIALIZE/ANALYZER program are DCx and DPx, where x is replaced by a letter that references an existing disk drive. For example, the letter A references drive 0.

The input drive prompt is repeated after the specified disk is analyzed. A null response to this prompt (a blank AX message) causes the program to go to end of job (EOJ).

PROGRAM OUTPUT

A printer listing is generated that describes each disk, naming each relocated sector and each removed sector, summarizes the number of relocated and removed sectors, and then lists label information that gives the identifier, serial number, type, and initialization date of the disk.

ERROR MESSAGES

INVALID RESPONSE – TRY AGAIN

An invalid unit mnemonic was entered, or the disk named could not be analyzed. Re-enter the drive specification.

BAD LABEL OR INITIALIZED VERSION – <drive>

Either the disk has a bad label, or it was initialized with an initializer program version other than a Mark 8.0 or later release of the SYSTEM/DISK-INIT or PACK/INIT programs. The disk must be reinitialized before the INITIALIZE/ANALYZER program can analyze the disk.

SECTION 19

LOGCONVERT

The LOGCONVERT program is a normal-state utility program that extracts useful items of information from a system log file to create another log file that presents this information in a format that is readable by COBOL and RPGII programs.

The LOGCONVERT program processes a SYSTEM/LOG file with a file name of LOG/#<integer> and produces a new file with a file name of NEWLOG/#<integer>. The LOGCONVERT program operates on SYSTEM/LOG files that were transferred by means of the LG or the TL system commands, but does not operate on the currently active SYSTEM/LOG file.

GENERAL INFORMATION

Records in the log file are generated as the information that they contain becomes available, which means that those records pertaining to a given job can be widely scattered throughout the log file. To simplify operator processing, a sort operation is invoked to restructure the NEWLOG/#<integer> file after it is filled, so that all records pertaining to a given job are grouped by job and, within each group, in the original sequence. This sort can be bypassed by setting program switch 9 to 1 when executing the LOGCONVERT program.

The input and output files have the internal identifiers LOGFILE and NEWLOGFILE, respectively. Each is declared with two buffers. The NEWLOGFILE file is automatically directed to the second drive by the MCPPII when multiple system disk drives have been assigned with the SD system command. Installations with only one system disk drive can minimize disk arm movement and, consequently, achieve substantial performance improvement if the NEWLOGFILE file is file-equated to a user disk through file equation of the PACK.ID (PID) file attribute.

OPERATING INSTRUCTIONS

The program is executed by entering

```
EX LOGCONVERT
```

from the ODT and then, in response to the ACCEPT message, entering <integer>. (<integer> is a six-digit number; leading zeros, if any, are optional.) Invalid responses result in error messages.

The user can monitor program progress at any time after the start of processing by entering STATUS or ST following an AX. The LOGCONVERT program responds by displaying the current input record count.

Output File Format

The LOGCONVERT program executes most efficiently if the area size of the log, as specified by means of the SL system command, is an integral multiple of 5. The NEWLOG/#<integer> file contains 180-byte records blocked 5.

Error and Advisory Messages

ENTER LOG NUMBER

The integer representing the log to be processed is requested. Input is free-format with leading zeros optional. On an invalid entry, one of the following messages is displayed followed by another request for input prefixed by the phrase

LET'S TRY AGAIN ...

LOG # CAN'T BE ZERO

The MCP assigns log numbers sequentially from 1 through 999999. Zero is invalid.

LOG NUMBER TOO LONG - MAX 6 DIGITS

The integer identifying a transferred file cannot exceed six digits.

<token> ISN'T NUMERIC

Non-numeric data was input in response to the log identification request.

NULL INPUT...PROGRAM EOJ

A null input (transmission of an empty line) is considered a request to terminate by LOGCONVERT.

<current-record> of <total-records> RECORDS CONVERTED

This is the response to a STATUS or ST request.

HUH?

This is the way LOGCONVERT responds to an invalid entry after processing begins.

RECORD TYPES

Eleven distinct record types, identified by the first two digits of each record, can be found within the NEWLOG/#<integer> file:

CLEAR/START record

One for each CLEAR/START operation performed during the scope of the log.

Schedule record

One for each job or task that is scheduled for execution.

BOJ record

One for each job or task that reaches beginning of job (BOJ).

Open record

One for each instance of a file being opened by a program.

Close record

One for each time a file is closed by a program.

Data Management System Statistics record

One for each structure used within a particular DMSII data base when the data base is closed.

Data Management System Open/Close record

One for each instance of a DMSII program opening or closing a data base.

Peripheral Assignment/Release record

One for each time an SDL or SDL2 program using special Initializer I/O communications, such as the SYSTEM/PANDA or SYSTEM/DISK-INIT programs, assigns a peripheral unit for its own exclusive use, or releases the peripheral back to the system.

EOJ record

One for each program (1) reaching end of job (EOJ), (2) being terminated due to a CLEAR/START operation, or (3) being removed from the schedules (RS system command).

Comment record

One for each operator comment placed into the log through the LC ODT command.

Log Transfer record

The last record in the file, identifying the next log file by a sequential integer maintained by the operating system (MCPII).

RECORD FORMAT TABLES

Generic descriptions of the record formats found in the output file are given in tables 19-1 through 19-11. Within these tables, references are made to hardware types, file types, access modes, and execute types; these are summarized in tables 19-12 through 19-15.

In the record format tables, data type N represents a packed numeric field (4-bit digit with high-order sign), and data type A represents an 8-bit alphanumeric field (EBCDIC). The 30-character A fields all represent the standard <pack-id>/<family-id>/<file-id> triples of ten characters each. The 16-digit N fields represent three distinct subfields: four digits each for the last two numbers in the year and the Julian day and eight digits for the time in tenths of seconds past midnight.

Specific COBOL, COBOL74 and RPG data layouts follow the tables.

Table 19-1. LOGCONVERT CLEAR/START Record Format

Field Name	Field Length	Data Type	Value and Function
Record Type	1	A	A = CLEAR/START record.
Job Accounting Number	8	N	of the next job to be scheduled.
Job Number	6	N	of the next job to be scheduled.
Time	16	N	YYYYDDDDTTTTTTTT
MCPII Name	30	A	
Interpreter Name	30	A	of the MCP's interpreter.
MCPII Version	10	A	MARK.LEVEL.PATCH (e.g ., 11.0.2)
S-Memory Size	10	N	in bits.
GISMO Name	30	A	
Micro MCPII Name	30	A	
New Log Flag	2	N	Indicates first CLEAR/START in log.

Table 19-2. LOGCONVERT SCHEDULE Record Format

Field Name	Data Length	Type	Value and Function
Record Type	1	A	B = schedule record.
Job Accounting Number	8	N	
Job Number	6	N	
Time	16	N	YYYYDDDDTTTTTTTT
Program Name	30	A	
Object Program Name	30	A	if execute type is a compilation.
Schedule Priority	4	N	0-15
Execute Type	2	N	Refer to table 19-15.
Static Memory	10	N	in bits.
Dynamic Memory	10	N	
Total Memory	10	N	Run Structure size.
Charge Number	8	N	0-9999999
Virtual Disk	6	N	segments for program data overlays.
Number of Files	4	N	
Interpreter Name	30	A	
Date of Compilation	16	N	
Parent Job Number	6	N	

Table 19-3. LOGCONVERT BOJ Record Format

Field Name	Field Length	Data Type	Value and Function
Record Type	1	A	C – beginning of Job record.
Job Accounting Number	8	N	
Job Number	6	N	
Time	16	N	YYYYDDDDTTTTTTTT
Program Name	30	A	
Object Program Name	30	A	if this is a compilation.
Execute Type	2	N	Refer to table 19-15.
Processor Priority	4	N	0-15
Memory Priority	4	N	0-15
Charge Number	8	N	0-9999999
Virtual Disk	6	N	in segments.
Interpreter Name	30	A	

Table 19-4. LOGCONVERT File Open Record Format

Field Name	Field Length	Date Type	Value and Function
Record Type	1	A	D = Open record.
Job Accounting Number	8	N	
Job Number	6	N	
Time	16	N	YYYYDDDDTTTTTTTT
File Number	4	N	0 if first file declared in program.
Internal Name	10	A	
External Name	30	A	
Hardware Type	4	N	Refer to table 19-12.
Variable Record File	2	N	As with all flags, 1 = TRUE.
Pseudo Reader File	2	N	
DMS	2	N	DMSII file.
Emulator Tape	2	N	To be processed by emulator tape communicates in SDL.
Record Size	6	N	in bits.
Blocking Factor	6	N	
Maximum Block Size	6	N	For variable record files, RSZ*R.B.
Unit Name	6	A	Explicit user peripheral designation
Number of Stations	4	N	
Access Mode	4	N	Refer to table 19-14.
Number of Areas	4	N	Declared for disk file.
Blocks per Area	8	N	
Autoprint Flag	2	N	
Filler	11	A	
File Type	4	N	Refer to table 19-13.
Serial Number	6	A	
Queue Family Size	4	N	
Queue Max Messages	4	N	
Securitytype	2	N	0 = PUBLIC, 1 = PRIVATE
Securityuse	2	N	0 = INPUT/OUTPUT, 1 = INPUT, 2 = OUTPUT
Number of Buffers	4	N	
Memory Space Required	8	N	Bits: FIB + buffers + I/O Descriptor
Open Input Flag	2	N	
Open Output Flag	2	N	
New File Flag	2	N	
Open with Punch Flag	2	N	
Open with Print	2	N	
Open No Rewind Flag	2	N	
Open Reverse Flag	2	N	
Open Lock Flag	2	N	
Open Lockout Flag	2	N	
Filler	3	A	
Open with Interpret	2	N	
Open with Stackers	2	N	
Open Insecure	2	N	
Open On Behalf Of	2	N	
Open Extend	2	N	
Filler	19	A	

Table 19-5. LOGCONVERT File Close Record Format

Field Name	Field Length	Data Type	Value and Function
Record Type	1	A	E = Close
Job Accounting Number	8	N	
Job Number	6	N	
Time	16	N	YYYYDDDDTTTTTTTT
File Number	4	N	0 is first file declared in program.
Internal Name	10	A	
External Name	30	A	
Hardware Type	4	N	Refer to table 19-12.
Variable Record File	2	N	As with all flags, 1 = TRUE.
Pseudo Reader File	2	N	
DMS	2	N	DMSII file.
Emulator Tape	2	N	Was processed by emulator tape communicates in SDL.
Record Size	6	N	in bits.
Blocking Factor	6	N	
Maximum Block Size	6	N	For variable record files, RSZ*R.B
Unit Name	6	A	Explicit user peripheral designation
Number of Stations	4	N	
Access Mode	4	N	Refer to table 19-14.
Number of Areas	4	N	Declared for disk file.
Blocks per Area	8	N	
Autoprint Flag	2	N	
Record Count	8	N	Logical I/O's.
Block Count	8	N	Physical I/O's.
Errors	6	N	
File Type	4	N	Refer to table 19-13.
Serial Number	6	A	
Queue Family Size	4	N	
Queue Max Messages	4	N	
Protection	2	N	
Protection I/O	2	N	
Number of Buffers	4	N	
Memory Space Required	8	N	Bits: FIB + buffers + I/O descriptor
Close Reel Flag	2	N	
Close Release Flag	2	N	
Close with Purge Flag	2	N	
Close Remove Flag	2	N	
Close with Crunch Flag	2	N	
Close No Rewind Flag	2	N	
Close Insecure	2	N	
Close with Lock Flag	2	N	
Close Conditional	2	N	
Close Rollout Flag	2	N	

Table 19-5. LOGCONVERT File Close Record Format (Cont)

Field Name	Field Length	Data Type	Value and Function
DMS Audit Switch Flag	2	N	
Filler	1	A	
Close Unavailable	2	N	
Filler	18	A	
Close Report Errors on Output	2	N	
Close for SORT	2	N	
MCPII Close due to EOJ	2	N	
Close Backup Exceeded	2	N	
Close Subport	2	N	
Protection Wait Count	4	N	

Table 19-6. LOGCONVERT DMS Statistics Record Format

Field Name	Field Length	Data Type	Value and Function
Record Type	1	A	F = DMS Statistics record.
Job Accounting Number	8	N	
Job Number	6	N	
Time	16	N	YYYYDDDDTTTTTTTT
Database Name	10	A	
Structure Number	4	N	
Invoke Number	4	N	
Structure Close Flag	2	N	
Structure Name	17	A	
Filename	30	A	
Hardware Type	4	N	Refer to table 19-12.
Unit Name	6	A	
Number of Random Finds	10	N	
Number of Sequential Finds	10	N	
Number of Inserts	10	N	
Number of Updates	10	N	
Number of Deletes	10	N	
Number of Sys. Changes	10	N	
Number of Exceptions	10	N	
Number of Logical Reads	10	N	
Number of Logical Writes	10	N	
Number of Physical Reads	10	N	
Number of Physical Writes	10	N	

Table 19-7. LOGCONVERT DMS Open/Close Record Format

Field Name	Field Length	Data Type	Value and Function
Record Type	1	A	G = DMS open or close record.
Job Accounting Number	8	N	
Job Number	6	N	
Time	16	N	YYYYDDDDTTTTTTTT
Physical Database Name	10	A	
Logical Database Name	17	A	
Close Flag	2	N	0 = open, 1 = close
Update Flag	2	N	
Audit Flag	2	N	
Hardware Type	4	N	Refer to table 19-12.
Unit Name	6	A	
Pack Name	10	A	
Number of Update Open	10	N	
Number of Non-update Op	10	N	
DMCP	10	N	
Exception Count	10	N	
Transaction Count	10	N	
Transaction State Time	10	N	
Contention Count	10	N	
Contention Wait Time	10	N	
IO Wait Count	10	N	
IO Wait Time	10	N	

Table 19-8. LOGCONVERT Peripheral Assignment/Release R. F. Record Format

Field Name	Field Length	Data Type	Value and Function
Record Type	1	A	H = Peripheral assign/release record.
Job Accounting Number	8	N	
Job Number	6	N	
Time	16	N	YYYYDDDDTTTTTTTT
Hardware Type	4	N	Refer to table 19-12.
Unit Name	6	A	The unit that was assigned/released.
Action	2	N	0 = assigned, 1 = released

Table 19-9. LOGCONVERT EOJ Record Format

Field Name	Field Length	Data Type	Value and Function
Record Type	1	A	I=End of Job record.
Job Accounting Number	8	N	
Job Number	6	N	
Time	16	N	YYYYDDDDTTTTTTTT
Program Name	30	A	
Object Program Name	30	A	if this was a compilation.
Execute Type	2	N	Refer to table 19-15.
End of Job Reason	2	N	0=normal EOJ 1=DS=ed 2=program error (DS or DP condition) 3=execution aborted by CLEAR/START. 4=RS=ed. 5=death in family. 6=In schedule at CLEAR/START.
Processor Time	8	N	in tenths of seconds.
Code Overlays	8	N	
Data Overlays	8	N	
Successor Program	30	A	Next program to run in a job stream.
EOJ Diagnostic Message	60	A	Specific "DS or DP" message.

Table 19-10. LOGCONVERT Log Comment Record Format

Field Type	Field Length	Data Type	Value of Function
Record Type	1	A	J=Comment Record.
Filler	7	A	
Time	16	N	YYYYDDDDTTTTTTTT
Comment Length	4	N	in characters.
Comment Text	162	A	

Table 19-11. LOGCONVERT Log Transfer Record Format

Field Name	Field Length	Data Type	Value and Function
Record Type	1	A	K=Log Transfer record
Filler	7	A	Hexadecimal @FF@' s
Time	16	N	YYYYDDDDTTTTTTTT
Next Logfile Name	30	A	

Table 19-12. LOGCONVERT Hardware Type Definitions

Value	Device
0	Dummy file
1	80-column data recorder
2	80-column card punch
3	Not used
4	Diskette (floppy disk)
5	96-column reader/punch/printer
6	Paper tape reader
7	Paper tape reader
8	Printer, other than B 1247-4 printer control
9	MICR reader-sorter
10	MICR reader-sorter
11	Disk file (any head-per-track unit)
12	Head-per-track disk
13	Disk cartridge (Disk Cartridge Control-2 or -3)
14	Disk cartridge (Disk Cartridge Control-1)
15	Disk pack (205, 206, or 225)
16	Disk pack (any available disk pack)
17	Disk (any available disk of any kind)
18	B 9470 Head-per-track disk (DFC-3 -- 5N disk)
19	96-column card reader
20	Paper tape punch
21	80-column card reader
22	Hard copy ODT (TTY -- no longer supported)
23	CRT ODT
24	9-track NRZ tape
25	7-track tape (upright drive)
26	9-track PE tape
27	Any tape unit
28	Any 9-track tape unit
29	7-track tape (MTC-1)
30	Cassette
31	Printer, B 1247-4 printer control (PC-5)
32	Disk pack (DSC -- 206, 207)
33	Printer Control 7
34	Burroughs Super Mini (Liege only)
35	Disk (DC-1 -- Liege only)
36	PE tape (MTC-6)
37	Printer control in DSC-2
38	80-column card reader (CRC-1T)
39-59	not used
60	Port file
61	Queue file
63	Remote

Table 19-13. LOGCONVERT File Type Definitions

System Files

Value	Meaning
1	Log
2	Directory (entry points to sub-directory)
3	Pseudo reader deck
4	Backup print
5	Backup punch
6	Dump type
10	Inactive log
12	Intrinsic
16	Usercode
23	MCP temporary
24	Job log
25-59	Reserved

Data Files

Value	Meaning
0	Unspecified data file
9	Data file
11	Variable-length
13	DMS
14	DMS dictionary
15	DMS audit
17	Relative
18	Index sequential global
19	Index sequential data set
20	Index sequential index
21	Indexed tag
22	Indexed data
60-73	Symbol Files as follows:
60	SDL
61	COBOL68
62	RPG
63	NDL
64	FORTTRAN
65	MIL
66	BASIC
67	UPL
68	COBOL74
69	FORTTRAN77
70	IBASIC
71	DASDL
72	PASCAL
73	SDL2
74	Non-native
75	IBASIC internal
76	FORTTRAN77 unformatted
77	COBDMS library
78	RPGDMS library
79	Network information
80	NDL library
81	Pascal interchange
82	Pascal module
83	SORT symbol
84	Sequential
85	JOB symbol
86-109	Reserved for additional data files.

Code Files

Value	Meaning				
8	Code file				
110-130	Code Files as follows:				
110	SDL	117	UPL	124	MCS
111	COBOL68	118	COBOL74	125	non-native
112	RPG	119	FORTRAN77	126	B500
113	NDL	120	IBASIC	127	IBM1400
114	FORTRAN	121	PASCAL	128	SORT
115	MIL	122	SDL2	129	unbound
116	BASIC	123	SMCP	130	JOB
131-139	Reserved for additional code files.				

Microcode Files

Value	Meaning				
7	Interpreter				
140-156	Interpreter Files as follows:				
140	SDL	146	UPL	152	see note
141	COBOL68	147	COBOL74	153	see note
142	RPG	148	FORTRAN77	154	see note
143	see note	149	IBASIC	155	B500
144	FORTRAN	150	PASCAL	156	IBM1400
145	BASIC	151	SDL2		
note:	143	NDL microcode			
	152	Micro MCP			
	153	GISMO			
	154	System Initializer			
157-169	Reserved for additional microcode files.				
170-255	Reserved for additional file types.				

Table 19-14. LOGCONVERT Access Mode Definitions

Value	Meaning
0	Serial
1	Random
2	I/O sequential
3	Input/stackers data recorder
4	Input/output data recorder
5	Emulator tape
6	Delayed random
7	Reserved for extended I/O sequential which is now implemented by setting FPB access = 0 and FPB extended sequential = 1.
8	Dynamic

Table 19-15. LOGCONVERT Execute Type Definitions

Value	Meaning
1	Regular execution
2	Compile and go
3	Compile for syntax
4	Compile to library
5	Compile and save
6	Execute (go) phase of compile and go
7	Execute (go) phase of compile and save
8	Interprogram communication (IPC) call

SAMPLE PROGRAM RECORD DECLARATIONS

The following are sample program record declarations for COBOL, COBOL74, and RPGII programs.

COBOL Record Formats

The following sample COBOL and COBOL74 declarations correspond to the record formats previously detailed in tables 19-1 through 19-15. They can be used by COBOL and COBOL74 programs that process the output of the LOGCONVERT program.

Clear/Start

```

01      CLEAR-START RECORD.
      02  CS-REC-TYPE                PIC X.
      02  CS-JOB-ACCOUNTING-NUMBER  PIC S9(7) CMP.
      02  CS-JOB-NUMBER             PIC S9(5) CMP.
      02  CS-YEAR                   PIC S9(3) CMP.
      02  CS-JULIAN-DAY             PIC S9(3) CMP.
      02  CS-TIME                   PIC S9(7) CMP.
      02  CS-MCP-NAME               PIC X(30).
      02  CS-INTERP-NAME            PIC X(30).
      02  CS-MCP-VERSION            PIC X(10).
      02  CS-MAIN-MEMORY-SIZE       PIC S9(9) CMP.
      02  CS-GISMO-NAME             PIC X(30).
      02  CS-MMCP-NAME              PIC X(30).
      02  CS-NEW-LOG-FLAG           PIC S9 CMP.
    
```

Put on Release tape? (Roy)

Schedule

01	SCHEDULE-RECORD.	
02	SR-REC-TYPE	PIC X.
02	SR-JOB-ACCOUNTING-NUMBER	PIC S9(7) CMP.
02	SR-JOB-NUMBER	PIC S9(5) CMP.
02	SR-YEAR	PIC S9(3) CMP.
02	SR-JULIAN-DAY	PIC S9(3) CMP.
02	SR-TIME	PIC S9(7) CMP.
02	SR-PROGRAM-NAME	PIC X(30).
02	SR-OBJECT-PROGRAM-NAME	PIC X(30).
02	SR-SCHEDULE-PRIORITY	PIC S9(3) CMP.
02	SR-EXECUTE-TYPE	PIC S9 CMP.
02	SR-STATIC-MEMORY	PIC S9(9) CMP.
02	SR-DYNAMIC-MEMORY	PIC S9(9) CMP.
02	SR-TOTAL-MEMORY	PIC S9(9) CMP.
02	SR-CHARGE-NUMBER	PIC S9(7) CMP.
02	SR-VIRTUAL-DISK	PIC S9(5) CMP.
02	SR-NUMBER-OF-FILES	PIC S9(3) CMP.
02	SR-INTERPRETER-NAME	PIC X(30).
02	SR-YEAR-COMPILED	PIC S9(3) CMP.
02	SR-JULIAN-DAY-COMPILED	PIC S9(3) CMP.
02	SR-TIME-COMPILED	PIC S9(7) CMP.
02	SR-PARENT-JOB-NUMBER	PIC S9(5) CMP.

Beginning-of-Job

01	BOJ-RECORD.	
02	BOJ-RECORD-TYPE	PIC X.
02	BOJ-JOB-ACCOUNTING-NUMBER	PIC S9(7) CMP.
02	BOJ-JOB-NUMBER	PIC S9(5) CMP.
02	BOJ-YEAR	PIC S9(3) CMP.
02	BOJ-JULIAN-DAY	PIC S9(3) CMP.
02	BOJ-TIME	PIC S9(7) CMP.
02	BOJ-PROGRAM-NAME	PIC X(30).
02	BOJ-OBJECT-PROGRAM-NAME	PIC X(30).
02	BOJ-EXECUTE-TYPE	PIC S9 CMP.
02	BOJ-PROCESSOR-PRIORITY	PIC S9(3) CMP.
02	BOJ-MEMORY-PRIORITY	PIC S9(3) CMP.
02	BOJ-CHARGE-NUMBER	PIC S9(7) CMP.
02	BOJ-VIRTUAL-DISK	PIC S9(5) CMP.
02	BOJ-INTERPRETER-NAME	PIC X(30).

File Open

01	OPEN-RECORD.	
02	OR-REC-TYPE	PIC X.
02	OR-JOB-ACCOUNTING-NUMBER	PIC S9(7) CMP.
02	OR-JOB-NUMBER	PIC S9(5) CMP.
02	OR-YEAR	PIC S9(3) CMP.
02	OR-JULIAN-DAY	PIC S9(3) CMP.
02	OR-TIME	PIC S9(7) CMP.
02	OR-FILE-NUMBER	PIC S9(3) CMP.
02	OR-INTERNAL-NAME	PIC X(10).
02	OR-EXTERNAL-NAME	PIC X(30).
02	OR-HARDWARE-TYPE	PIC S9(3) CMP.
02	OR-VARIABLE-RECORDS	PIC S9 CMP.
02	OR-PSEUDO-READER-FILE	PIC S9 CMP.
02	OR-DMS-FILE	PIC S9 CMP.
02	OR-EMULATOR-TAPE-FILE	PIC S9 CMP.
02	OR-RECORD-SIZE	PIC S9(5) CMP.
02	OR-BLOCKING-FACTOR	PIC S9(5) CMP.
02	OR-MAXIMUM-BLOCK-SIZE	PIC S9(5) CMP.
02	OR-UNIT-NAME	PIC X(6).
02	OR-NUMBER-OF-STATIONS	PIC S9(3) CMP.
02	OR-ACCESS-MODE	PIC S9(3) CMP.
02	OR-NUMBER-AREAS-DECLARED	PIC S9(3) CMP.
02	OR-BLOCKS-PER-AREA	PIC S9(7) CMP.
02	OR-AUTOPRINT	PIC S9 CMP.
02	FILLER	PIC X(11).
02	OR-FILE-TYPE	PIC S9(3) CMP.
02	OR-SERIAL-NUMBER	PIC X(6).
02	OR-QUEUE-FAMILY-SIZE	PIC S9(3) CMP.
02	OR-QUEUE-MAX-MESSAGES	PIC S9(3) CMP.
02	OR-PROTECTION	PIC S9 CMP.
02	OR-PROTECTION-IO	PIC S9 CMP.
02	OR-NUMBER-OF-BUFFERS	PIC S9(3) CMP.
02	OR-MEMORY-REQUIRED	PIC S9(7) CMP.
02	OR-OPEN-INPUT	PIC S9 CMP.
02	OR-OPEN-OUTPUT	PIC S9 CMP.
02	OR-NEW-FILE	PIC S9 CMP.
02	OR-OPEN-WITH-PUNCH	PIC S9 CMP.
02	OR-OPEN-WITH-PRINT	PIC S9 CMP.
02	OR-OPEN-WITH-NO-REWIND	PIC S9 CMP.
02	OR-OPEN-REVERSE	PIC S9 CMP.
02	OR-OPEN-LOCK	PIC S9 CMP.
02	OR-OPEN-LOCKOUT	PIC S9 CMP.
02	FILLER	PIC X(3).
02	OR-OPEN-WITH-INTERPRET	PIC S9 CMP.
02	OR-OPEN-WITH-STACKERS	PIC S9 CMP.
02	OR-OPEN-INSECURE	PIC S9 CMP.
02	OR-OPEN-ON-BEHALF-OF	PIC S9 CMP.
02	OR-OPEN-EXTEND	PIC S9 CMP.
02	FILLER	PIC X(19).

File Close

01	CLOSE-RECORD.	
02	CR-REC-TYPE	PIC X.
02	CR-JOB-ACCOUNTING-NUMBER	PIC S9(7) CMP.
02	CR-JOB-NUMBER	PIC S9(5) CMP.
02	CR-YEAR	PIC S9(3) CMP.
02	CR-JULIAN-DAY	PIC S9(3) CMP.
02	CR-TIME	PIC S9(7) CMP.
02	CR-FILE-NUMBER	PIC S9(3) CMP.
02	CR-INTERNAL-NAME	PIC X(10).
02	CR-EXTERNAL-NAME	PIC X(30).
02	CR-HARDWARE-TYPE	PIC S9(3) CMP.
02	CR-VARIABLE-RECORDS	PIC S9 CMP.
02	CR-PSEUDO-READER-FILE	PIC S9 CMP.
02	CR-DMS-FILE	PIC S9 CMP.
02	CR-EMULATOR-TAPE-FILE	PIC S9 CMP.
02	CR-RECORD-SIZE	PIC S9(5) CMP.
02	CR-BLOCKING-FACTOR	PIC S9(5) CMP.
02	CR-MAXIMUM-BLOCK-SIZE	PIC S9(5) CMP.
02	CR-UNIT-NAME	PIC X(6).
02	CR-NUMBER-OF-STATIONS	PIC S9(3) CMP.
02	CR-ACCESS-MODE	PIC S9(3) CMP.
02	CR-NUMBER-AREAS-DECLARED	PIC S9(3) CMP.
02	CR-BLOCKS-PER-AREA	PIC S9(7) CMP.
02	CR-AUTO-PRINT	PIC S9 CMP.
02	CR-RECORD-COUNT	PIC S9(7) CMP.
02	CR-BLOCK-COUNT	PIC S9(7) CMP.
02	CR-ERROR-COUNT	PIC S9(5) CMP.
02	CR-FILE-TYPE	PIC S9(3) CMP.
02	CR-SERIAL-NUMBER	PIC X(6).
02	CR-QUEUE-FAMILY-SIZE	PIC S9(3) CMP.
02	CR-QUEUE-MAX-MESSAGES	PIC S9(3) CMP.
02	CR-PROTECTION	PIC S9 CMP.
02	CR-PROTECTION-IO	PIC S9 CMP.
02	CR-NUMBER-OF-BUFFERS	PIC S9(3) CMP.
02	CR-MEMORY-REQUIRED	PIC S9(7) CMP.
02	CR-CLOSE-REEL	PIC S9 CMP.
02	CR-CLOSE-RELEASE	PIC S9 CMP.
02	CR-CLOSE-PURGE	PIC S9 CMP.
02	CR-CLOSE-REMOVE	PIC S9 CMP.
02	CR-CLOSE-WITH-CRUNCH	PIC S9 CMP.
02	CR-CLOSE-WITH-NO-REWIND	PIC S9 CMP.
02	CR-CLOSE-INSECURE	PIC S9 CMP.
02	CR-CLOSE-WITH-LOCK	PIC S9 CMP.

02	CR-CLOSE-CONDITIONAL	PIC S9 CMP.
02	CR-CLOSE-ROLLOUT	PIC S9 CMP.
02	CR-DMS-AUDIT-SWITCH	PIC S9 CMP.
02	FILLER	PIC X(1).
02	CR-CLOSE-UNAVAIL	PIC S9 CMP.
02	FILLER	PIC X(18).
02	CR-CLOSE-REPORT-ERRORS-ON-OUTPUT	PIC S9 CMP.
02	CR-CLOSE-FOR-SORT	PIC S9 CMP.
02	CR-CLOSE-TERMINATE	PIC S9 CMP.
02	CR-CLOSE-BACKUP-EXCEEDED-FILESIZE	PIC S9 CMP.
02	CR-CLOSE-SUPPORT	PIC S9 CMP.

DMS Statistics

01	DMS-STATISTICS-RECORD.	
02	DMS-RECORD-TYPE	PIC X.
02	DMS-JOB-ACCOUNTING-NUMBER	PIC S9(7) CMP.
02	DMS-JOB-NUMBER	PIC S9(5) CMP.
02	DMS-YEAR	PIC S9(3) CMP.
02	DMS-JULIAN-DAY	PIC S9(3) CMP.
02	DMS-TIME	PIC S9(7) CMP.
02	DMS-DATABASE-NAME	PIC X(10).
02	DMS-STRUCTURE-NUMBER	PIC S9(5) CMP.
02	DMS-INVOKE-NUMBER	PIC S9(3) CMP.
02	DMS-STRUCTURE-CLOSE	PIC S9 CMP.
02	DMS-STRUCTURE-NAME	PIC X(17).
02	DMS-FILE-NAME	PIC X(30).
02	DMS-HARDWARE-TYPE	PIC S9(3) CMP.
02	DMS-UNIT-NAME	PIC X(6).
02	DMS-NUMBER-RANDOM-FINDS	PIC S9(9) CMP.
02	DMS-NUMBER-SEQUENTIAL-FINDS	PIC S9(9) CMP.
02	DMS-NUMBER-OF-INSERTS	PIC S9(9) CMP.
02	DMS-NUMBER-OF-UPDATES	PIC S9(9) CMP.
02	DMS-NUMBER-OF-DELETES	PIC S9(9) CMP.
02	DMS-NO-OF-SYSTEM-CHANGES	PIC S9(9) CMP.
02	DMS-NUMBER-OF-EXCEPTIONS	PIC S9(9) CMP.
02	DMS-NUMBER-LOGICAL-READS	PIC S9(9) CMP.
02	DMS-NUMBER-LOGICAL-WRITES	PIC S9(9) CMP.
02	DMS-NUMBER-PHYSICAL-READS	PIC S9(9) CMP.
02	DMS-NUMBER-PHYSICAL-WRITES	PIC S9(9) CMP.

DMS Open/Close

01	DMS-OPEN-CLOSE-RECORD.	
02	DMSOC-RECORD-TYPE	PIC X.
02	DMSOC-JOB-ACCOUNTING-NUMBER	PIC S9(7) CMP.
02	DMSOCA-JOB-NUMBER	PIC S9(5) CMP.
02	DMSOC-YEAR	PIC S9(3) CMP.
02	DMSOC-JULIAN-DAY	PIC S9(3) CMP.
02	DMSOC-TIME	PIC S9(7) CMP.
02	DMSOC-PHYSICAL-DATABASE-NAME	PIC X(10).
02	DMSOC-LOGICAL-DATABASE-NAME	PIC X(17).
02	DMSOC-CLOSE-FLAG	PIC S9 CMP.
02	DMSOC-UPDATE-FLAG	PIC S9 CMP.
02	DMSOC-AUDIT-FLAG	PIC S9 CMP.
02	DMSOC-HARDWARE-TYPE	PIC S9(3) CMP.
02	DMSOC-UNIT-NAME	PIC X(6).
02	DMSOC-PACK-NAME	PIC X(10).
02	DMSOC-NUMBER-OF-UPDATE-OPS	PIC S9(9) CMP.
02	DMSOC-NUMBER-NON-UPDATE-OPS	PIC S9(9) CMP.
02	DMSOC-DMCP-TIME-TENTHS	PIC S9(9) CMP.
02	DMSOC-EXCEPTION-COUNT	PIC S9(9) CMP.
02	DMSOC-TRANSACTION-COUNT	PIC S9(9) CMP.
02	DMSOC-TRANS-STATE-TIME	PIC S9(9) CMP.
02	DMSOC-CONTENTION-COUNT	PIC S9(9) CMP.
02	DMSOC-CONTENTION-WAIT-TIME	PIC S9(9) CMP.
02	DMSOC-IO-WAIT-COUNT	PIC S9(9) CMP.
02	DMSOC-IO-WAIT-TIME	PIC S9(9) CMP.

Peripheral Assignment/Release

01	PERIPHERAL-ASSIGN-RELEASE-RECORD.	
02	PAR-RECORD-TYPE	PIC X.
02	PAR-JOB-ACCOUNTING-NUMBER	PIC S9(7) CMP.
02	PAR-JOB-NUMBER	PIC S9(5) CMP.
02	PAR-YEAR	PIC S9(3) CMP.
02	PAR-JULIAN-DAY	PIC S9(3) CMP.
02	PAR-TIME-TENTHS	PIC S9(7) CMP.
02	PAR-HARDWARE-TYPE	PIC S9(3) CMP.
02	PAR-UNIT-NAME	PIC X(6).
02	PAR-ACTION	PIC S9 CMP.

End-of-Job

01	EOJ-RECORD.	
02	EOJ-RECORD-TYPE	PIC X.
02	EOJ-JOB-ACCOUNTING-NUMBER	PIC S9(7) CMP.
02	EOJ-JOB-NUMBER	PIC S9(5) CMP.
02	EOJ-YEAR	PIC S9(3) CMP.
02	EOJ-JULIAN-DAY	PIC S9(3) CMP.
02	EOJ-TIME-TENTHS	PIC S9(7) CMP.
02	EOJ-PROGRAM-NAME	PIC X(30).
02	EOJ-OBJECT-PROGRAM-NAME	PIC X(30).
02	EOJ-EXECUTE-TYPE	PIC S9 CMP.
02	EOJ-END-OF-JOB-REASON	PIC S9 CMP.
02	EOJ-CPU-TIME-TENTHS	PIC S9(7) CMP.
02	EOJ-NUMBER-OF-CODE-OVERLAYS	PIC S9(7) CMP.
02	EOJ-NUMBER-OF-DATA-OVERLAYS	PIC S9(7) CMP.
02	EOJ-SUCCESSOR-PROGRAM	PIC X(30).
02	EOJ-DS-OR-DP-MESSAGE	PIC X(60).

Log Comment

01	LOG-COMMENT-RECORD.	
02	LC-RECORD-TYPE	PIC X.
02	FILLER	PIC X(7).
02	LC-YEAR	PIC S9(3) CMP.
02	LC-JULIAN-DAY	PIC S9(3) CMP.
02	LC-TIME-TENTHS	PIC S9(7) CMP.
02	LC-COMMENT-LENGTH	PIC S9(3) CMP.
02	LC-COMMENT-TEXT	PIC X(162).

Log Transfer

01	LOG-TRANSFER-RECORD.	
02	LT-RECORD-TYPE	PIC X.
02	FILLER	PIC X(7).
02	LT-YEAR	PIC S9(3) CMP.
02	LT-JULIAN-DAY	PIC S9(3) CMP.
02	LT-TIME-TENTHS	PIC S9(7) CMP.
02	LT-NEXT-LOGFILE-NAME	PIC X(30).

DMS Statistics

	NS 06 1 CF	
		1 1 DRTYPE
		P 2 50DJAN
		P 6 80DJOBNO
		P 9 100DYEAR
		P 11 120DJDAY
		P 13 160DTIME
		17 26 DDBNAM
		P 27 280DSTRNO
		P 29 300DINVNO
		P 31 310DSTRCL
		32 48 DSTRNM
		49 78 DFILNM
		P 79 800DHDWR
		81 86 DUNI
		P 87 910DRFIND
		P 92 960DSFIND
		P 97 1010DINSTR
		P 102 1060DUPDAT
		P 107 1110DDELET
		P 112 1160DSYSCH
		P 117 1210DEXCEP
		P 122 1260DLREAD
		P 127 1310DLWRIT
		P 132 1360DPREAD
		P 137 1410DPWRIT
	0---1---*---2---*---3---*---4---*---5---*---6---*---7---	
	6 0 0 0 0 0 0 0 0 0 0	

DMS Open/Close

	NS 07 1 CG	
		1 1 MRTYPE
		P 2 50MJAN
		P 6 80MJOBNO
		P 9 100MYEAR
		P 11 120MJDAY
		P 13 160MTIME
		17 26 MPHSNM
		27 43 MLOGNM
		P 44 440MCLOSE
		P 45 450MUPDAT
		P 46 460MAUDIT
		P 47 480MHDWR
		49 54 MUNI
		55 64 MPID
		P 65 690MUPOPS
		P 70 740MPNONUP
		P 75 790MPMCPTI
		P 80 840MEXCPCT
		P 85 890MTRANCT
		P 90 940MTRSTTI
		P 95 990MCONTCT
		P 100 1040MCNTWTI
		P 105 1090MIOWCT
		P 110 1140MIOWTI
	0---1---*---2---*---3---*---4---*---5---*---6---*---7---	
	6 0 0 0 0 0 0 0 0 0 0	

SECTION 20

PACK/INIT

The PACK/INIT program is a stand-alone cassette utility program that labels, initializes, verifies, or reconfigures disk packs (but not cartridges) for use on B 1000 systems. The PACK/INIT program also relocates bad sectors on the B 9494-5, B 9494-10, and B 9484-12 disks without the loss of data. The resulting pack types are System (S) or User (U). The PACK/INIT program is a stand-alone program and requires exclusive use of the system. This program is very similar to the SYSTEM/DISK-INIT system utility program, which is an on-line, normal-state program that does not require the exclusive use of the entire system. The SYSTEM/DISK-INIT program also offers more extensive debugging capabilities.

All disks must be initialized before they can be used with the B 1000 System Software. Fixed disks such as the B 9499-6, B 9494-5, and B 9494-10 are initialized during the manufacturing process. A B 9499-6 disk should not need more than a cylinder zero initialization on site. The B 9494-5 and B 9494-10 disk need only an "FC" Factory Convert. Whenever a factory initialized disk must be initialized on site, the "CI" Cylinder Initialize or "RI" Re-initialize actions should be used, if possible, to preserve the factory determined relocation of bad sectors.

During initialization, the PACK/INIT program formats each sector by writing addresses, sync bits, data patterns, and error protection codes, and then reads the addresses and data to check for errors.

If five or fewer sectors in any cylinder are bad, they are relocated to one of the 5 spare sectors in that cylinder. If more than five sectors in a cylinder are bad and the REMOVE option is set, the first five are relocated to the spare sectors, and the remainder are removed (by the equivalent of XD) from the Master Available Table on the disk. The REMOVE option is set by default for B 9499-6, B 9494-5, and B 9494-10 disks and for the RC, RI and CI actions.

If more than five of the first 64 sectors in cylinder zero are defective, that disk cannot be used with the B 1000 System Software.

The program individually processes each cylinder of a disk to verify its integrity. The actual number of read and write operations performed on each cylinder is based on the characteristics of the disk device being processed. Cylinder zero, the outermost cylinder of the disk, is initialized first. The Master Available Table and the Relocate Table are constructed in cylinder zero to store information on relocated and removed sectors. A printed analysis of these tables can be obtained by executing the normal-state utility program INITIALIZE/ANALYZER. Label information and the remainder of the MCP structures are built into cylinder zero at the end of the initialization process.

Disk device characteristics and initialization characteristics are documented in appendix A of this volume.

PROGRAM LOADING INSTRUCTIONS

Two sets of procedures for program loading follow. The first is for B 1990 systems and the second is for all other B 1000 systems.

Program Loading on B 1990 Systems

To load the program on B 1990 systems:

1. Halt the system by entering the HALT system command. For dual-processor systems be sure both processors are halted.

NOTE

Instructions for using the B 1990 Soft Panel in this fashion are hereafter presented in either of the following forms:

Transmit <keyword-1> <keyword-2> ...

<keyword-1> <keyword-2> ... [XMT]

A looping condition could deny access to the ODT. In such cases, press the INTRPT push button. If the system still does not halt, press the HALT/RUN push button.

2. Place the cassette into the cassette reader and ensure that the tape rewinds to BOT.
3. Transmit CLEAR MTR GO.

The cassette automatically rewinds on B 1990 systems.

Program Loading on other B 1000 Systems

To load the program on all other B 1000 systems:

1. Halt the system by entering the HALT system command if the MCP is running. If the MCP is not running, halt the system by using the INTRPT push button or the console INTERRUPT switch. If the INTERRUPT switch is used, return it to its initial position following the halt. If the system continues to run, press the HALT push button. If the system still continues to run, press the HALT and CLEAR push buttons simultaneously. For dual-processor systems be sure both processors are halted.
2. Place the cassette into the cassette drive and ensure that the tape rewinds to BOT.
3. Press (1) the CLEAR push button, (2) the MODE push button to obtain the MTR mode, and (3) the START push button.
4. After the bootstrap loader is read from the cassette tape, the processor halts. The L register must contain @AAAAAA@ at this time; if not, the cassette must be rewound and the procedure restarted from step 3.
5. Press the START push button. Reading of the cassette resumes. When the program is completely loaded, program operation begins.
6. Rewind the cassette tape by pressing the REWIND button. (Rewind is automatic on B 1900 systems.)

NOTES

If the tape does not automatically rewind when it is inserted into the console cassette drive, press the REWIND button. If the tape is wound completely onto the right spool, remove the cassette and manually rewind it until the clear leader is no longer at the read gap.

If the system halts with @000004@ or @0D0053@ in the L register or the PERP register equal to 1, a cassette error during program loading is indicated. Restart the procedure from step 1.

PROGRAM OPERATING INSTRUCTIONS

After the program is loaded from the cassette, the following messages are displayed on the Operator Display Terminal (ODT):

```
DISK PACK INITIALIZER <mark-and-patch-level> (<compile-date/time>) -- PLEASE  
PUSH SPCFY IF YOU SEE AN ETX ON THIS SCREEN--
```

If an ETX character appears on the screen, press the SPCFY key once. This is required because of the variety of ODT device types and firmware settings available on B 1000 systems.

If the system contains a card reader control, the following message is also displayed:

```
IS CARD INPUT DESIRED? <YES OR NO>
```

Initialization specifications are then entered, either through the ODT or through a card reader. Several disks may be processed consecutively during one program execution. When input specifications are entered through a card reader, a set of input specifications is entered for each disk to be processed. The program continues to read the specifications until it reaches the end of file (a blank card) when card input is used.

ODT Operation

Under ODT operation, the program prompts the operator for all disk initialization specifications.

At the beginning of the program, and after processing each disk, the program begins the prompt sequence again with the following message:

ENTER UNIT MNEMONIC <DP? OR BLANK FOR EOJ>

A blank input response terminates the program with the message:

END OF INITIALIZER

Card Reader Operation

When the response to the IS CARD INPUT DESIRED prompt is YES, the program accepts the disk initialization specifications from cards. The initialization specifications are not contained in a data deck; therefore, neither a ?DATA nor a ?END card is used.

The user interface information is in free-form format (up to 96 columns) and the specifications are separated by one or more spaces.

The specification card format follows the same order as the prompt sequence. The syntax is described below. The semantics of the commands are described in the User Interface Information subsection.

Syntax (Initialization):

— <drive> I <sn> <id> <type> _____

 _____ <date> _____
 _____ <name> _____

Syntax (Verification):

— <drive> V _____

Syntax (Reconfiguration):

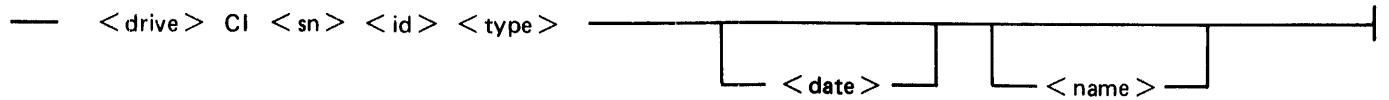
— <drive> R <type> _____

Syntax (Reinitialization):

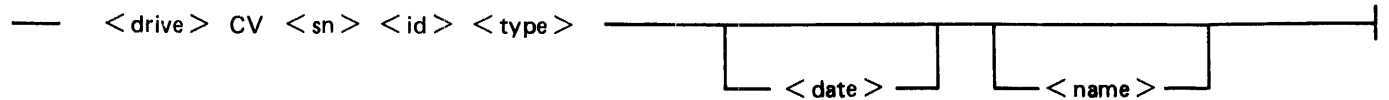
— <drive> RI <sn> <id> <type> _____

 _____ <date> _____
 _____ <name> _____

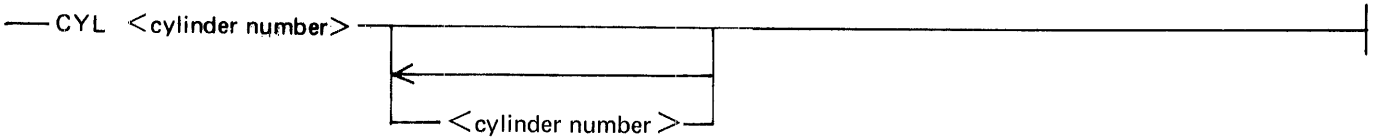
Syntax (Cylinder Reinitialization):



Syntax (Cylinder Verification):



When cylinder initialization (CI) or cylinder verification (CV) is specified, an additional card is used to specify the cylinders to be processed. The syntax is described below.



The following conventions are used in the preceding syntax diagrams:

- <drive> is the unit mnemonic.
- <sn> is the serial number.
- <id> is the pack id.
- <type> is the pack type.
- <date> is the Julian date.
- <name> is the owner's name.
- <cylinder number> is the number of the cylinder to be processed.

The initialization options used when the program is executed from cards are identical to those used when it is executed from the ODT. They are described in the Initialization Options subsection.

STATUS Command

While the program is executing, under either ODT or card control, the operator can interrogate the program status with a STATUS or ST command. In response, the program displays a message that indicates the cylinder that is the next to be processed and the number of sectors that have been relocated or removed thus far. An example of this message follows:

CYL 302 of 814 (37%): 8 BAD SECTORS (7 RELOCATED, 1 REMOVED)

INITIALIZATION SPECIFICATIONS

The following paragraphs are the initialization specifications.

User Interface Information

Drive

Drive specifies a unit on which the disk to be processed is located; for example, DPC.

Prompt: ENTER UNIT MNEMONIC <DP? OR BLANK FOR EOJ>

Action

Action specifies the function to be performed on the drive. When an I, RI, RC, or V action is requested, the program processes the entire disk in sequential cylinder order, beginning with cylinder zero. When the requested action is CI or CV, the program issues prompts during processing that ask the operator to specify the cylinders to be processed. When an RR action is requested, the program prompts for each single sector to be relocated or reallocated.

Prompt: ENTER ACTION: <RR, I, V, RC, RI, CI, OR CV>

The possible actions are:

I = Initialization

Initialize and verify the entire disk using the default initialization characteristics documented in appendix A of this volume.

V = Verification only

Check that all sectors can be found and the data read with good parity. Only one pass is made on the disk and the reads are not stressed (no head offset or threshold changes).

RC = Reconfiguration

Purge the disk, verify it, and change the pack type to that specified. No cylinder initialization is performed unless a new bad sector is found on a cylinder. The affected cylinder is initialized, any sectors that had previously been relocated are relocated again, and if there are no more spares available on the cylinder, the new bad sector is relocated or removed.

RI = Reinitialization

Reinitialization responds the same as an I action, with the addition that all sectors found to be bad on the previous initialization are automatically considered bad on this initialization. This action is highly recommended for fixed disks, such as the B 9499-6, B 9494-5, and B 9494-10 that have been previously initialized at the factory.

CI = Cylinder Initialization

Initialize single cylinders or a range of cylinders as specified in answer to the following CYLINDER NUMBER prompts. Any sectors that had previously been relocated or removed are automatically relocated or removed again. This action may also be used to restart a previously aborted initialization within 50 cylinders of where it was aborted.

If the disk has a bad label, Master Available Table, or Working Available Table, or the operator specifies that the disk is to be purged, then cylinder zero is automatically initialized, thus purging the disk.

If all of the necessary structures on the disk are found to be valid and the previous initialization did not abort before it initialized the entire disk, the following prompt is displayed on the ODT:

IS A PURGE DESIRED? <YES OR NO>

A YES response causes the PACK/INIT program to prompt for SERIAL NUMBER, PACK ID, PACK TYPE, OWNER'S NAME and OPTIONS. The PACK/INIT program then does an initialization of cylinder zero, thus rendering inaccessible all data previously on the disk. Any further initialization is performed in response to CYLINDER NUMBER prompts. A NO response results in the initialization of cylinders specified in response to CYLINDER NUMBER prompts. The disk is not purged, and the program does not initialize any cylinder or range of cylinders that is in use (not entirely available).

CV = Cylinder Verification

Verify only those cylinders specified in answer to the CYLINDER NUMBER prompts, except cylinder zero, which is automatically verified.

RR = Relocate and Reallocate

This action is only offered for B 9494-5, B 9494-10, and B 9484-12 disks. It is used to relocate or reallocate (the reverse or undoing of a relocate) a single sector at a time. The sector may be in use and its data is retained, whenever possible. When this action is requested, the following prompt sequence ensues:

Prompt: A SECTOR CAN BE RELOCATED TO A
SPARE OR
REALLOCATED FROM A SPARE
ENTER <SECTOR ADDRESS> TO
RELOCATE, " - "
<SECTOR ADDRESS> TO
REALLOCATE
WHICH SECTOR (BLANK TO QUIT)?

Response examples:

```
1000
@0003E8@
- 1000
- @0003E8@
```

The first two examples cause sector 1000 to be relocated; the next two cause it to be reallocated, if it is found to be relocated. Entering a blank input terminates the RR action.

FC = Factory Convert

This action is only offered for the B 9494-5 and B 9494-10 fixed disk that is in the format delivered by the factory. It converts the factory Interchange format to the B 1000 native mode format and is used only once when the disk is first installed. The prompt sequence asks for the PACK TYPE and OWNER'S NAME.

NOTE

The actions RR, RC, RI, CI, and CV are valid only when processing a disk that has previously been initialized with the Mark 8.0 or later version of the SYSTEM/DISK-INIT or PACK/INIT programs and the disk label, Master Available Tables, and Relocate Table are intact and valid.

SERIAL NUMBER

The SERIAL NUMBER message requests a six-digit decimal number to be used on a new or purged pack to identify the disk. This number must be non-zero and should be unique to avoid confusion. Note that the MCP does not allow two User disks with the same serial number to be on-line at the same time.

Prompt: ENTER 6 DIGIT SERIAL NUMBER

PACK ID

The PACK ID message requests a name that is assigned to the disk, using 10 characters or less and containing no embedded blank characters. The system uses this name to reference the disk. When input specifications are entered from cards, omitting this entry causes an error for pack-type. If the disk is a User disk, the reserved name DISK is not allowed.

Prompt: ENTER PACK ID

PACK TYPE

The PACK TYPE message requests the type of use this disk will be dedicated to until the next initialization. The type is specified by a one-letter abbreviation of the allowable pack types, which are User (U) or System (S).

Prompt: ENTER PACK TYPE - <U OR S>

DATE

The DATE message requests the current date in Julian date format. A Julian date is a five-digit field, with two digits for the year followed by three digits for the absolute day of the year, for example; 84256.

Prompt: ENTER 5 DIGIT JULIAN DATE (YYDDD)

OWNER'S NAME

The OWNER'S NAME message requests an optional 14-character field available to identify the person responsible for maintaining the disk. The system does not reference this field.

Prompt: ENTER OWNER'S NAME

OPTIONS

The OPTIONS message requests any of several options to be entered. These options are available during initialization of a disk to provide a more flexible initialization and verification procedure. The options allow the operator to extend or enhance the default initialization characteristics of the disk.

The OPTION prompt is repeated until a blank input message is received, at which point the previously requested action begins.

Prompt: ENTER OPTIONS

NOTE

The options are described in a following paragraph, Initialization Options.

CYLINDER NUMBER

The CYLINDER NUMBER message asks for the number of the cylinder or for a range of cylinders to be processed. This information is required only when either the Cylinder Initialization (CI) or Cylinder Verification (CV) action is specified. The PACK/INIT program sequentially processes the specified cylinders and reissues this prompt until it receives a blank or null input response. A single cylinder (indicated by one number) or a range of cylinders (indicated by number-space-number or by number-space-dash-space-number) may be specified. The following example illustrates this prompt and valid responses.

Prompt: ENTER CYLINDER NUMBER
Responses: 25 40
 25 - 40
 300

When card input is used, the cylinder number specification must be on a separate card from the input specifications, and the keyword CYL must appear in columns 1-3, followed by the cylinder numbers to be processed. Several CYL cards can be used if needed.

Initialization Integrity Protection

When the CI action is requested, the PACK/INIT program prevents a disk from being only partially initialized by checking the Relocate Table maintained on the disk for the initialization status. If the table indicates that the previous initialize did not finish, the program automatically completes the initialization before processing any operator-specified cylinders. The ENTER CYLINDER NUMBER prompt is not issued until the program ensures that the disk is completely initialized.

Initialization Options

The operator can extend or enhance the default initialization characteristics for a disk by using the initialization options. The REMOVE and LIMIT options control the total number of bad sectors that are handled for a single cylinder and for an entire disk before the disk becomes a rejection candidate. The Marginal Sectors option specifies sectors that are known to be bad and requests that they be automatically relocated or removed. The Dollar Sign options control the physical I/O operations that occur when processing a disk.

For ODT operation, the options are entered after the ENTER OPTIONS prompt appears.

For card reader operation, each option must appear on a separate card from all other specifications.

REMOVE Option

The REMOVE option allows the default errors-per-cylinder initialization error limit for disk packs (see appendix A) to be overridden. It specifies that after the first five bad sectors in the same cylinder have been relocated to a spare sector, the remaining bad sectors are to be removed (XD system command) from the Master Available Table.

The default errors-per-pack limit might also cause a pack to be rejected unless the LIMIT option has also been specified.

The REMOVE option is set by default when processing B 9499-6, B 9494-5, B 9494-10, and B 9484-12 disks and when a RC, RI, or CI action is requested.

The REMOVE option is set with the following syntax:

```
REMOVE
```

LIMIT Option

The LIMIT option allows the default errors-per-pack initialization error limit for disk packs (see appendix A) to be overridden. It specifies that a disk is not to be rejected until the errors-per-pack count exceeds the value contained in the specification.

For B 9484-5 and B 9499-8 disk packs, the default errors-per-cylinder limit of five might also cause a pack to be rejected unless the REMOVE option has also been requested.

The LIMIT option is specified with the following syntax:

```
LIMIT <integer>
```

Marginal Sectors Option

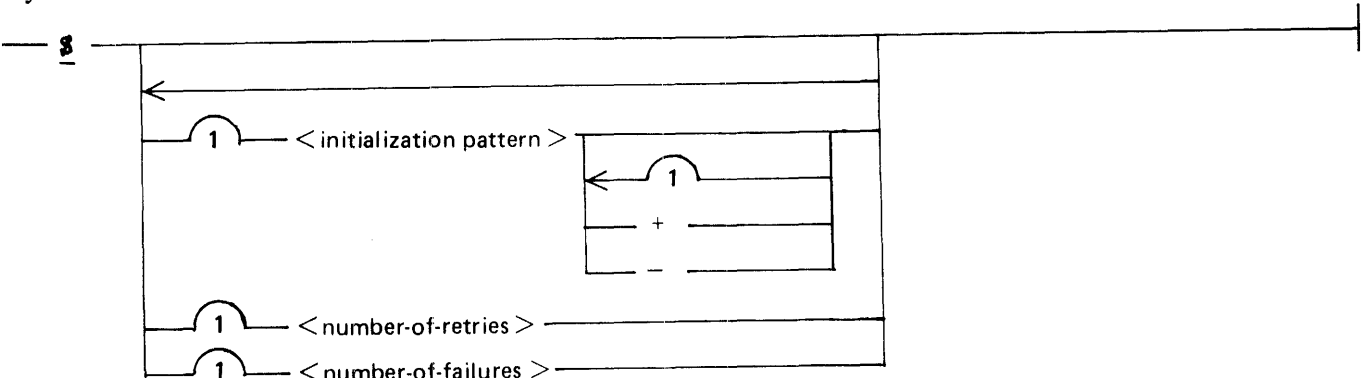
This option removes or relocates sectors that might pass the disk integrity verification check, but still be unreliable in normal use. Marginal sectors that are removed or relocated are included in the error count totals. Sector addresses specified in this option must be valid decimal addresses. More than one sector address can appear in each response to the ENTER OPTIONS prompt, or on each input specification card, with the restriction that the addresses are separated by spaces, not by commas.

If CI or CV is requested as the action (Cylinder Initialization or Cylinder Verification), then the operator must ensure that the specified marginal sector resides within a cylinder that is to be processed (has been included in a CYLINDER NUMBER input specification). Note that the first cylinder is cylinder zero, not cylinder one.

Dollar Sign Options

These options allow the operator to verify with or without offset, to compare during verification, to change the number of retries on bad sectors, or to modify the number of failures that can occur before a sector is considered bad.

Syntax:



Semantics:

Initialization Pattern

The initialization pattern entry is a four-digit representation of a hexadecimal pattern (0000 through FFFF). The first dollar-sign statement in a series of dollar-sign statements that omits this entry causes the default pattern to be used for that pass (refer to the disk device characteristics for the default patterns).

Offset: + or -

The term "offset" refers to the ability to modify the alignment of a disk pack read head to test marginal disk conditions. The direction of offset is specified by the sign. For all disk pack units except the B 9499-6, plus (+) causes the head to be moved toward the inner side of the track (closer to disk center), and minus (-) causes the head to be moved toward the outer side of the track (away from disk center). For B 9499-6 units, plus (+) results in a higher threshold and minus (-) results in a lower threshold. The default is no offset or normal threshold.

Offset is used in the alignment process of the disk recording heads. It is only recommended for use by Field Engineers who need to realign the disk recording heads. This capability is not available in Design Level I B 9499-7 or the B 9499-8 Disk Pack Drives or the Design Level I DPEC. Specifying offset on these disk packs has no effect on initialization.

Number-of-retries

The number-of-retries entry specifies the number of times during verification that the same I/O operation is attempted after an exception condition has occurred. Default is ten retries. This specification is not valid for B 9494-5, B 9494-10, or B 9484-12 type disks.

Number-of-failures

The number-of-failures entry specifies how many of the retries specified by the number-of-retries entry are allowed to fail before the sector is considered bad and is relocated or removed. Default allows one failure. This specification is not valid for B 9494-5, B 9494-10, or B 9484-12 disks.

NOTE

If the default values for both number-of-retries and number-of-failures are used while verifying a pack, all ten retries must be successful to allow the sector to remain.

INFORMATIVE MESSAGES

The PACK/INIT program issues messages that describe the processing that is being performed. The following is an example of the information in these informative messages when cylinder initialization (CI) is requested.

```
[ INITIALIZE BEGINS          - CYL 0 - DPD
  INITIALIZE COMPLETED     - CYL 0 - DPD
  INITIALIZE BEGINS          - CYL 96 - DPD
```

RELOCATED

```
THE FOLLOWING SECTORS ARE IN ERROR AND WILL BE - DPD
SECTOR @01F914@ = 129300
SECTOR @0203AD@ = 132013
SECTOR @021A31@ = 137777
```

```
INITIALIZE COMPLETED - CYL 103 - DPD
DPD(659): ID = "BIGGIE", SER. # 120340, TYPE = U
3 BAD SECTORS RELOCATED
INITIALIZE COMPLETED - DPD
```

The following sequence of messages exemplifies the informative messages issued by the PACK/INIT program when the requested action is an INITIALIZE (I):

```
INITIALIZE BEGINS - CYL 0 -- DPC
```

```
DPC(206): ID = "LADN", SER. # 091184, TYPE = S
0 BAD SECTORS
INITIALIZE COMPLETED - DPC .td
```

ok IN SOURCE

ERROR MESSAGES

NOTE

If a B 1990 system halts with an irrecoverable exception, the following error messages are displayed on page 2 of the ODT and the Maintenance Card switches the ODT display to page 1 and allows command line input. The operator should look to page 2 for the program error message and, for those halts that specify "press the START push button," transmit GO on the command line of page 1. This is equivalent to pushing the START push button on other B 1000 systems.

INVALID RESPONSE <response>

The prompt response is invalid; enter a valid response.

INVALID SERIAL NUMBER <serial-number>

Enter a valid serial number. It must be a six-digit, non-zero decimal number.

INVALID UNIT MNEMONIC <unit-mnemonic>

The unit mnemonic is invalid; enter a valid unit mnemonic (DPx).

INVALID SECTOR NUMBER <sector-number>

The addresses of marginal sectors must be specified in decimal form.

INVALID CYLINDER NUMBER <cylinder-number>

The cylinder number requested is beyond the capacity of the attached disk. Try again with a valid cylinder number.

THE FOLLOWING SECTORS ARE IN ERROR AND WILL BE RELOCATED

This informative message heading is issued when the first bad sector on a disk is encountered during an initialization. The addresses of the bad sectors follow the message.

DISK ERROR (10 TRIES) <disk-address> -

RESULT = <disk=io=result=descriptor>

An irrecoverable read/write error still exists after 10 retries. The operator is offered the option of retrying the read/write operation another 10 times.

THE ENTIRE PROGRAM MUST BE RELOADED TO CONTINUE

The PACK/INIT program cannot continue execution. Reload the cassette to restart the program.

THE PACK ON <unit-mnemonic> CANNOT BE USED WITH THE MCP

The initialization of the current disk was aborted before the entire disk was initialized. The disk is not usable by the MCP because it does not have a label or directory.

SECTOR <disk-address> REMOVAL REQUIRED - CANNOT REMOVE ANY OF THE 1ST 64 SECTORS

All of the first 64 sectors of a disk must be available or the disk cannot be used on a B 1000 system. The disk has been rejected, and the program will terminate following this message.

REMOVED <disk-address>

This is an informative message that specifies that the sector has been removed (XDed) from the Master Available Table because no more spares were available on its associated cylinder.

ERROR LIMITS HAVE BEEN EXCEEDED – <unit-mnemonic>

The maximum number of bad sectors allowed for the currently attached disk (either the default value or the value specified by the LIMIT option) has been exceeded. Therefore, the disk has been rejected and the program has terminated.

WOULD YOU LIKE TO RETRY? <YES OR NO>

This message follows various other error messages. It offers the operator the option of retrying a read/write operation that has failed. Depending on the reason for the failure, it may be necessary for the operator to take some action to correct the failure, such as removing a WRITE LOCKOUT condition from the disk drive, before allowing the program to continue. A YES response causes the read/write operation to be retried. A NO response terminates the program.

WRITE LOCK OUT <unit-mnemonic>

Reset the WRITE LOCKOUT status (turn on WRITE ENABLE) on the specified disk unit and respond to any following ODT prompts.

DISK NOT READY <unit-mnemonic>

Ready the specified disk unit and respond to any following ODT prompts.

MORE THAN <remove-limit> SECTORS HAVE BEEN REMOVED ON <unit-mnemonic>

MORE THAN <cylinder-transient-limit> TRANSIENT ERRORS HAVE BEEN ENCOUNTERED ON CYLINDER <cylinder-number>

MORE THAN <disk-transient-limit> TRANSIENT ERRORS HAVE BEEN ENCOUNTERED ON <unit-mnemonic>

Any of the preceding messages may be accompanied by the following message:

WARNING: <unit-mnemonic> DOES NOT MEET BURROUGHS STANDARDS FOR AN ACCEPTABLE PACK

DO YOU WANT TO IGNORE THIS WARNING AND CONTINUE?

<YES OR NO>

One of the specified error limits has been exceeded. This would normally result in the rejection of the disk; however, the operator will usually be given a chance to override these limits. A YES response causes the limit to be ignored; a NO response results in disk rejection and program termination. Use caution when ignoring limits, particularly in the case of transient errors, because of possibilities like the following: (1) A hardware problem may exist; that is, the failure may be associated with the disk control or the drive rather than the disk medium. (2) The disk may exhibit a high error and retry rate when used with the MCP. (3) A failing sector may not have failed enough times to be considered as a candidate for relocation, or a sector failing when read as one of a group of sectors may defy isolation to a single failing sector.

SECTION 21

QWIKLOG

The QWIKLOG program is a normal-state utility program designed to provide a convenient and compact analysis of job information contained within the MCP II log file. The MCP II log file is a disk file of system activity maintained by the MCP II.

The QWIKLOG program presents the results of its analysis in two distinct formats:

1. A tabular summary of selected information about each job.
2. A graph that depicts schedule and mix activity chronologically.

PRELIMINARY SETUP

The following MCP options must be set prior to using the QWIKLOG program.

LOG Option

The LOG option must be set for the MCP II to maintain the log information. (For information on setting the log, refer to the SL system command in section 2 of Volume 1 of the *B 1000 Systems System Software Operation Guide*.)

RMOV Option

The RMOV option must be set because the QWIKLOG program replaces the unsorted input file with a sorted output file of the same name. The following ODT command sets the RMOV option:

```
SO RMOV
```

OPERATING INSTRUCTIONS

By default, the QWIKLOG program analyzes the currently active log file named SYSTEM/LOG. The active log file can be transferred at any time by means of the LG or TL system command. Also, this file is transferred automatically by the MCP II after reaching maximum capacity. This action renames SYSTEM/LOG as LOG/#<integer> and creates a new SYSTEM/LOG file. Such a transferred log file can be analyzed by use of the following command:

```
EXECUTE QWIKLOG FILE LOGFILE NAME LOG/#<integer>;
```

The 6-digit <integer> identifies the desired log file.

As an operator convenience, if the FILE statement is used to change the NAME file attribute of the LOGFILE file to the single name of LOG, the QWIKLOG program issues a prompt requesting the <integer> to be entered in an AC or AX command. Leading zero (0) characters need not be provided and, as with all program input, the operator's response to the ACCEPT message may be entered in free-form format.

PROGRAM SWITCHES

The QWIKLOG program recognizes switches 0 and 8:

Switch 0

Used to specify the width of the print line. A value of 0 implies 132 print positions, and a value of 1 implies 120 print positions. Users with 120-position line printers can permanently set this switch with the following control instruction:

```
MODIFY QWIKLOG SWITCH 0=1
```

Switch 8

By default, the QWIKLOG program prints both parts of the analysis for all jobs within the log file. If program switch 8 is set to 1, however, the QWIKLOG program issues the following prompts requesting information to control its actions:

JOB SUMMARY? (Y/N)

Enter Y, YES, N, or NO.

GRAPH? (Y/N)

Enter Y, YES, N, or NO.

JOB NUMBER RANGE: "ALL" OR <from> [<to>]

This prompt requests selection of a specific range of jobs for analysis. The keyword ALL specifies that all jobs within the given log file are to be processed (as is done in the default case); otherwise, the first integer (<from>) is used as the lower bound for the job number range to be analyzed. Analysis continues to the end of the log file unless a second integer (<to>) is provided to indicate an upper bound. The specified range can span zero, that is, the <stopping job number> can be less than the <starting job number>. The input for all three QWIKLOG requests can be provided in a single AX or AC system command, in which case processing commences immediately.

Example:

```
% QWIKLOG =907 JOB SUMMARY? (Y/N)  
907AX Y NO 800 860
```

If analysis is done on the currently active SYSTEM/LOG file, the QWIKLOG program stops on reaching its own BOJ record in the log unless a lower value is specified for the <stopping job number>.

WORK FILES

The work file SLOG can be file-equated to a user disk (with the PACK.ID [PID] file attribute) to minimize disk arm movement. For installations with multiple system disk drives, SLOG is automatically placed on the second system disk drive, again in an attempt to minimize disk arm movement. If a second user disk is available, the other work file, GRAFILE, can be file-equated to further reduce disk arm movement. (Refer to the SD system command in section 5 of Volume 1 of the *B 1000 Systems System Software Operation Guide* for information on creating additional system disks.)

OUTPUT – SUMMARY LISTING

The following information is provided in the job summary portion of the QWIKLOG output. Fields inappropriate for a particular job are left blank. For example, a job that has been removed from the schedule (RS system command) has no BOJ or EOJ date and time.

Job Number

An asterisk (*) preceding the job number indicates that the job was part of a job stream. (Refer to the AFTER, AFTER.NUMBER, and THEN program control instructions in section 4 of Volume 1 of the *B 1000 Systems System Software Operation Guide*.)

Program name

Job Accounting Number

This is a sequential value associated with each job which, unlike the job number, is not reset at 9999, and hence provides a unique identification number for each job.

Object program name

This is provided if the execute type is a compilation. For a CLEAR/START operation, this field contains the MCP II version number.

Charge number

This is provided unless the number is zero.

Execute type

Items that can appear are EXECUTE, COMPILE & GO, COMPILE & SAVE, COMPILE TO LIBRARY, GO PART OF COMPILE & GO, and GO PART OF COMPILE & SAVE.

Priorities

Schedule, processor, and memory priorities.

Interpreter <directory-name>

For example, COBOL.

Date and time scheduled

BOJ date and time

EOJ date and time

Elapsed execution time

Environment Statistics

The relevant information for the primary, DMS and IBASIC environments is given. Depending on the job, interpreter time, MMCP time, SMCP time, code overlays and data overlays are given for each environment.

Interpreter time

Accumulated processor time for each environment.

MMCP time

Accumulated MMCP processor time for each environment.

SMCP time

Accumulated SMCP processor time for each environment.

Code overlays

The number of code overlays for each environment.

Data overlays

The number of data overlays for each environment.

Job termination type

Possible termination types are NORMAL, discontinued (DS or DP system command), SYNTAX ERRORS, ABORTED (by a CLEAR/START operation), RS-ed (removed from the schedule with the RS system command), FAM.DEATH, and RUNNING (which indicates that the job was still in the mix when the QWIKLOG program was executed or when the logfile was transferred).

Static memory assigned (in bits)

Dynamic memory assigned (in bits)

Total Run Structure size

In bits. Defined as <static memory> + <dynamic memory> + <run structure nucleus>. This figure does not include structures such as file buffers. Users interested in a more complete analysis of memory requirements for a given program should use the CODE/ANALYZER utility program. In the entry for a CLEAR/START, this field contains the system main memory size in KB (1 KB = 1024 bytes = 8192 bits).

Files

The number of files declared by this program.

Open and closes

The number of opens and number of closes performed by the program or automatically performed by the MCP.

VIRTUAL.DISK segments

Segments of VIRTUAL.DISK (disk for data overlays) assigned.

MNEMONICS

The following mnemonics are used with the chronological graph:

Mnemonic	Meaning
*	CLEAR/START
S	Program Scheduled
E	Program Execution
C	Compiler Execution
R	Program Running (Program still running at time of analysis or transfer of log.)
X	Abnormal Termination (Spotlights an abnormal termination by substituting the X for C or E for the last minute of a job that was discontinued (DS or DP system command) or was aborted due to a CLEAR/START operation.)

ERROR MESSAGES

EMPTY LOG FILE... EOJ

[TRY AGAIN...] ENTER LOG NUMBER

If LOGFILE file is file-equated to the single name of LOG, the QWIKLOG program requests the log identification <integer> with this prompt. The phrase TRY AGAIN... prefaces the next request if the previous AX message was invalid.

FIRST 3 HEXITS NEQ @1AA@
INVALID RECORD TYPE

These two messages appear on the printer file labeled LINE, but not on the operator display terminal, the first five or fewer times an incorrectly-formatted record is found in the log being analyzed. A hexadecimal/EBCDIC printout of the offending record also appears. Processing continues, although results for the job to which the faulty record applies can be erroneous. This problem is indicative of disk file corruption.

HEY...I NEED SOME WORK...EOJ

This message is displayed if the operator enters NO or N in response to both the prompt JOB SUMMARY? (Y/N) and the prompt GRAPH? (Y/N).

SOFTWARE INCOMPATIBILITY: <level> QWIKLOG; <level> MCP...EOJ

The format of log records is subject to change between software releases. To guarantee valid results, the QWIKLOG program ascertains that it is being run with the proper MCPII and terminates with this message if a mismatch is detected.

INVALID INPUT "<token>"

The QWIKLOG program decoded operator input that was contextually invalid.

INVALID TRANSFER HEADER RECORD IN LOG...EOJ

The first record in the specified logfile was not the expected CLEAR/START or pseudo-CLEAR/START record.

LOG HAS NO JOBS IN SPECIFIED RANGE...EOJ

The QWIKLOG program found that the log file being analyzed had no jobs within the range specified by the operator.

LOG XFER RECORD IS GARBAGE...EOJ

In processing a transferred log file, the QWIKLOG program discovered that the last record in the file was not the expected log transfer record.

NO JOBS IN LOG...EOJ

The log file being processed by the QWIKLOG program contained no jobs.

TIMEOUT ON DISK I/O AT <absolute disk address>

A disk read operation timeout occurred when the QWIKLOG program was using INITIALIZER I/O operation to process the currently active SYSTEM/LOG file. This message causes the QWIKLOG program to go to EOJ.

IRRECOVERABLE I/O ERROR ON LOG READ AT <absolute disk address>;

RESULT= <result descriptor>

An exception condition occurred when the QWIKLOG program was using INITIALIZER I/O to read the currently-active SYSTEM/LOG file. The QWIKLOG program goes to EOJ if an irrecoverable disk error occurs.

<file-name> NOT ON DISK...EOJ

The log file on which analysis was requested was not in the disk directory.

NULL INPUT INVALID

The operator entered a null AC or AX response to an ACCEPT message.

(STARTING/STOPPING) JOB NUMBER. "<token>" ISN'T NUMERIC

A non-numeric response was entered for an <integer> specifying the job number range.

PROGRAM TRAPS

The following four program traps within the QWIKLOG program reflect probable software errors. Should they occur, provide your Burroughs Technical Representative with a printout of the dump that is produced and a machine-readable copy of the log file for analysis by Burroughs QWIKLOG development and support personnel.

BINARY SEARCH FAILED: PLEASE SUBMIT LOG TO BURROUGHS...EOJ

COULDN'T HANDLE CLEARSTART: PLEASE SUBMIT LOG TO BURROUGHS...EOJ

GRAFILE EOF: PLEASE SUBMIT LOG TO BURROUGHS...EOJ

LOST SLOG: PLEASE SUBMIT LOG TO BURROUGHS...EOJ

SECTION 22

RD

This section is comprised of seven subsections:

1. **Introduction:** lists the functions of the RD program, provides an overview of the overall section, and presents common syntax usage that underlies use of the RD program.
2. **Capabilities:** summarizes the capabilities of the RD program and specifies hardware, memory, terminal, and software requirements.
3. **Installation:** lists and describes the procedures used to initialize the RD program.
4. **RD Operating Instructions:** covers batch and SMCS execution of the RD program and summarizes termination procedures.
5. **Terminal Commands:** defines the RD remote terminal commands.
6. **ODT Commands:** defines the ODT commands.
7. **Batch Operation:** defines commands for batch execution.

INTRODUCTION

The RD program allows an operator of a remote display terminal to obtain system status information. The information available includes file and program attributes and current system configuration. Inquiry is made by using the standard B 1000 system commands.

The following information can be obtained from the RD program.

- The names of the printer backup, punch backup, and memory dump files.
- The creation dates for disk files.
- The internal and external file names declared in an object program.
- The disk file attributes for a specified file or file family.
- The most recent messages displayed on the system Operator Display Terminal (ODT) from the ODT.QUEUE.
- A display of selected disk segments in either character or hexadecimal format.
- The status of the system and system peripherals.
- A directory of files for a specified file, partial file id, or file family.
- The backup files in the autoprint queue.
- The status of program control attributes for a program on disk, a microcode (firmware) file, or an executing program (job).
- The current time and date.
- The current MCP system options.
- The programs in the waiting and active schedules that are waiting execution.
- A summary of programs currently executing on the system. The user can optionally display all of the jobs currently executing, only the visible jobs, or a specific job number.
- A summary of processor utilization for all programs and the MCP.

Common Syntax

Fundamental syntactical information that pertains to the operation of the RD commands is presented next. The information is presented here rather than with each command to avoid repetition. Once understood, most of this information becomes second nature and may only require review if an RD command performs differently than expected.

Following this subsection, syntax diagrams are not included for those commands that consist of command keyword alone.

Common Syntax:

— < family name > _____|
— < multifile name > _____|
— < file name > _____|
— < partial-file name > _____|
— < mix number > _____|
— = _____|

Semantics:

Family Name

<family name> specifies the user disk identifier of the file. It must be 10 or fewer characters less in length. If <family name> is blank, the system disk is assumed.

Multifile Name

<multifile name> specifies the directory name or family name of the file. It must be 10 or fewer characters in length.

File Name

<file name> specifies the subdirectory file name of the file. It must be 10 or fewer characters in length.

Partial-File Name

<partial-file name> specifies the first or last n characters of a <file name>, where n is between 1 and 9, inclusive. The <partial-file name> must immediately be preceded or followed by the question mark (?) character. The question mark is replaced by any character that, in combination with the n characters entered, generates a valid file name.

Mix Number

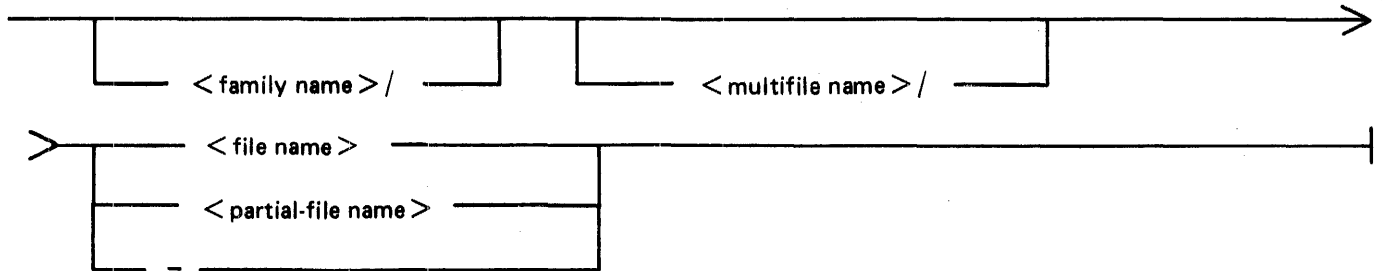
The <mix number> is an integer between 1 and 9999 assigned by the MCP to a program at beginning-of-job.

=

The equals sign character specifies all files under the specified directory.

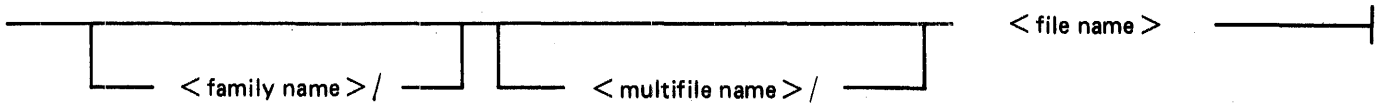
File Identifier

Syntax:



Program Name

Syntax:



SKIP and PAGE Keywords

Syntax:



SKIP

The SKIP keyword causes the RD program to begin listing information after skipping <integer> items.

PAGE

The PAGE keyword causes the RD program to begin listing information beginning on <integer> page. There 22 lines of information per page.

CAPABILITIES AND REQUIREMENTS

The RD program normally responds to an inquiry from a data communications terminal, but can be directed to run in batch mode. In batch mode, the RD program accepts commands from the system ODT or from a card reader and creates line printer output.

The following methods can be used to communicate from a terminal to the RD program:

1. A terminal user can sign on to the RD program using a Message Control System (MCS) program. Text transmitted from that terminal goes to the RD program and responses are returned to that terminal.
2. A terminal user can pass information to the Supervisory Message Control System (SMCS) and receive the response from the RD program without being signed on to the RD program.
3. A system operator can execute the RD program from the system ODT using the Master Control Program (MCP) file-equate capability. The file-equate applies to the remote file defined in the network controller assigned to that terminal.

The RD program does not require the SMCS program product; any Message Control System can be used that permits signing on to programs.

The RD program provides the same information as many of the MCP system commands. The advantage of using the RD program is that it requires less MCP involvement than if the comparable MCP commands were used. By decreasing the time required by the MCP to perform system commands, more processor time is made available for other programs in the mix. MCP system commands such as KA, MX, PD, and KB require the MCP to suspend all other programs in the mix and dedicate itself to processing the request until it is complete. This is not the case when the RD program is used to retrieve the same information.

Support Capabilities

The RD program can support input from a maximum of 16 stations.

A maximum of 22 jobs can be displayed on the screen at a time. Mix information is available on all jobs in the mix.

Hardware Requirements

The following hardware elements are required to execute the B 1000 RD program:

- B 1800 (except B 1830) or B 1900 central processor
- Multiline control or single-line control
- Line adapters as required
- Disk storage
- Operator Display Terminal (ODT)
- Acceptable Terminal Devices: MT 983, MT 985, TD 830

Memory Requirements

The RD program requires 7.5K bytes for static data space and up to 8K bytes for code space depending upon which RD program command is performed. This memory requirement is in addition to the memory required for the MCP in a data communications environment.

Terminal Requirements

The output of the RD program is especially designed for use on the MT 983/MT 985 and TD 830 terminals with a 1920-character screen (24 lines by 80 characters). The RD program utilizes the cursor positioning, line insert, line delete, clear-to-end-of-page, and clear-to-end-of-line capabilities of these terminals.

The TD 820 terminal can also be used with the RD program. However, that terminal does not have all the features of the other terminals, so all the RD program features are not available on the TD 820 terminals. Limitations occur mostly in the job information commands such as JOBS and MX and are noted in the command descriptions.

In batch mode, the RD program requires only a line printer device for output.

Software Requirements

The following software elements are required:

- Operating System (MCP)
- Network Controller
- Supervisory Message Control System (SMCS)
- SYSTEM/MAKEUSER Program

INSTALLATION PROCEDURES

The following steps are performed before the RD program is executed. (Step 2 is required; the others are optional.)

1. Create the (SYSTEM)/USERCODE file.
2. (Required) Compile the network controller, unless the RD program is to be used only in a batch environment.
3. Initialize the MCP Program Table
4. Create the SMCS JOBS file when running with the SMCS (optional).
5. Assign the RD program priorities (optional).
6. Set the RD program switches (optional).

Creating the (SYSTEM)/USERCODE File

The file security system implemented on the B 1000 system requires that a file labeled (SYSTEM)/USERCODE be in the system disk directory. The (SYSTEM)/USERCODE file contains a list of all valid usercode/password combinations and can be created or updated with the SYSTEM/MAKEUSER program.

Compiling the Network Controller

Before executing the RD program, the user must generate the desired network controller program for the RD program. The network controller program is compiled using the NDL compiler. Refer to the *B 1000 Systems Network Definition Language (NDL) Language Manual* for more information on generating the network controller program.

Initializing the MCP Name Table

In order for the network controller program to be automatically executed by the MCP, the name of the network controller program must be given to the MCP. This is done by placing the name of the network controller program into the C slot of the MCP Name Table. To accomplish this, enter the following MCP command:

```
CM C <network controller program name>
```

The MCP responds with the following message:

```
NETWORK CONTROLLER CHANGED FROM "" TO  
"<network controller program name>"
```

To verify that the C slot in the MCP Name Table contains the correct network controller program name, enter the following MCP command:

```
WW C
```

The MCP responds with the following message:

```
C = "<network controller program name>"
```

Creating the SMCS Jobs File

If the RD program is to be executed in conjunction with the SMCS program, placing the RD program in the SMCS JOBS file causes the SMCS program to automatically execute the RD program (1) when the SMCS program goes to beginning of job, or (2) when an SMCS user enters the PASS RD <command>, or (3) when the user signs on to the RD program while it is not in the mix.

The SMCS JOBS file is an optional file containing information about other programs to be run in conjunction with the SMCS program. If the SMCS JOBS file exists, the SMCS program checks it at BOJ as well as when a program SIGN ON or program PASS operation occurs. For a detailed explanation of the SMCS JOBS file and its creation, refer to the *B 1000 Systems Supervisory Message Control System (SMCS) Reference Manual*.

The name of the RD program in the SMCS JOBS file must be RD in order for the MX, WY, and JOBS commands to be automatically forwarded to the RD program by the SMCS program. In addition, when the RD program builds a continuation pass string, the name of the RD program in the SMCS JOBS file is assumed to be RD.

The following is an example entry in the SMCS JOBS file for the RD program:

```
$RD          AUTO-START EXCEPT LOG-ON NO-EOF;  
EXECUTE RD;  
FILE RMT PTL = 2;  
PRIORITY 13;  
CHARGE = 99999;  
SW = @040A00010F@;
```

Setting the RD Program Priority

For optimum system response, the RD program should be assigned a program priority higher than other programs in the mix, but less than the priority of the network controller and the SYSTEM/ODT program. The following example shows how to permanently assign a priority of 13 to the RD program.

```
MODIFY RD PRIORITY 13;
```

If a priority is specified in the SMCS/JOBS file, it takes precedence over any modifications.

Setting the RD Program Switches

Program Switch 0

Used to specify input mode.

Program switch 0 = 1: The RD program accepts input commands from the system ODT until a BYE or EOJ command is entered.

Program switch 0 = 2: The RD program reads a card file labeled CARDS until the end of file is read unless a BYE or EOJ command comes first. The command set accepted from the card file is limited to the batch commands. All output is printed except for error messages and the TEACH command output, which are displayed on the system ODT as well as printed.

Program Switch 1

Used to control the time between memory scans for mix table updating.

Program switch 1 = 0: Default wait time (explained under W (Wait) Command) is used. If program switch 1 is non-zero, the W (Wait) command has no effect. The only way to change the wait-time is to change the value of program switch 1 during the execution of the RD program.

Program switch 1 = not zero: Mix table updating time (in seconds) is calculated by multiplying the switch value by 5; thus, if the value is 2, the time between scans is 10 seconds.

Program Switch 3

Used to specify the limit on the number of programs that can be included in the mix table display. The RD program checks this switch at BOJ.

Program switch 3 = no value, 0, or greater than 11: Defaults to a value of 8 which means that the mix table program display limit is 16.

Program switch 3 = 1 to 11: A value in this range, multiplied by 2, specifies the limit on the number of programs in the mix table display.

Program Switch 7

Enables for CANDE request sets.

Program switch 7 = not zero: Causes the RD program to set TOG[1] in the MCS-to-NDL message equal to 1 instead of 0 on output messages. This allows the RD program to use the CANDE request sets in the network controller. Also, scrolling is disabled by the output request sets.

Program Switch 9

Program termination mode.

Program switch 9 = 0: The RD program terminates whenever the last user signs off, or on completion of a single output if the only input was a PASS input message by way of the SMCS program.

Program switch 9 = 1 through 14: The RD program waits a number of minutes specified by the switch value before terminating. Example: switch = 3, time = 3 minutes.

Program switch 9 = 15: The RD program remains in the mix and does not go to end of job.

OPERATING INSTRUCTIONS

The RD program can be executed in batch mode or in conjunction with the SMCS program, and is terminated with the BYE or EOJ command.

Execution Of The RD Program (Batch Mode)

To execute the RD program in batch mode, enter either of the following commands on the system ODT:

```
EXECUTE RD SWITCH 0 = 1;  
EXECUTE RD SWITCH 0 = 2;
```

If program switch 0 is set to 1, the RD program accepts input commands from the system ODT until a BYE or EOJ command is entered.

If program switch 0 is set to 2, the RD program reads a card file labeled CARDS until the end of file is read or a BYE or EOJ command is found.

The commands available when executing the RD program in batch mode are described later in this section.

Execution Of The RD Program (With SMCS)

When the RD program is executed in conjunction with the SMCS program, the RD program can be automatically executed by the SMCS through the SMCS job-spawning facility. This can occur when the first terminal user signs on or passes to the RD program from a local terminal or when the AUTO-START option is specified in the SMCS JOBS file. In either case, the RD program does not have to be executed through the system ODT.

The remote file for RD program is labeled RMT and uses PROTOCOL TYPE 2 as defined by the SMCS program. PROTOCOL TYPE 2 provides the RD program with the usercode, password, usercode index, and station signal character for a user at a terminal when the user passes commands to the RD program by way of the SMCS program.

Any PASS input command through the SMCS program to the RD program causes the RD program to apply the usercode/password to file identifiers unless the asterisk (*) character convention is used to override the usercode. Therefore, if a user is logged on to the SMCS program, a PASS RD of any directory-search command, such as PD, KA, BF, or DF for a single-name file that is not under the usercode of the user must be preceded with an asterisk (*) character. For example, entering PASS RD PD MCPH causes the RD program to search the disk directory for a file labeled (usercode)/MCPH, but entering PASS RD PD *MCPH causes the RD program to search the system disk directory for MCPH. If the usercode has a default family name, the RD program searches that disk for the file identifier.

Termination Of The RD Program

When the RD program is running, the BYE or EOJ command can be used to bring the RD program to an orderly termination.

The BYE command sends the RD program to end of job if there are no other users signed on to the RD program and if program switch 9 is equal to 0.

The EOJ command sends the RD program to end of job if the EOJ command is entered from one of the following sources:

1. The only station signed on to the RD program.
2. A station that uses the PASS RD EOJ command by way of the SMCS program, if there are no other stations signed on to the RD program.
3. The system ODT, by means of the MCP ACCEPT mechanism; for example, by entering <RD mix number>AX EOJ

TERMINAL COMMANDS

The RD program commands that are entered from remote terminals are described in this section. Syntax diagrams are included for all commands except those that consist of a single term only.

AVL (Available Disk)

The AVL command causes the RD program to display the available disk areas on the specified disk. If a family name is not specified, the system disk is searched for available disk areas.

Syntax:

— AVL —————
 ┌──────────────────┐
 │ │
 │ │
 └──────────────────┘
 < family name >

Example input:

AVL

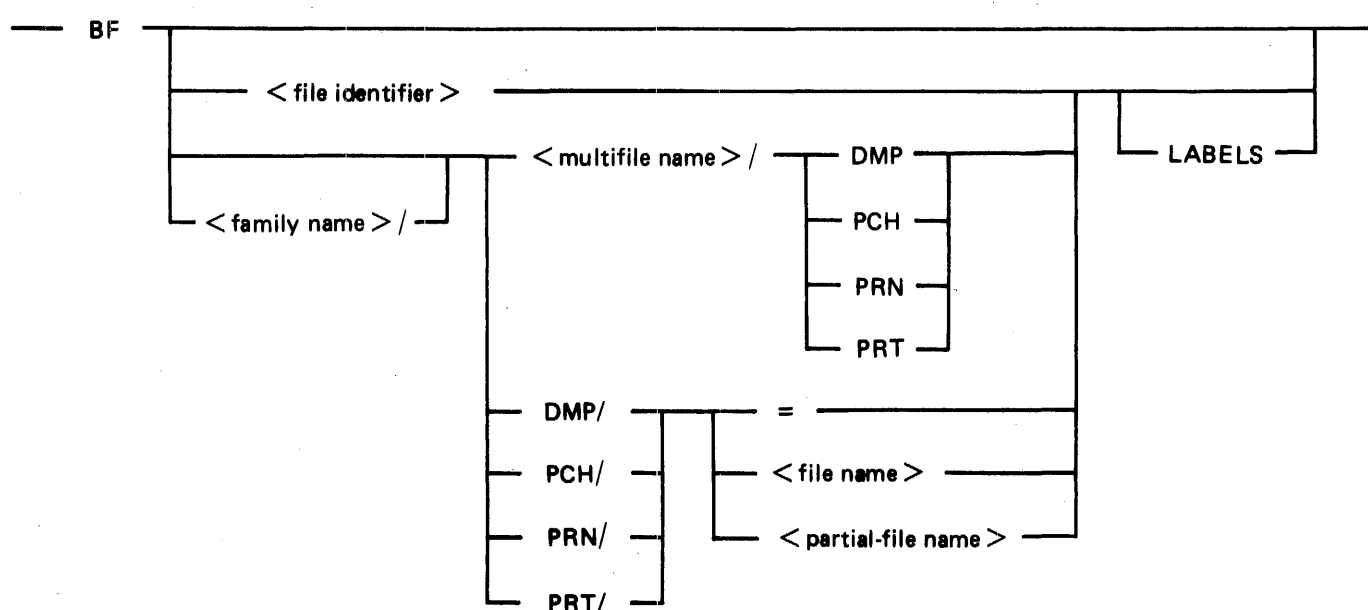
Example output:

```
AVAILABLE DISK SUMMARY:
NUMBER OF CHUNKS      44
LARGEST CHUNK         89842
TOTAL SPACE           127138
TEMPORARY AREAS SUMMARY:
NUMBER OF CHUNKS      76
LARGEST CHUNK         3060
```

BF (Backup Files)

The BF command causes the RD program to search the disk directory for a specific file or for all files with print, punch, or dump file types.

Syntax:



Semantics:

PCH

Displays punch backup file identifiers only.

PRN, PRT

Either displays printer backup file identifiers only.

DMP

Displays dumpfile names only.

LABELS

Displays the name of the file and the MCP-generated label record.

Example input and output:

Input	Output
BF PRT/=	BF = "BACKUP.PRT/1205" BF = "BACKUP.PRT/1300"
BF (USER)/=	BF = "(USER)/#1200" BF = "(USER)/#1201"
BF DMP/1234	BF = "DUMPFIL/1234"
BF PCH/890	BF = "BACKUP.PCH/890"
BF USERPACK/PRT/=	BF = "USERPACK/BACKUP.PRT/1400"
BF PRT/1?	BF = "BACKUP.PRT/1205" BF = "BACKUP.PRT/1300"
BF (USER)/?RD	BF = "(USER)/PRT.RD" BF = "(USER)/XREF.RD" BF = "(USER)/XMAP.RD"
BF *PRT/=	BF = "BACKUP.PRT/1600"
BF *PRT/= LABELS	LINE BACKUP.PRT/1205 TUESDAY,11 DEC 79, 10.11 AM DMPALL JOB #1250 MCP11 ** MARK IX.0 11/29/79 13:35

BRK (Break)

The BRK command causes the RD program to terminate output streams from any of the following commands: BF, DF, KA, KB, KC, KP, MIX, OMX, PD.

Example input and output:

Input	Output
----- BRK	----- GOODBYE or BREAK DONE

BYE (Terminal User Sign Off)

The BYE command causes the RD program to sign a terminal user off the RD program when running under an MCS.

Example input and output:

Input	Output
----- BYE	----- GOODBYE

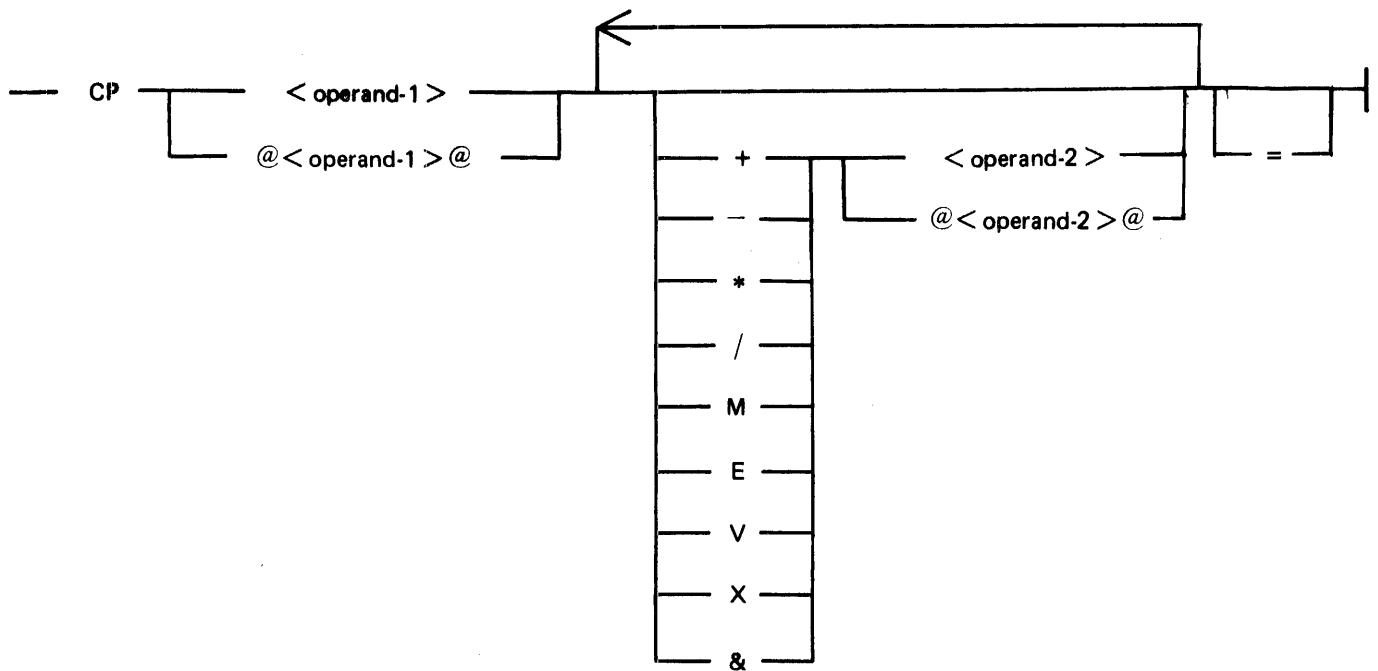
CP (Compute)

The CP command causes the RD program to perform the specified arithmetic operation. The following are the valid operators allowed in the CP command.

- + Add
- Subtract
- * Multiply
- ÷ Divide
- M Mod
- E Exponent
- V Logical Or
- X Exclusive Or
- & Logical And
- = Total

Each operator must be separated by a space.

Syntax:



Semantics:

<operand-1>, <operand-2>

Each specifies the decimal or hexadecimal number to be used. This number can have a maximum value of 16,777,215 or @FFFFFF@.

+, -, *, /

Cause the RD program to perform, respectively, the addition, subtraction, multiplication, and division operations on the operands.

@

Specifies that the enclosed operand is a hexadecimal number.

M

Causes the RD program to perform the modular operation on the operands.

E

Causes the RD program to perform the exponential operation on the operands.

V

Causes the RD program to perform the logical-OR operation on the operands.

X

Causes the RD program to perform the exclusive-OR operation on the operands.

&

Causes the RD program to perform the logical-AND operation on the operands.

=

Causes the RD program to generate the total from an operation. The equal sign (=) character is optional.

Example input and output:

Input	Output
CP 1389	CP: @00056D@ = 1389
CP 2 + 66	CP: @000044@ = 68
CP 138 = 9 =	CP: @000081@ = 129
CP 578 * 999	CP: @08CF8E@ = 577422
CP 690 / 78 * 12 = 10 + 789 =	CP: @00036B@ = 875
CP 780 M 7 =	CP: @000003@ = 3
CP @489@ V 987	CP: @0007DB@ = 2011
CP 89 X 7777	CP: @001E38@ = 7736
CP 777 & @61435@	CP: @000001@ = 1

DF (Date of File)

The DF command causes the RD program to list the creation date and the compile date of a file or files. The dates are given in MM/DD/YY format, where MM is the month, DD is the day, and YY is the year. A maximum of 22 files are displayed per page of output.

Syntax:

— DF < file identifier >

— < skip/page > —

Example input:

```
DF USERPACK/(USER)/=  
DF (USER)/L?  
DF (USER)/?RD  
DF SYSTEM/= PAGE 2  
DF MASTER/M? SKIP 7  
DF *DMPALL
```

Example output:

```
"SYSTEM/INIT"          COMPILED ON 05/31/79 AT 18:40:20.9  
"SYSTEM/ARCH"         COMPILED ON 06/19/79 AT 10:09:41.8  
"SYSTEM/ARCHIVDUMP"   COMPILED ON 06/25/79 AT 09:43:55.4  
"SYSTEM/BUILDTRAIN"   COMPILED ON 10/24/78 AT 08:36:40.8  
"SYSTEM/DISK.DUMP"    COMPILED ON 08/16/79 AT 15:55:54.3
```

Example input:

```
DF USERPACK/(USER)/=  
DF (USER)/L?  
DF (USER)/?RD  
DF SYSTEM/= PAGE 2  
DF MASTER/M? SKIP 7  
DF *DMPALL
```

Example output:

```
"SYSTEM/INIT"          COMPILED ON 05/31/79 AT 18:40:20.9  
"SYSTEM/ARCH"         COMPILED ON 06/19/79 AT 10:09:41.8  
"SYSTEM/ARCHIVDUMP"   COMPILED ON 06/25/79 AT 09:43:55.4  
"SYSTEM/BUILDTRAIN"   COMPILED ON 10/24/78 AT 08:36:40.8  
"SYSTEM/DISK.DUMP"    COMPILED ON 08/16/79 AT 15:55:54.3
```

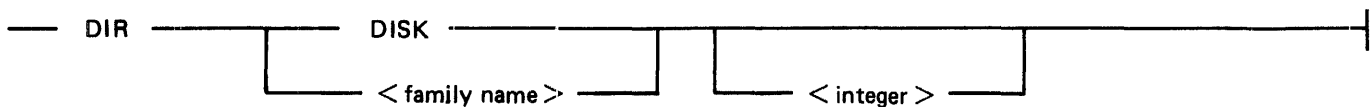
DIR (Directory)

The DIR command causes the RD program to display all directory names for a specified disk. Names of directories with no subdirectories are prefixed with asterisk (*) characters.

Specifying DISK as the family name causes the RD program to search the system disk for directory names.

The main disk directory is maintained in 16 disk segments. If an integer between 0 and 15, inclusive, follows the family name in the DIR command, the RD program begins listing the directory names at the integer segment specified.

Syntax:



Semantics:

DISK

Specifies the system disk for the disk directory search.

<integer>

Specifies the segment offset in which to begin the disk directory search. The integer must be between 0 and 15, inclusive.

Example input and output:

Input	Output
DIR DISK	*DMPALL
DIR USER	MASTERFILE
DIR DISK 7	(USER)
	*A
	UPDATE

EOJ (End of Job)

The EOJ command causes the RD program to terminate. The EOJ command can only be entered from (1) the only station signed on to the RD program, (2) a station that passes to the RD program by way of the SMCS program when there are no other stations signed on to the RD program, or (3) the system ODT by means of the MCP ACCEPT mechanism.

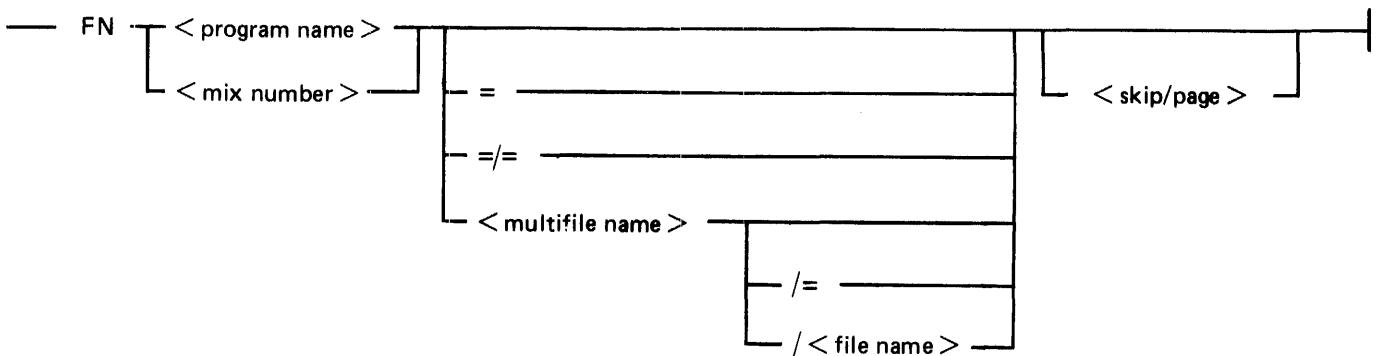
FN (File Name)

The FN command causes the RD program to display internal and external file identifiers of a codefile or a running job. The external name(s) and the corresponding internal file name(s) are listed in relative file number order beginning with 0.

If all of the requested file identifiers are not displayed on the first output screen, the RD program provides the necessary input to continue the FN command.

A maximum of 22 internal and external file identifiers are displayed per page of output.

Syntax:



Example input:

```
FN RD =/=
FN PROGRAM/USER =/= SKIP 7
FN DMPALL = SKIP 5
FN CANDE = PAGE 3
```

Example output (displayed if "FN RD =/=" is entered):

```
0 = RMT      "RMT"
1 = DSK      "DSK"
2 = LINE     "LINE"
3 = CARDS    "CARDS"
END FN.
```

FRZ (Freeze)

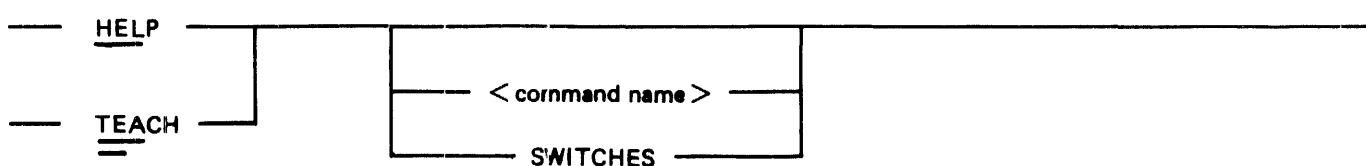
The FRZ command causes the RD program to remain in the system mix. This command is useful only when the RD program is scheduled to terminate if no input is processed after a time period specified by program switch 9

Example input and output:

Input	Output
FRZ	FRZ SET
	FRZ RESET

HELP (or TEACH)

The HELP (or TEACH) command causes the RD program to display all the valid RD commands. If the command is followed by a valid RD command, the RD program displays the syntax and a brief description of the RD command.



Semantics:

<command name>

The <command name> specifies a valid RD program command.

SWITCHES

If HELP SWITCHES (or TEACH SWITCHES) is entered, the RD program displays the current switch settings and the description of each switch.

Example output:

If HELP KA (or TEACH KA) is entered, the following is displayed:

```
PD KA <FILE NAME> <FAMILY>/= [SUM(MARY)]
[SKIP <SKIP-COUNT> PAGE <PAGE #> :
LISTS FILE(S) (AND ATTRIBUTES IF 7KA7; FILE ID 7DSKAVL7
LISTS AVAILABLE DISK SPACE)
```

If HELP FN (or TEACH FN) is entered, the following is displayed.

```
FN <PROGRAM-NAME> <JOB NUMBER> [<EXTERNAL-FILE-NAME>]
[SKIP PAGE <#>] : LIST ASSOCIATED INTERNAL AND EXTERNAL
FILE NAME(S) DECLARED IN <PROGRAM> OR <JOB>
```

ID (Identification)

The ID command causes the RD program to display the version, release level, compile date, compile time of the RD program, and the NDL Logical Station Number (LSN) of the station entering the ID command.

Example input and output:

Input	Output
ID	RD 10.0.7 OF 02/21/81 09:29, LSN 009

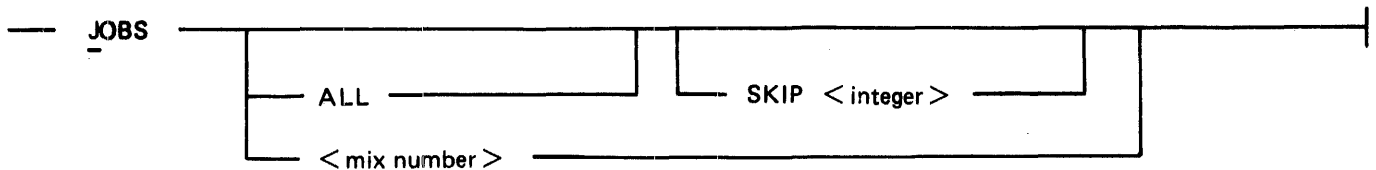
JOBS (Program Information)

The JOBS command causes the RD program to display information regarding a program, a specified range of programs, or all the programs in the waiting schedule and system mix.

The RD program continues to update the mix table for terminals that are signed on to the RD program.

Refer to the WS and MIX commands for an explanation of the waiting schedule and system mix functions and output.

Syntax:



Semantics

ALL

Specifies that both the visible and invisible jobs are to be displayed in the mix table; otherwise, only the visible jobs are displayed.

SKIP

Causes <integer> number of jobs to be skipped when displaying the mix table.

Example input:

```
JOBS
JOBS ALL
JOBS 6523
JOBS SKIP 22
```

Example output:

Refer to the WS command for an example of the format of the waiting schedules and to the MIX command for an example of the format of the mix tables.

KA

The KA command causes the RD program to display the following information:

** NAME **

The name of the file(s) given in the KA command.

TYPE

File type; for example, code, data, printer backup, dump, intrinsic.

RSZ

Number of characters (bytes) allowed in a record.

R.B

Number of records per block.

B.A

Number of blocks per area.

S.A

Number of segments per area.

AREAS

Total number of areas in use and requested.

EOF

End-of-file pointer or total number of records.

CREATN

File creation date.

UPDATE

Last date the file was updated.

ACCESS

Last date the file was accessed.

ST

File security type: public (PB) or private (PV).

SU

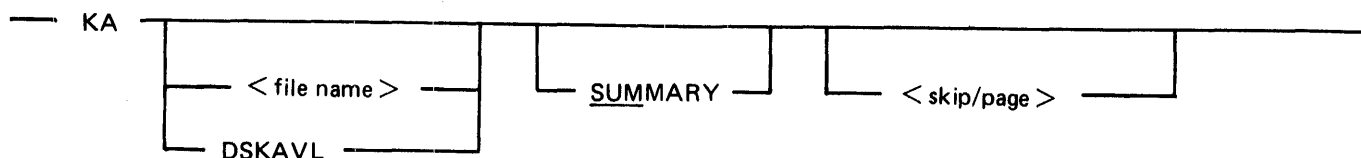
File security use: input (IO), output (OT), or input/output (IO).

US

Number of users who have the file open.

A maximum of 22 files are displayed per page of output.

Syntax:



Semantics:

DSKAVL

Displays the available disk space on the specified disk in two categories: available disk space and space that is in use temporarily. The summary for each category includes the number of areas, the size of largest area, and the total space available or in temporary use.

To obtain these statistics while logged on to the SMCS program and/or the RD program, an asterisk (*) must prefix the DSKAVL option.

Examples:

```
KA *DSKAVL
KA USERPACK/*DSKAVL/
```

SUMMARY

Displays the total number of files, the largest file, the total number of segments, records, and bytes and the average number of segments, records, and bytes for a family of files.

```
KA = SUMMARY
```

If logged on to the SMCS program and/or the RD program, the equal sign (=) character is valid and causes the RD program to give a summary of all the files for the usercode logged on.

```
KA USERPACK/MASTER/= SUMMARY
```

Produces a summary of all the files with a file name USERCODE/MASTER/<file-name>.

Example inputs:

```
KA SYSTEM/F?
KA USERPACK/(USER)/= PAGE 2
KA DSKAVL
KA MASTERFILE
KA MASTERFILE/I?
KA MASTERFILE/?I
KA *DMPALL
KA *MASTER/= SUMMARY
```

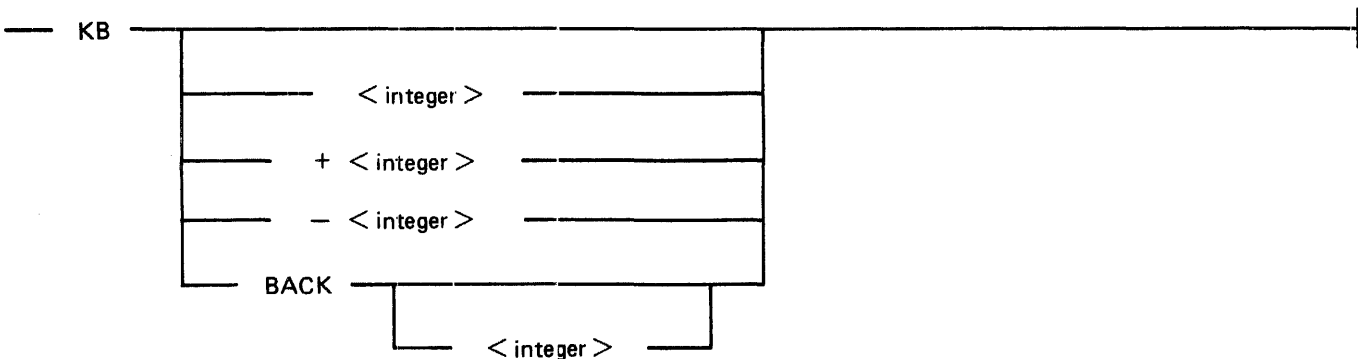

Example output:

** NAME **	TYPE	RSZ	R.E	B.A	S.A	AREAS	EOF	CREATN	UPDATE
ODT.QUEUE	DATA	180	1	200	200	1/ 1	200	NODATE	NODATE
TRAIINTABLE	DATA	360	1	37	74	1/ 1	37	012679	120980
BUILDTRAIN	CODE	180	1	28	28	1/ 1	28	102478	120980
INIT.OLD	INTP	180	1	59	59	1/ 1	59	081380	021381
ODT	CODE	180	1	209	209	1/ 1	209	021781	031081
ELOG	LOG	180	1	100	100	7/ 25	700	032781	032781
CONTROLLER	CODE	180	1	423	423	1/ 1	423	031781	033081
NIF	DATA	90	1	3	6	1/ 1	9	031781	033081
BACKUP	CODE	180	1	103	103	1/ 1	103	031181	033081
COMPARE	CODE	180	1	107	107	1/ 1	107	031281	033081
FILE.INIT	CODE	180	1	37	37	1/ 1	37	091879	120980

KB (ODT Queue)

The KB command causes the RD program to display the system ODT queue. Various KB command parameters can be specified for paging through the system ODT queue.

Syntax:



Semantics:

< integer >

Causes the RD program to start listing the ODT queue from the segment number specified by <integer>. Valid entries for <integer> range from 0 to 1023, inclusive.

+ < integer >

Causes the RD program to start listing the ODT queue from the (current + <integer>) segment.

- < integer >

Causes the RD program to start listing the ODT queue from the (current - <integer>) segment.

BACK

Causes the RD program to display one screen page back from the most recently displayed page of the ODT queue. The station must be signed on the RD program in order to use the BACK option.

BACK <integer>

Causes the RD program to start listing the ODT queue from the (current - <integer>) segment. The station must be signed on the RD program in order to use the BACK <integer> option.

Example inputs:

KB	Displays most current ODT queue entries.
KB BACK	Displays the ODT queue one back from the current page.
KB BACK 2	Displays the ODT queue two segments back from the current page.
KB 30	Displays ODT queue beginning at segment 30.
KB +10	Displays ODT queue beginning at (current +10) segment.
KB -50	Displays ODT queue beginning at (current -50) segment.

Example output:

```

THIS=48 LATEST=53
13:35:16.1 0 RMT (SOFT) DMPALL =7619 "MCP/LIST/" = "MCP/(SOFT)/#3264"
13:35:17.0 | ZIP QU CANDE/CONTROL LS SZ 91 US SOFT/??? 7619AXLST EMPNO
A
13:35:21.2 0 RMT (SOFT) DMPALL =7619 "MCP/(SOFT)/#3264" RELEASED
13:35:27.5 0 RMT (SOFT) DMPALL =7619 "MCP/LIST/" = "MCP/(SOFT)/#3265"
13:35:28.4 | ZIP QU CANDE/CONTROL LS SZ 91 US SOFT/??? 7619AXLST PROJEC
T A
13:35:32.6 0 RMT (SOFT) DMPALL =7619 "MCP/(SOFT)/#3265" RELEASED
13:35:38.8 0 RMT (SOFT) DMPALL =7619 "MCP/LIST/" = "MCP/(SOFT)/#3266"
13:35:42.4 0 RMT (SOFT) DMPALL =7619 "MCP/(SOFT)/#3266" RELEASED
13:35:44.2 0 RMT % (SOFT) DMPALL =7619 ENTER SPECS
13:35:45.7 0 RMT (SOFT) DMPALL =7619 ACCEPT.
13:35:51.1 | ZIP QU CANDE/CONTROL LS SZ 91 US SOFT/??? 7619AX
13:35:54.1 0 RMT (SOFT) DMPALL =7619 EOJ. TIME = 13:35:53.8
13:36:05.1 | ZIP ZQ SMCS/SCHQ QU SMCS/##00000013 LS SZ 0 US SOFT/??? PB
=/=
13:36:13.5 0 RMT (SOFT) SYSTEM/BACKUP =7620 BOJ. PP=9, MP=5 TIME = 13:
13:36:35.1 0 "MCP/(SOFT)/#3263" REMOVED
13:36:45.6 0 "MCP/(SOFT)/#3264" REMOVED
13:36:56.3 0 "MCP/(SOFT)/#3265" REMOVED
13:37:06.0 0 "MCP/(SOFT)/#3266" REMOVED
13:37:11.2 0 RMT (SOFT) SYSTEM/BACKUP =7620 EOJ. TIME = 13:37:10.9

```

The system ODT queue can be up to 1024 (0 through 1023) segments long and wraparound can occur. When the MCP has an entry for the system ODT queue and the current ODT queue pointer is at the maximum segment, the ODT queue pointer is reset to 0 and the MCP then writes over older ODT queue messages. Wraparound in the ODT queue, whether it occurs naturally or by entry of a + <integer> or a - <integer>, is valid and can be handled by the RD program.

Example 1:

Assume an ODT queue size of 200, which is the default. If KB 193 is entered, the RD program displays a screen of output containing segments 193 through 198. If a blank prompt is transmitted, the RD program displays another screen containing segments 198 through 2.

Example 2:

If the ODT queue is set at 175 and KB +100 or KB -100 is entered, the RD program displays a screen of output beginning at segment 75.

The KB command can repeat one or two lines from the ODT queue on successive screens of output.

When the ODT queue is new or has just been expanded, the unused portions of the queue are filled with zeros. The RD program will skip these entries.

Each message displayed contains a time stamp, I or O (input or output) indication, message source, and text. Between three and six disk segments of ODT queue information are displayed on each screen.

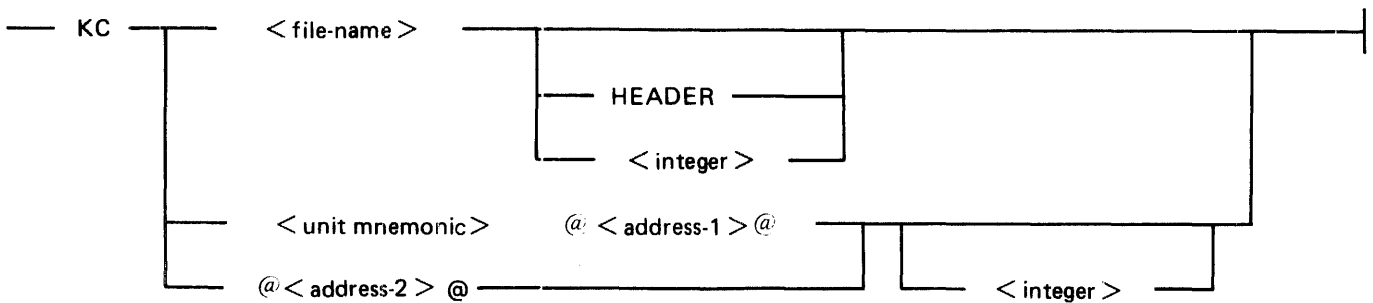
Each screen of output includes a heading, with the words THIS and LATEST. THIS indicates the segment number of the ODT queue where the current screen begins, and LATEST refers to the segment to which the MCP is adding new entries.

In batch mode, the RD program only allows the syntax of the form KB or KB <integer>. Entering the KB command causes the entire system ODT queue to be printed. Entering KB <integer> causes the most current <integer> segments of the system ODT queue to be printed.

KC

The KC command causes the RD program to display the specified disk sectors in character format. A prompt is supplied by the RD program if a user is signed on to the RD program or passes to the RD program by way of the SMCS program. If the prompt is transmitted, the RD program displays the next disk sectors. Private files are listed by the KC command only if the user has logged on with the same usercode as the multifile name of the file or if the user has logged on with a privileged usercode. All public files are listed.

Syntax:



Semantics:

HEADER

Displays the disk file header of the named file.

<integer>

Displays the number of sectors specified by <integer>. The maximum number is 1000 in remote mode. There is no maximum number in the batch mode, and the default is 1.

<unit mnemonic>

Specifies the unit name of the disk drive. Unit mnemonic can be of the form DPx, DCx, and DKx, where x is A, B, C, ..., Z.

<address-1>

Specifies the sector address on the disk unit.

<address-2>

Specifies the disk address.

Example inputs:

```
KC DMPALL HEADER 5
KC DPA @456789@
KC @F20123456 7
KC RD
KC *DMPALL
```

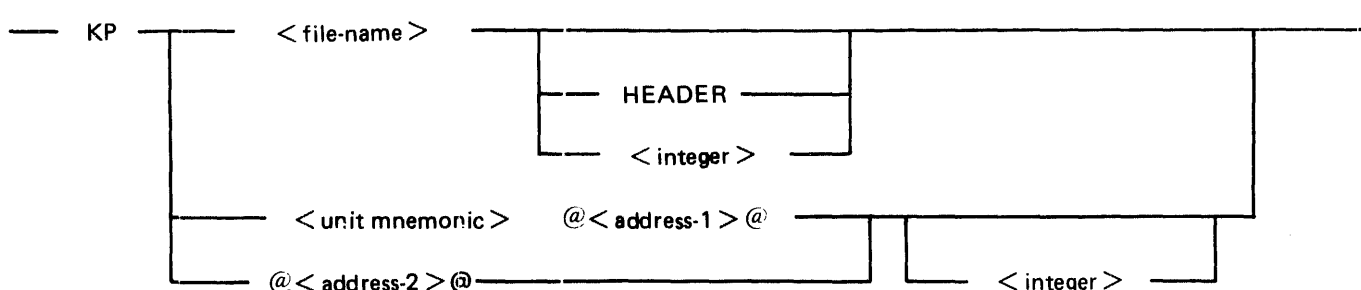
Example output:

```
@F2400088C@
"00100H
"
"          DISK          00200FONEDISK IPE F 180 180
"
@F2400088D@
"00210 C DIGIT8
"
"A1          1 DISK          00300FISAM    0  F 180 10 5"
"
@F2400088E@
"004001ONEDISK NS  01
"
"          1  1 DUMMY          005001
"
@F2400088F@
"0060001SAM    D          01
"
" DUMMY          5          007000
"
```

KP

The KP command causes the RD program to display the specified disk sectors in hexadecimal format. A prompt is supplied by the RD program if a user is signed on to the RD program or passes to the RD program by way of the SMCS program. If the prompt is transmitted, the RD program displays the next disk sectors. Private files are listed by the KP command only if the user has logged on with the same usercode as the multifile name of the file or if the user has logged on with a privileged usercode. All public files are listed.

Syntax:



Semantics:

HEADER

Displays the disk file header of the named file.

<integer>

Displays the number of sectors specified by <integer>. The maximum number is 1000 in remote mode. There is no maximum number in the batch mode and the default is 1.

<unit mnemonic>

Specifies the unit name of the disk drive. Unit mnemonic can be of the form DPx, DCx, and DKx, where x is A, B, C, ..., Z.

<address-1>

Specifies the sector address on the disk unit.

<address-2>

Specifies the disk address.

Example inputs:

```
KP DMPALL HEADER
KP DPA @456789@
KP @F20123456 5
KP RD 7
KP *DMPALL
```

Example output:

```
@F2400088C@
FOFOF1FOFO C840404040 4040404040 4040404040 4040404040 4040404040
4040404040 4040404040 4040404040 4040404040 4040404040 4040404040
4040404040 4040404040 4040404040 4040404040 4040404040 4040404040
FOFOF2FOFO C6D6D5C5C4 C9E2D240C9 D7C540C640 F1F8FO4OF1 F8FO404040
4040404040 40404040C4 C9E2D24040 4040404040 4040404040 4040404040
4040404040 4040404040 4040404040 4040404040 4040404040 4040404040
```

```
@F2400088D@
FOFOF2F1FO 405B40C4C9 C7C9E3F840 4040404040 4040404040 4040404040
4040404040 4040404040 4040404040 4040404040 4040404040 4040404040
4040404040 4040404040 4040404040 4040404040 4040404040 4040404040
FOFOF3FOFO C6C9E2C1D4 40404040D6 404040C640 F1F8FO4040 F1FO4040F5
C1C9404040 4040F140C4 C9E2D24040 4040404040 4040404040 4040404040
4040404040 4040404040 4040404040 4040404040 4040404040 4040404040
```

MIX

The MIX command causes the RD program to display information regarding a program, a specified range of programs, or all the programs in the system mix.

The RD program continues to update the mix table for terminals that are signed on to the RD program.

The following information is provided for each job in the mix:

Heading	Description
MIX#	Job Number
PP	Processor Priority
MP	Memory Priority
CPU TIME	Cumulative Processor (CPU) Time
M	Media 1 - Job is in memory. 0 - Job is "rolled out" to disk.
OVLY	Cumulative Overlay Count

The current system time and the number of jobs running are displayed on the heading line.

For each job in the mix table, the job number, CPU time, the overlay count, whether or not the job is in memory, the program name, and job status are displayed. If the job was executed with a usercode, the usercode is enclosed in parentheses and displayed before the program name. If the job is a compile, the name of the resulting object program is displayed next. If the job requires operator action, the job status is displayed in negative video.

If the user is signed on the RD program, the MCP and program processor utilization is displayed to the right of the job status.

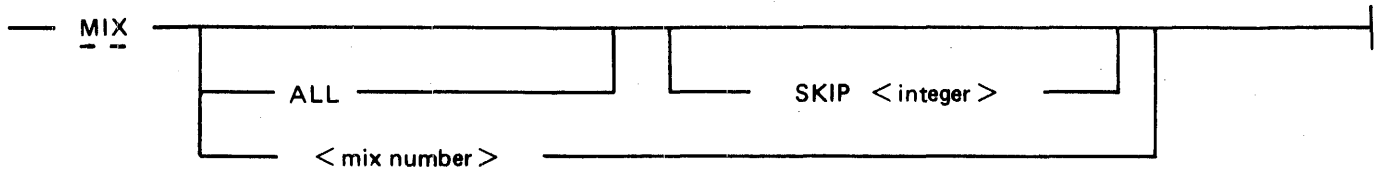
If no options are specified, only the visible jobs are displayed in the mix table. If the ALL option is specified, the invisible jobs are displayed in addition to the visible jobs.

If a mix number is specified, the beginning of job (BOJ) time, the compile date of the program, the compiler used to compile the program, the static memory requirements, the execute type, the number of files, the number of code and data overlays, the status of the protected boolean, the status of the privileged boolean, the parent job number, the parent queue of the program, the program switches, the memory and processor priorities, the processor time used, and the elapsed time since BOJ are displayed for the specified job. This information is not continually updated.

The MIX and JOBS commands perform the same function except that the JOBS command provides information regarding both the waiting schedule and the system mix whereas the MIX command only provides information regarding the system mix.

The number of jobs that are displayed in the mix table is controlled by the setting of program switch 3 and the value specified by the PRG ODT command. A maximum of 22 jobs can be displayed in the mix table.

Syntax:



Semantics:

ALL

The ALL option specifies that both the visible and invisible jobs are to be displayed in the mix table; otherwise, only the visible jobs are displayed in the mix table.

SKIP

The SKIP option causes <integer> number of jobs to be skipped when displaying the mix table.

Examples:

If "MIX" is entered, only the visible jobs are displayed.

AARDVARK		SAVE=216 KB		20%		8:15:00.4		WED. 03/02/83	
MX#	PP MP	CPU TIME	OVLY	M	JOBS RUNNING= 7		MIX LIMIT= 15		
234	13 13	5.9	88	1	SMCS	----->	WAIT		
238	12 12	1.0	35	0	CANDE	----->	WAIT		
235	11 11	26.6	19	1	RD	----->	EXECUTING		
237	11 11	1:15.6	39	1	(TEST) APPROG	----->	WAIT		
236	09 09	.4	33	0	SYCOM:PS	----->	WAIT		

If "MIX ALL" is entered, then all jobs, both visible and invisible are displayed.

AARDVARK		SAVE=219 KB		20%		8:16:27.2		WED. 03/02/83	
MX#	PP MP	CPU TIME	OVLY	M	JOBS RUNNING= 7		MIX LIMIT= 15		
225	15 15	4.0	30	1	SYSTEM/ODT	----->	EXECUTING		
224	15 15	1:13.1	49	1	GEM/NC_15FEB	----->	EXECUTING		
234	13 13	6.0	88	1	SMCS	----->	WAIT		
238	12 12	1.0	35	0	CANDE	----->	WAIT		
235	11 11	27.7	19	1	RD	----->	EXECUTING		
237	11 11	1:18.0	53	1	(TEST) APPROG	----->	WAIT		
236	09 09	.4	33	0	SYCOM:PS	----->	WAIT		

If "MIX 237" is entered, the following information is displayed.

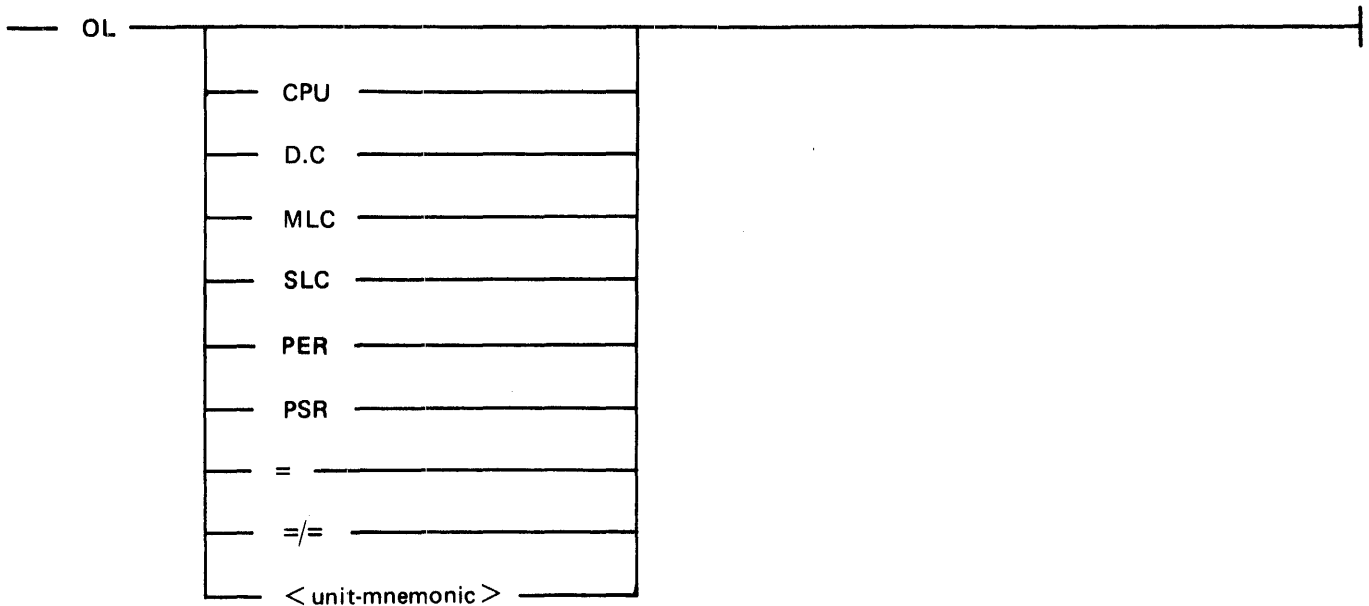
```
(TEST) APPROG = 237      8:21:46.8

BOJ TIME -----> 03/02/83 AT 7:31:23.5
DATE COMPILED -----> 02/09/83 AT 14:07:04.5
COMPILED WITH -----> SDL2/BINDER VERSION (11/15/82)
STATIC / DYNAMIC MEMORY (BYTES) -----> 8750 + 25000 = 33750
VIRTUAL DISK SEGMENTS -----> 1500
EXECUTE TYPE -----> (EXECUTE)
CHARGE NUMBER -----> 92084
NUMBER OF FILES -----> 10
CODE / DATA OVERLAYS -----> 37 + 16 = 53
PROTECTED -----> YES
PRIVILEGED -----> YES
SESSION NUMBER -----> 4
PARENT JOB NUMBER -----> 234
PROGRAM'S PARENT QUEUE -----> ("SMCS"/"MCPQ")
PROGRAM SWITCHES -----> @0000000000@
MEMORY / PROCESSOR PRIORITY -----> 11 / 11
SECONDS BEFORE DECAY -----> NONE
PROGRAM MAXIMUM TIME -----> NO LIMIT
PROCESSOR / ELAPSED TIME -----> 1:31.2 / 50:23.4
AVERAGE PROCESSOR UTILIZATION -----> 3%
```

OL

The OL command entered alone causes the RD program to return a complete display that includes CPU type, memory size and type, I/O type, slave processor presence and port, and the hostname, plus the unit, type, PCU, and status (all explained below) of each peripheral device. Compound entries are described under Semantics.

Syntax:



Semantics:

CPU

Displays the processor type (B 1860, B 1900, etc), S-memory type and size, I/O type, slave processor presence and port number, hostname.

D.C

Provides data communications information, including the number of stations in the network controller, the lines that are in use, the name of the network controller in use, and the locations of the data communication controls if any are present.

MLC

Displays the location of the multiline control.

SLC

Displays the location of the single-line control.

PER

Displays the status of each peripheral unit, as follows:

UNIT – the unit mnemonic name,

TYPE – name of the hardware control,

PCU – specifies a 12-bit (hex) port, channel, and unit number.

STATUS – provides any of the following that are applicable: label (name), serial number, creation date, tape type, printer translate table-id, current job number, and program name; plus any applicable items from the following table:

AVL-IN, AVL-OUT	Available as input, output
SVD, TO-BE-SVD	Saved, to be saved
LCKD	Locked
XCH	Unit is on an exchange
OFFLN-IN-USE	Off line but in use
DMS-AUDIT	In use as DMSII audit

PSR

Displays the status of the pseudo readers.

=, =/=

Identical to OL alone.

<unit mnemonic>

The name of one peripheral unit. (OL PER displays all of them.)

Example inputs:

OL CPU	OL PER	OL DP
OL D.C	OL PSR	OL MTC
OL MLC	OL =	OL CRA
OL SLC	OL =/=	OL MT

Example output:

UNIT	TYPE	PCU	STATUS
LPA	PC.5	730	"016" AVL-OUT
MTA	PETAPE	7B2	"016" XCH
DPA	DPC1	790	"SYSTEMIX.0 (S) SER. #04125
CDA	CD80	740	

OMX (Offline Mix)

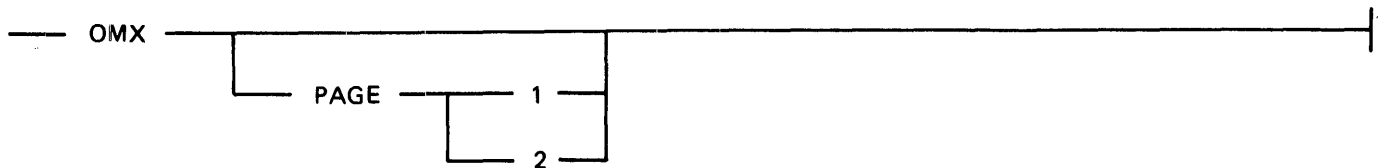
The OMX command causes the RD program to start sending MIX messages to a station that is not signed on to the RD program. This allows the current mix table information to be available without being signed on to the RD program.

To start the offline mix function, enter OMX. If not signed on to the RD program, enter PASS RD OMX to the SMCS program. If the command is entered while signed on to the RD program, then the station is automatically signed off the RD program. The PAGE 2 option causes the RD program to send the mix table output to page 2 of a MT 983/MT 985 or TD 830 terminal.

The BYE or BRK commands are used to terminate the OMX command and are sent to the SMCS program using the SMCS PASS command.

Refer to the MIX command for an explanation of the output formats produced by the OMX command.

Syntax:



Semantics:

PAGE

Causes the mix table to be displayed on screen 1 or screen 2. This option is only valid for the MT 983/MT 985 and TD 830 terminals.

Example inputs:

```
OMX
OMX PAGE 1
OMX PAGE 2
```

```
PASS RD BRK
PASS RD BYE
```

Example output:

Refer to the MIX command for an example of the output that results from the OMX command.

PD (Print Directory)

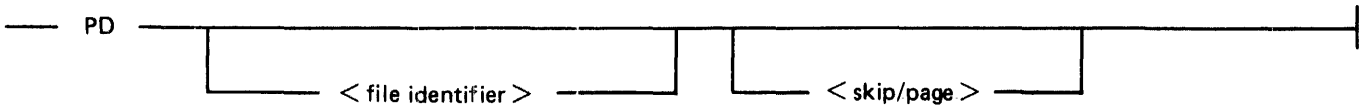
The PD command causes the RD program to display the directory for a file identifier or a family of files.

Standard B 1000 system file identifiers are used, including usercodes and the asterisk (*) character. The default family name is applied to usercode families if a family name or asterisk (*) character is not specified. The SKIP or PAGE options are valid only for the <multifile name>/= form of the command. The PD command displays 44 file identifiers per screen. The PD command displays the entire file identifier for each file found.

The PD command can handle a limit of 999 files for each user signed on to the RD program.

A maximum of 44 files can be displayed per page of output.

Syntax:



Example inputs:

```
PD USERPACK/(USER)/J?
PD (USER)?RD
PD SYSTEM/=
PD =/=
PD =/= SKIP 40
PD (USER)/= PAGE 3
PD *DMPALL
```

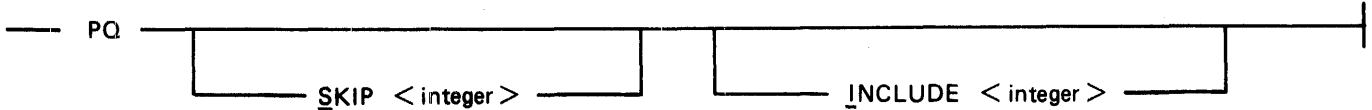
Example output:

```
PD = "SYSTEM/INIT"           PD = "SYSTEM/BUILDTRAIN"
PD = "SYSTEM/COMPARE"       PD = "SYSTEM/DISK.INIT"
PD = "SYSTEM/ELOG"          PD = "SYSTEM/TRAINABLE"
PD = "SYSTEM/BACKUP"        PD = "SYSTEM/FILE.INIT"
PD = "SYSTEM/ICMD.INIT"     PD = "SYSTEM/IS.MAINT"
PD = "SYSTEM/LDCONTRL"      PD = "SYSTEM/LOGOUT"
PD = "SYSTEM/MAKEUSER"      PD = "SYSTEM/ELOGOUT"
END PD.
```

PQ (Print Queue)

The PQ command causes the RD program to list file identifiers that are in the AUTOPRINT queue of the system, waiting to be printed. Also, the number of entries in the queue is displayed. Each entry of the queue that is listed includes the queue element number and the file identifier. Portions of the AUTOPRINT queue can be requested by using the SKIP and INCLUDE options.

Syntax:



Semantics:

SKIP

Causes the RD program to begin displaying the backup printer files at the <integer> number printer file in the AUTOPRINT queue.

INCLUDE

Causes the RD program to display up to the <integer> number of backup printer files contained in the AUTOPRINT queue.

Example inputs:

```
PQ  
PQ SKIP 9  
PQ S 7  
PQ INCLUDE 6  
PQ I 6
```

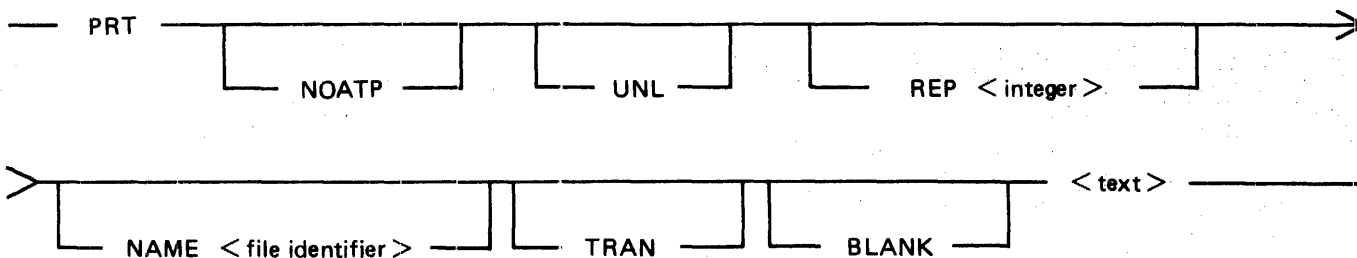
Example output:

```
AP Q CONTAINS 4/1023 MAX MESSAGES  
 1 = BACKUP.PRT/183  
 2 = BACKUP.PRT/184  
 3 = BACKUP.PRT/185  
 4 = BACKUP.PRT/186
```

PRT (Print)

The PRT command causes the RD program to write text from a terminal screen to a printer file. This text must follow the PRT command and be included in the transmission to the RD program. The records in the printer file are written in the image of the terminal screen (80 by 20) and can be useful for a hard copy history of important information.

Syntax:



Semantics:

NOATP

Inhibits printing of the file. The default is autoprint.

UNL

Creates an unlabeled printer file. The default is labeled.

REP <integer>

Repeats printing of the file <integer> number of times. The default is one copy.

NAME

Specifies the user-named backup file identifier. The default name is LINE.

TRAN

Translates the <text> to upper case before printing. The default is no translation.

BLANK

Causes the RD program to leave blank all of the PRT command options that appear between PRT and the text when the information on the terminal screen is transmitted to the RD program. The default is no blank.

<text>

Specifies the data to be printed.

Example inputs:

```
PRT NOATP UNL REP 9 <text>  
PRT NAME (USER)/#1245 <text>  
PRT TRAN <text>
```

When the RD program is executed in conjunction with the SMCS program, print files created by the PRT command have the usercode name applied to the print file identifier. This can be overridden by using the NAME option in the PRT command. The following examples illustrate the use of the PRT command.

Examples:

PRT NAME A/B

If a user is logged on to the RD program with a nonprivileged usercode, a security error results.

PRT NAME *A

The print file identifier is A.

PRT NAME A

The print file identifier is A if the user is not logged on to the RD program. The print file identifier is (<usercode>)/A if the user is logged on to the RD program.

Syntax:

<skip/page>



Semantics:

SKIP

The SKIP keyword causes the RD program to begin listing information after skipping <integer> items.

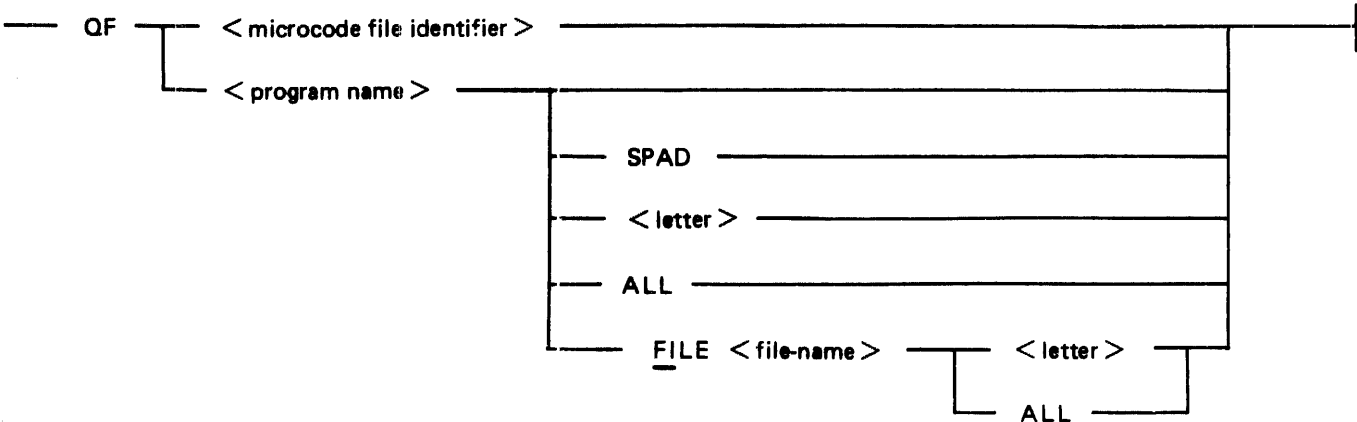
PAGE

The PAGE keyword causes the RD program to list information beginning on <integer> page of information. Each page can display 22 lines of information.

QF (Query File)

The QF command causes the RD program to query the attributes of a codefile.

Syntax:



Semantics:

<microcode file identifier>

Specifies the name of the microcode file. The same naming conventions used for **<microcode file identifier>** are also used for **<program name>**.

<letter>

When not used in the FILE option, **<letter>** causes the RD program to display all program attributes of the codefile that begin with the letter specified.

When used in the FILE option, the RD program displays all the file attributes of the internal file-id of the codefile that begins with the letter specified.

SPAD

Causes the RD program to display the scratch pad.

FILE

The FILE option causes the RD program to display the specified options for the internal program file, **<file-id>**.

ALL

The ALL option can be used only in batch mode and causes the RD program to display all the program attributes of the codefile.

When used in the FILE option, the ALL option causes the RD program to display all the file attributes for the internal file name of the codefile.

Example inputs:

```
QF RD SPAD
QF DMPALL ALL
QF PROGRAM FILE IN ALL
QF PROGRAM M
QF *DMPALL RSZ
```

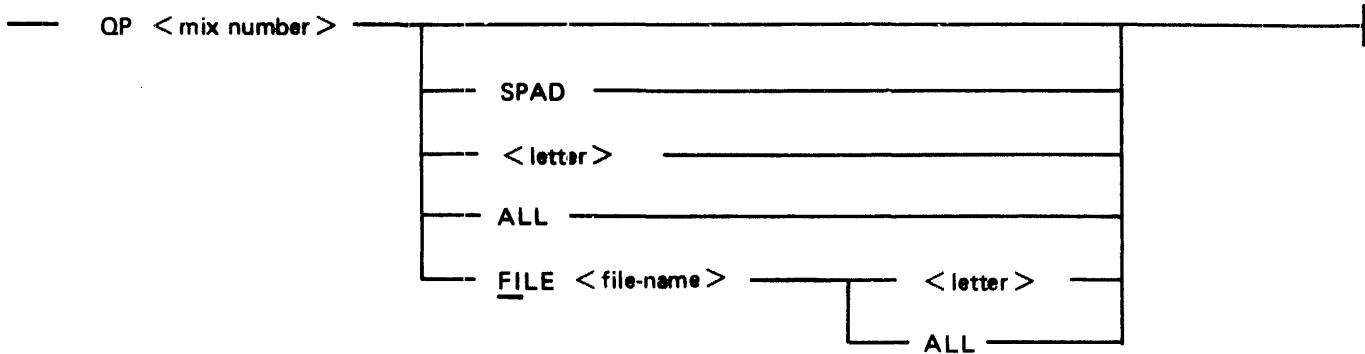
Example output if "QF RD FILE LINE B" is entered:

```
BUFFERS                2
RECORDS PER BLOCK      1
MAX BLOCK SIZE (BYTES) 120
BLOCKSIZE              0
BLOCKS PER AREA        100
BACKUP                 3 (DISK OR TAPE)
BACKUP OK              1
BACKUP ALREADY         0
BACKUP FILE EXTENSION  0
```

QP (Query Program)

The QP command causes the RD program to query the attributes of a program currently running in the mix.

Syntax:



Semantics:

<letter>

When not used in the FILE option, <letter> causes the RD program to display all program attributes of the program that begin with the letter specified.

When used in the FILE option, <letter> causes the RD program to display all the file attributes of the internal file name of the program that begin with the letter specified.

SPAD

Causes the RD program to display the scratch pad.

FILE

The FILE option causes the RD program to display the specified options for the internal program file, <file name>.

ALL

The ALL option can be used only in batch mode and causes the RD program to display all the program attributes of the program.

When used in the FILE option, the ALL option causes the RD program to display all the file attributes for the internal file name of the program.

Example inputs:

```
QP 12 SPAD
QP 80 FILE LINE H
QP 99 ALL
QP 578 FILE CARDS ALL
```

Example output:

Refer to the QF command for an example of QP output.

TEACH (See HELP Command)

TD (Time and Date)

The TD command causes the RD program to display the current system time (hh:mm:ss:t) and date (day of the week and Julian date).

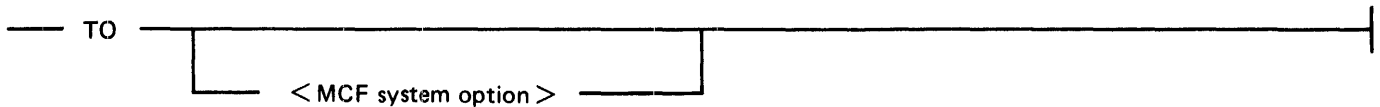
Example input and output:

Input	Output
TD	TIME = 13:37:53.2 (01:37:53.2 PM) DATE = WEDNESDAY, 2/6/80 (80037)

TO (Test Option)

The TO command displays the settings of the MCP system options or, if followed by an option name, the setting of that option.

Syntax:



Semantics:

<MCP system option>

Specifies a valid MCP system option. All MCP system options are explained in the *B 1000 Systems System Software Operation Guide, Volume 1*.

Example inputs:

```
TO  
TO RMOV
```

Example output if TO is entered:

```
AMCS= 1 BOJ = 1 BREL= 1 BRGR= 0 CHR= 0 CLOS= 0 COPY= 1 DATE= 1  
DEBUG= 1 DISP= 1 DUMP= 1 EOJ = 1 FLMP= 0 LAB = 1 LIB = 1 LOG = 1  
MEM = 0 MPRI= 1 ODTL= 1 OPEN= 1 PBD = 1 PBT = 0 RFAC= 0 RMOV= 1  
RMSG= 0 SCHM= 0 SQRM= 0 TERM= 0 THR = 0 TIME= 1 TOUT= 0 TRMD= 1  
VLCP= 0 VLIO= 0 ZIP = 1
```

TRANSLATE

Each input of the RD command TRANSLATE acts as a toggle to set or reset the translate option. When the translate option is set (the default condition) all lower-case letters in the RD program commands are converted to upper case.

Syntax:

— TRANSLATE —————

Pragmatics:

The TRANSLATE command is not valid if entered by way of the PASS command to the SMCS program. The SMCS program has a TRANSLATE command that can be used instead.

Example output:

TRANSLATE	NOW ON
TRA	NOW OFF

UF (User of File)

The UF command causes the RD program to display the users of a specified disk file.

Standard B 1000 system file identifiers are used, including usercodes and the asterisk (*) character. The usercode default family name is applied to file identifiers when a family name or asterisk (*) character is not specified.

Syntax:

UF < file identifier > < skip/page >

Example input:

```
UF  USERPACK/(USER)/PROG1
U   *INPUT          SKIP 22
UF  SDL2
```

Example output:

```
* NAME *   TYPE  RSZ R.B  B.A  S.A  AREAS      EOF CREATN UPDATE
-----
*SDL2      SD20  180  10  212 2120   1/ 1    2113 110983 011784

5579 SYSTEM/COPY -----> WTG I/O      FILE 0 :SOURCE
5583 SDL2:SYSTEM/COPY -----> EXECUTING  CODE FILE
END UF.
```

```
* NAME *   TYPE  RSZ R.B  B.A  S.A  AREAS      EOF CREATN UPDATE
-----
OOF1E1     IS.D   80   9   500 2500   3/ 25    20 110983 NODATE

 112 ISAM/UPDATE -----> WTG I/O      FILE 2 : UPDATE
 115 DMPALL -----> EXECUTING  FILE 0 : INPFILE
END UF.
```

```
* NAME *   TYPE  RSZ R.B  B.A  S.A  AREAS      EOF CREATN UPDATE
-----
INDEX_S    DATA  180   1   9    9    4/ 20    36 051884 051884

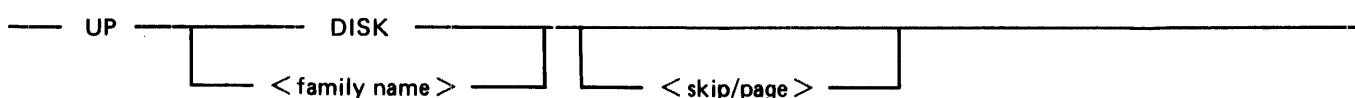
8779 ONLINE/UPDATE -----> WAIT      DB:INVENTORY ON DB
END UF.
```

UP (User of Pack)

The UP command causes the RD program to display the files that are in use on a specified disk.

Specifying DISK as the family name causes the RD program to display the files that are in use on the system disk.

Syntax:



Semantics:

DISK

Specifies the system disk.

Example inputs:

```
UP DISK SKIP 10
UP USERPACK
```

Example output:

UNIT	TYPE	PCU	STATUS						
DPH	DPC1	7A1	USERPACK	(U)	SER.#612344	6	USERS	RETRY	COUNT 2
(USER)/PROG1		1	(USER)/INPUT1		1	(USER)/LINE		1	
DMPALL		1	REMOTE/LISTER		2				
END UP.									

US (User)

The US command followed by a valid <usercode>/<password> logs a user on to the RD program if the station is attached to the RD program. While a user is logged on, the default family name and the usercode are applied to all file identifier references. By entering the US command followed by OFF, a user can log off without detaching the station from the RD program. Following this entry, the default family name and the usercode are not applied to file identifier references until the user again logs on.

Entering the asterisk (*) character before any file identifier overrides the default family name and the <usercode>.

When a user is logged on with a usercode, entry of the PD, KA, DF, or BF commands assumes a file name parameter of <usercode>/= even if the file identifier is omitted.

Syntax:



Semantics:

<usercode>/<password>

Specifies a valid usercode and password from the (SYSTEM)/USERCODE file.

OFF

Logs the user off the RD program.

Example inputs:

US DOE/JOHN

US OFF

USE (Usage Information)

The USE command causes the RD program to display four statistics:

ON.LINE

The number of users currently using the program or the OMX (Off-Line Mix) function.

WAIT.TIME

The time in seconds between updates of the mix table displayed by the OMX function.

INVALID

The count, for all commands, of failed attempts since BOJ of the RD program.

All other

The count, for each command, of successful attempts since BOJ of the RD program.

Example output:

INVALID	. . .	1	AVL	. . .	0
BF	. . .	0	BRK	. . .	1
BYE	. . .	9	CP	. . .	6
DF	. . .	18	DIR	. . .	5
FN	. . .	5	FRZ	. . .	0
ID	. . .	0	JOB	. . .	21
KA	. . .	25	KB	. . .	29
KC	. . .	0	KP	. . .	0
MIX	. . .	20	OL	. . .	15
OMX	. . .	10	PD	. . .	52
PQ	. . .	0	PRT	. . .	0
QF	. . .	9	QP	. . .	0
TO	. . .	0	TRANSLATE	. . .	0
US	. . .	4	USAGES	. . .	4
WAIT	. . .	0	WHO	. . .	0
WS	. . .	0	ON.LINE	. . .	0
WAIT.TIME	. . .	20			

W (Wait)

The W (Wait) command causes the RD program to change the length of time between each memory scan for an update to the mix table. The W command is valid only when there are fewer than two users signed on to the RD program or using the OMX facility. When there are two or more users, the following equation is used by the RD program to calculate the wait time:

$$W = U + J$$

where W is the wait time in seconds, U is the number of users, and J is the number of jobs in the mix.

The maximum value of (number of jobs in the mix) is 22. The wait time defaults to 10 seconds when there are fewer than two users.

The W command has no effect if program switch 1 is non-zero.

Syntax:

— W <integer> —————

Semantics:

<integer>

Specifies the number of seconds to wait before scanning memory for a update to the mix table. Valid for less than two users signed on to the RD program or using the OMX facility. There can be no blank space between the letter W and the integer.

WHO (Current User Info)

The WHO command causes the RD program to display the usercode of the current user logged on to the RD program. Included are the parameters of the usercode maintained in the (SYSTEM)/USERCODE file for the user with the exception of the password. This command is only valid when a user is logged on to the RD program.

Example output:

```
INDEX = 66, USERCODE = "(RD)", PACK.ID = "MCP", CHARGE = 7333, PRIORITY  
= 15, PRIVILEGED = 1, PUBLIC = 1, OVERRIDE = 0.
```

WS (Waiting Schedule)

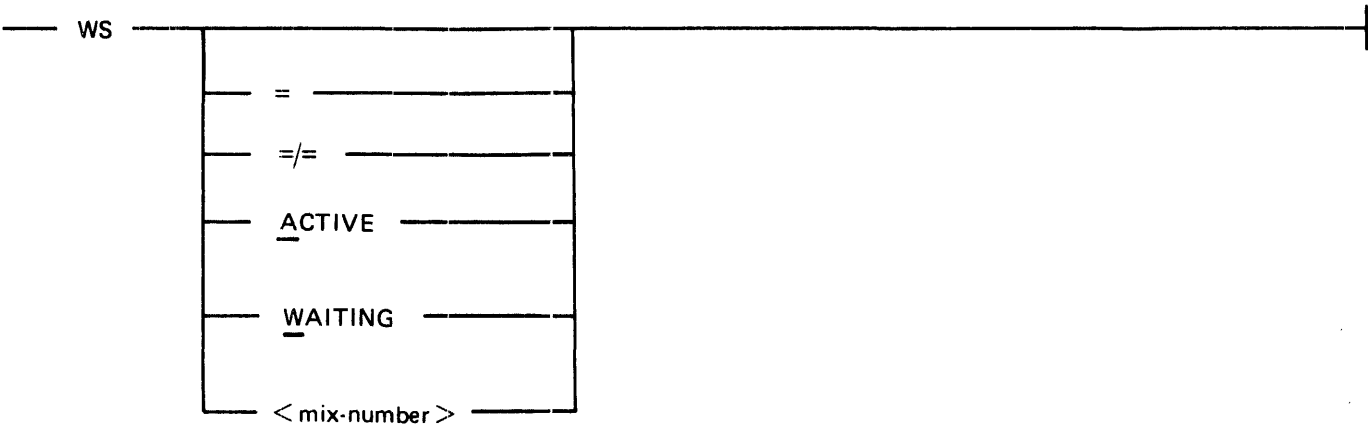
The WS command causes the RD program to display the names of all programs in the waiting and active schedules. The equal sign (=) character is optional. If a number is specified in the WS command, the RD program displays the jobs in the waiting and active schedules beginning with the nth scheduled job, where n is the number specified. For jobs in the waiting schedule, the RD program also displays the event for which the program is waiting. The following information is provided for each job in the waiting schedule.

Heading	Meaning
MIX#	Job Number
PR	Scheduled Priority
SSN	Session Number
#KB	Number of Bytes (in thousands)
	Required to Run
IN FOR	Elapsed Time Since Schedule Time

The name of the program is displayed for each job in the waiting schedule. If the program was executed with a usercode, the usercode is enclosed in parentheses and displayed prior to the program name. If the program is a compile, the resulting object program name is displayed.

The waiting event is displayed for all programs. If the program is waiting unconditionally on a job, the letters UC are displayed after the waiting event.

Syntax:



Semantics:

=, =/=

Causes RD to display all the jobs in the active and waiting schedules.

ACTIVE

Causes RD to display all the jobs in the active schedule.

WAITING

Causes RD to display all the jobs in the waiting schedule

Example inputs:

```
WS
WS =
WS 12
WS ACTIVE
WS A
WS WAITING
WS W
```

Example output:

```
MIX# PR  SSN #KB  IN FOR
ACTIVE SCHEDULE EMPTY.
6757 04 0041 011      11.8 DMPALL,WTG FS
6758 04 0041 008      11.0 RL,WTG #6757
```

ODT COMMANDS

The ODT commands described in the following paragraphs may be entered only from the the system ODT. The following is the format of each command.

<RD program mix number>AX<command>

The following commands are valid from the system ODT and perform the same functions as they do from a terminal.

EOJ	PQ (Print Queue)
FN (File Name)	TEACH
FRZ (Freeze)	TO (Test Option)
HELP	USE
ID	W (Wait)

PRG (Programs)

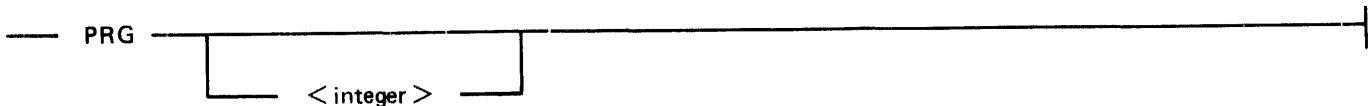
The PRG command causes the RD program to interrogate or assign the maximum number of programs that can be displayed in the program mix table. The value specified in the PRG command overrides the value assigned by program switch 3.

The PRG command is valid only from the system ODT and is described in the following paragraphs.

The value for the PRG command must be between 2 and 22, inclusive.

If the value was changed by the PRG command and if the RD program goes to end of job, then the program mix table limit defaults to the value assigned by program switch 3.

Syntax:



Semantics:

<integer>

Assigns the maximum number of programs that are to be displayed in the program mix table. The default is 16.

Example inputs and outputs:

Input	Output
PRG 3	PROGRAM LIMIT CHANGED TO 3
PRG 16	PROGRAM LIMIT CHANGED TO 16

BATCH COMMANDS

The RD program can be executed in a batch mode by setting program switch 0 to 1 or 2 in the MCP EXECUTE command. If program switch 0 is equal to 1, the RD program accepts input commands from the system ODT. If program switch 0 is equal to 2, the RD program reads from a card file labeled CARDS. A BYE or EOJ command sends the RD program to end of job.

The following commands are valid in batch mode and function the same as commands from a terminal.

BF	FN	KC	QP	US
BYE	HELP	KP	TD	USE
DF	ID	OL	TEACH	WHO
DIR	KA	PD	TO	
EOJ	KB	QF	TRANSLATE	

The CLS and PAGE commands are described in the following paragraphs.

CLS (Close)

The CLS command causes the RD program to close the printer file labeled LINE.

Example inputs:

CLS

PAGE

The PAGE command causes the RD program to skip to the top of a new page to print the input and output of the next command.

Example inputs:

PAGE
PAG

SECTION 23

REMOTE/LISTER

INTRODUCTION

The REMOTE/LISTER program displays the contents of disk files on terminals such as the TD 830, and the MT 985/MT 983. The program can display data files and printer backup files, and can also scan printer backup listings of compiler outputs for error messages and warnings.

NOTE

The REMOTE/LISTER program cannot be used to display Relative, Indexed, or Variable Length Record files.

OPERATING INSTRUCTIONS

The REMOTE/LISTER program can be explicitly executed from a terminal, SIGNED ON to through the SMCS, or PASSEd to by means of the SMCS program pass mechanism. The following terminal entry example is a string requesting explicit execution:

```
?EXECUTE REMOTE/LISTER;PRIORITY 9;
```

To interface with the SMCS program, the REMOTE/LISTER program must be put into the SMCS/JOBS file. (For information, see the *B 1000 System Supervisory Message Control System (SMCS) Reference Manual*.)

The following is an example of the SMCS/JOBS entry:

```
$ REMOTE/LISTER      LOG-ON NO-RR NO-SCROLL;  
  EXECUTE REMOTE/LISTER;  PROTECTED;  PR 9;
```

This allows the user to SIGN ON or PASS to the REMOTE/LISTER program using the SMCS sign-on and program-pass mechanisms.

PROGRAM SWITCHES

The following program switches are used in the REMOTE/LISTER program:

Program Switch 0

If program switch 0 is set, the REMOTE/LISTER program enters a batch mode, using the MCP ACCEPT mechanism for ODT input. The only valid commands in batch mode are PRINT and BYE (or EOJ). Batch mode allows the printing of files even if the data communications network is not running.

Program Switch 1

If program switch 1 is set, the heading information (the label record for backup files) line and the EOF/BOF or END OF FILE generated lines are omitted.

Program Switch 3

If program switch 3 is set, wrap-around from the input file is inhibited until a prompt is given. If program switch 1 is reset (0), the generated end line displays END OF FILE instead of EOF/BOF. If program switch 3 is reset (0), wrap-around is allowed.

Program Switch 4

If program switch 4 is set to a nonzero value, the REMOTE/LISTER program waits that number of minutes between commands before displaying a message indicating that it will go to end of job. If the user does not respond in one minute, the REMOTE/LISTER program goes to end of job.

Program Switch 7

Program switch 7 is used to indicate whether the MCS header TOG[1] should be set to 0 or 1. If program switch 7 is reset, TOG[1] is set to 0; otherwise it is set to 1. Program switch 7 should be reset as long as the CANDE NDL request and control sets are being used in the network controller. If the CANDE request and control sets are not being used, the program switch should be set. Setting TOG[1] causes output messages from REMOTE/LISTER to not be scrolled on the terminal.

Program Switch 8

Program switch 8 is used to indicate whether the PRINT command should assume that a 6 lines per inch or 8 lines per inch printer carriage control tape is in use. If program switch 8 is reset, 6 lines per inch is assumed; otherwise, 8 lines per inch is assumed.

COMMAND SYNTAX

The REMOTE/LISTER program remembers the last command entered if it was a BACK, NEXT, SEARCH, ERROR, WARNING, FIND, minus (-), or plus (+) command, and a blank input causes the same command to resume at the point the previous output page ended. Any non-blank entry terminates the current command. A PAGE END command is continued as a BACK command.

Any input can be entered during a SEARCH, FIND, ERROR, or WARNING command as a means of requesting a break in the routine that is in progress. Invalid input is ignored instead of causing an error.

Commands may be entered in lower or upper case, but file names must be exact. (They are not translated to upper case.)

Several commands may be entered in one input provided that none causes an output.

If the SMCS program PASS mechanism is used to PASS to the REMOTE/LISTER program, the signal character, the token PASS, and the program name are displayed in the upper left corner of the screen with each output. For example, if the REMOTE/LISTER program is renamed as RL, the following is displayed in the upper left corner after each output:

*PASS RL

Non-graphic characters are converted to the question mark character (?) by the REMOTE/LISTER program. The hexadecimal representations of these commands can be displayed by using the HEX command.

A workfile must be present for any of the following commands to be accepted by the REMOTE/LISTER program:

BACK	HEX	NEXT	SAME	WAIT
ERROR	MARGIN	PAGE	SEARCH	WARNING
FIND	MCP	REMOVE	SEQUENCE	

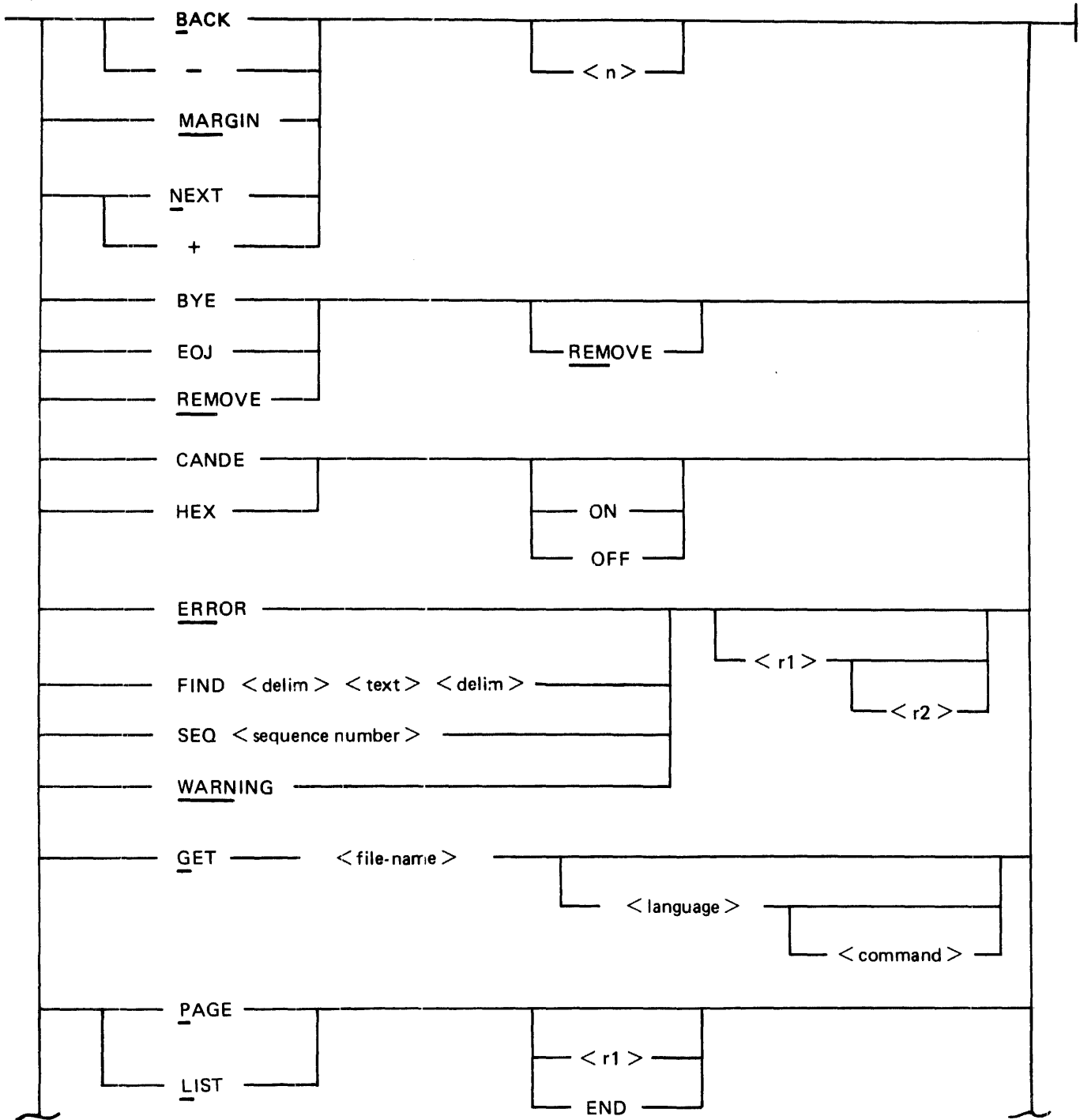
A workfile may be, but need not be, present for any of the following commands:

BYE	GET	TEACH	USER
CANDE	PRINT	TERMINAL	WHAT

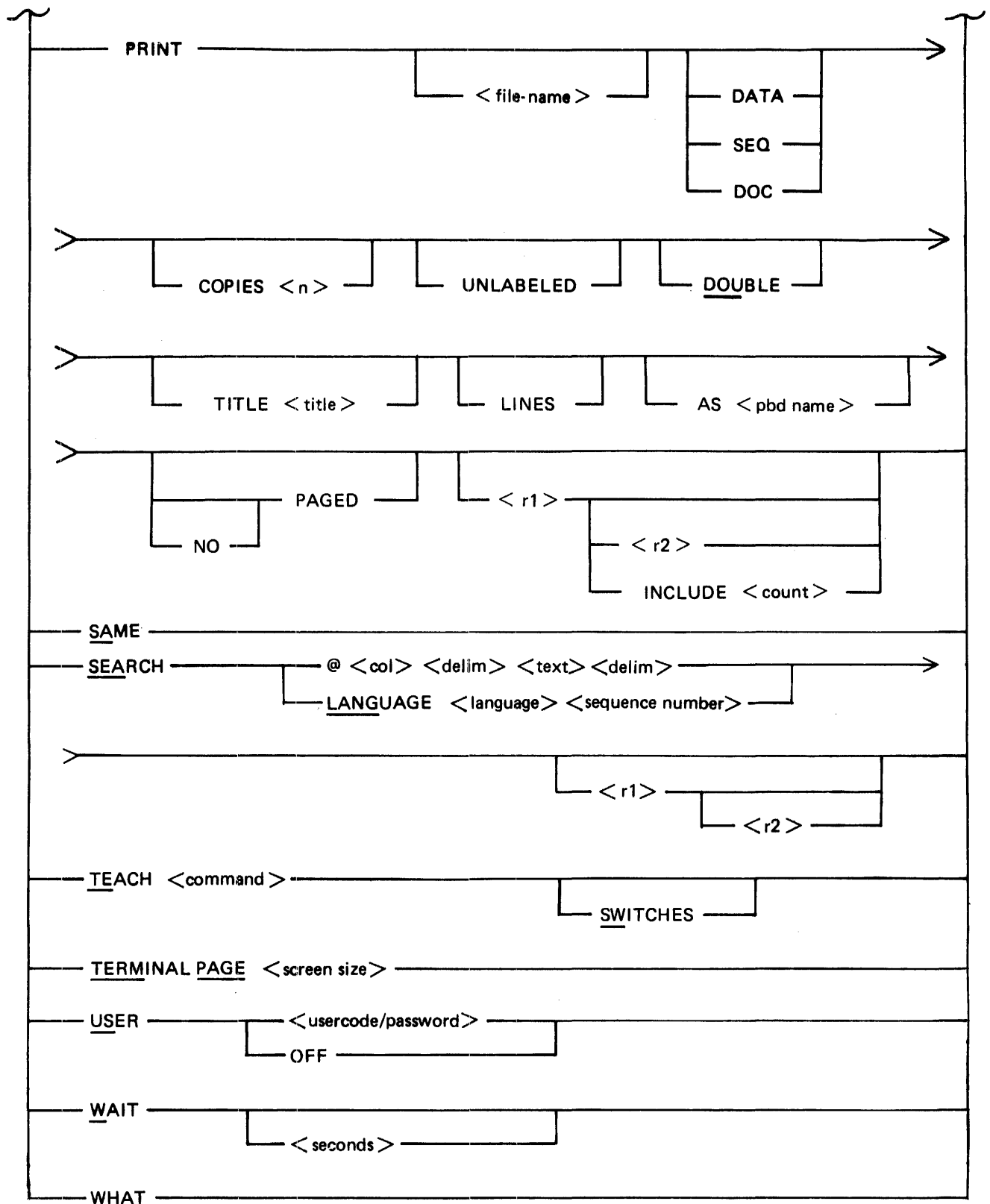
Program Commands

An overall syntax diagram of the REMOTE/LISTER program commands follows. The commands are grouped for clarity and a compact presentation. Command semantics, in alphabetical order, follow the diagram.

Overall REMOTE/LISTER Syntax:



B 1000 Systems System Software Operation Guide, Vol. 2
 REMOTE/LISTER



BACK

The BACK command causes the REMOTE/LISTER program to page <n> screen-lengths backward in the input file and to display the resulting page. Entry of the command without an <n> value pages one screen-length backward. The command may only be specified if a workfile is present. Blank inputs cause the BACK command to continue; any other input terminates the command.

Syntax:



Semantics:

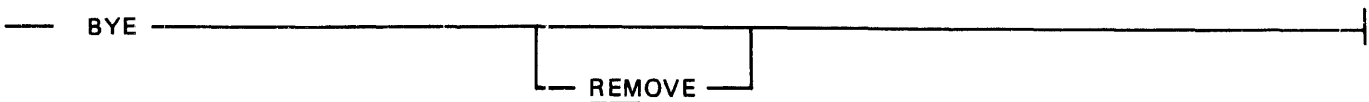
<n>

An integer value representing the number of pages to space back.

BYE

The BYE command signs the terminal off the REMOTE/LISTER program and causes the program to go to end of job. If the REMOVE option is used, the file is removed if it is a backup file.

Syntax:



Semantics:

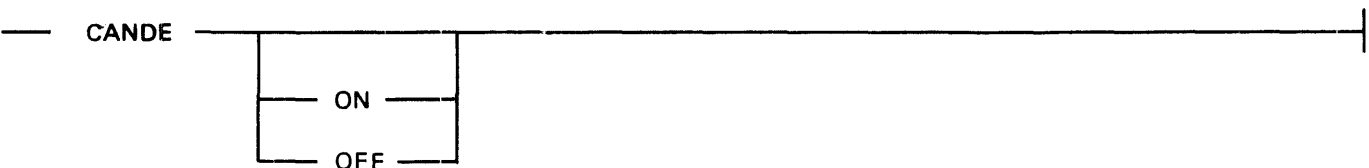
REMOVE

Causes the file to be removed if it is a backup file.

CANDE

The CANDE command causes the REMOTE/LISTER program to reformat the screen as a CANDE screen, with 8-digit sequence numbers on the left. Entering CANDE with no options is the same as entering the CANDE command with the ON option.

Syntax:



Semantics:

ON

Turns the CANDE command on.

OFF

Turns the CANDE command off.

EOJ

The EOJ command causes a Remote Lister program running in batch mode to go to end of job. The EOJ command is only valid in batch mode. If the REMOVE option is used, the file is removed if it is a backup file.

Syntax:



Semantics:

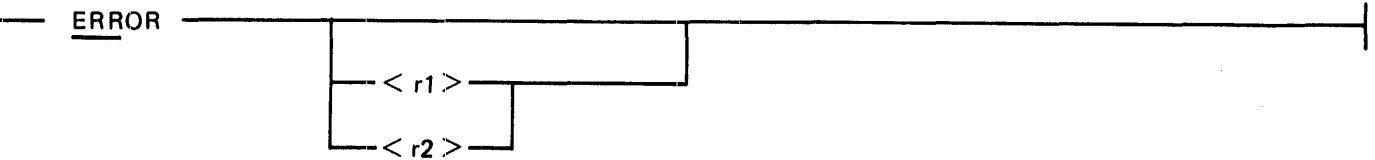
REMOVE

Causes the file to be removed if it is a backup file.

ERROR

The ERROR command searches a printer backup compile listing for error messages. Unless the file is an SDL or UPL file, it must have been loaded with the appropriate language specified. Each error message and the five preceding records are displayed. Blank inputs cause the command to continue; all other inputs cause it to terminate. The ERROR command can only be specified if a workfile is present.

Syntax:



Semantics:

< r1 >

An integer representing the starting record (not sequence) number. at which the search is to begin.

< r2 >

An integer representing the ending record number at which the search is to end.

FIND

The FIND command searches the file for a token string. Blank inputs cause the command to continue; and other input terminates the command. The FIND command can only be specified if a workfile is present.

Syntax:



Semantics:

<delim>

Any nonblank character except an alphanumeric, period (.), or underscore (__) that does not appear in the text.

<text>

The target text to be sought. It can be up to 40 characters in length.

<r1>

An integer representing the starting record number (not sequence number), at which the search begins.

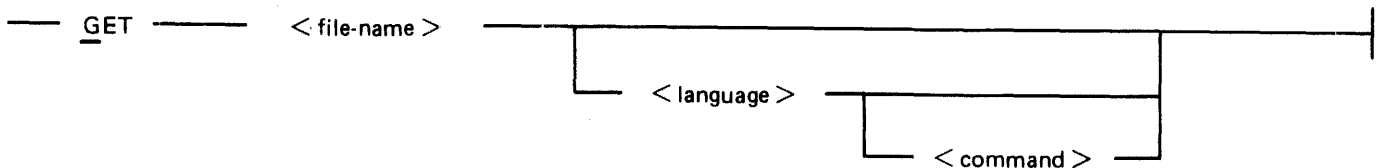
<r2>

An integer representing the ending record number, at which the search ends.

GET

The GET command begins the display of a file. A file of language type may be specified. If it is, a REMOTE/LISTER program command may also be specified. The GET command causes the REMOTE/LISTER program to begin the display with the first page of the file; but the specified command may override this. The default language type is SDL/UPL. The GET command sets the MARGIN parameter to 1 unless MARGIN with some other value is specified as the <command>. The GET sets the WAIT parameter to 0. (See MARGIN and WAIT, later in this section.)

Syntax:



Semantics:

<file name>

Specifies the name of the disk file to be displayed.

② WFL JOBS
7

<language>

Specifies the compiler or program that produced the printer backup file. The following names are valid:

BASIC	DASDL	DOCTOR	MIL	PASCAL	SDL
COBOL	DATA	FORTRAN	NDL	RPG	SDL2
COBOL74	DMPALL	FORTRAN77	<u>1</u> NONE	SDL	UPL UPL2

The first three characters of the name are acceptable abbreviations for all the languages except COBOL74 (C74) and FORTRAN77 (F77).

If a file that is to be loaded is a program source and <language> is not specified, the <language> is taken from the filekind attribute in the disk file header. If <language> is specified, the <language> specified overrides the filekind attribute in the disk file header.

If the file that is to be loaded is a printer backup file, the REMOTE/LISTER program attempts to assign a <language> from the label record of the printer backup file. If <language> is specified, the <language> specified overrides the <language> found in the label record.

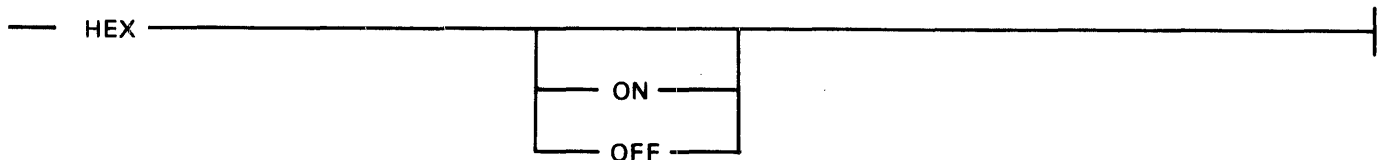
<command>

The <command> option is any valid REMOTE/LISTER command or sequence of commands and can only be specified if <language> has been specified.

HEX

The HEX option, when set (ON), causes the REMOTE/LISTER program to display output records in hexadecimal format. The HEX command can only be specified if a workfile is present. This option is reset (OFF) each time a new file is loaded using the GET command.

Syntax:



② ~~OFF~~
RIC2527

MARGIN

The MARGIN command causes the REMOTE/LISTER program to set the left margin at a column specified by the value of <n>. The resulting display includes 80 characters of data from each record, beginning at the column specified by <n>. If MARGIN 0 is entered, entire records are displayed.

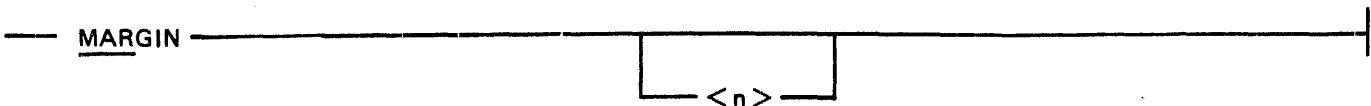
The MARGIN command may be specified only when a workfile is present.

For printer backup files that are loaded with a language specified, the margin defaults to the location of the sequence numbers. For non-backup files, the margin defaults to 1.

Each time the GET command is performed, the margin is set to 1 unless otherwise specified in the GET command.

Each entry of the MARGIN command resets the CANDE command.

Syntax:



Semantics:

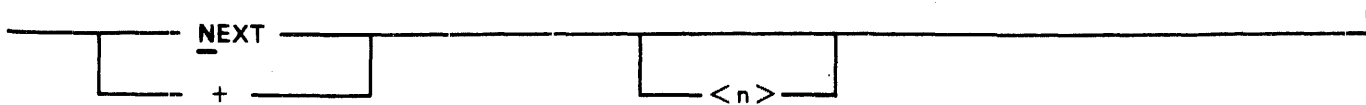
<n>

An integer value that specifies the margin setting.

NEXT

The NEXT command (or the + character) causes the REMOTE/LISTER program to page the input <n> screen-lengths forward and to display the resulting page. Entry of the command without an <n> value pages one screen-length forward. Blank inputs cause the command to continue; all other inputs cause it to terminate.

Syntax:



Semantics:

<n>

An integer value representing the number of pages to space forward.

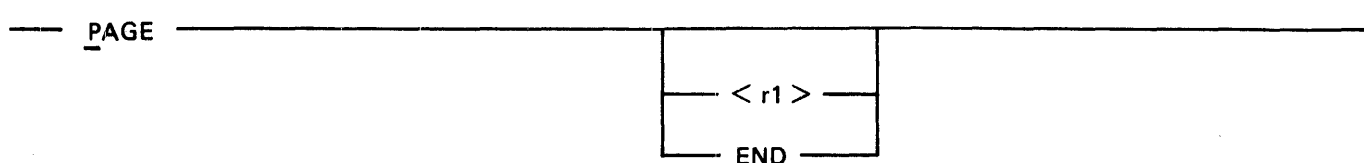
PAGE

The PAGE (or LIST) command causes the REMOTE/LISTER program to begin the display of a file at the record specified or at the first record if no record is specified, and to initiate a NEXT function.

If the PAGE END option is used, the REMOTE/LISTER program causes the last page of the file to be displayed and initiates a BACK function.

The PAGE command only may be specified if a workfile is present.

Syntax:



Semantics:

<r1>

This integer value causes the display to begin at record <r1>. (Note that r1 is not a sequence number.)

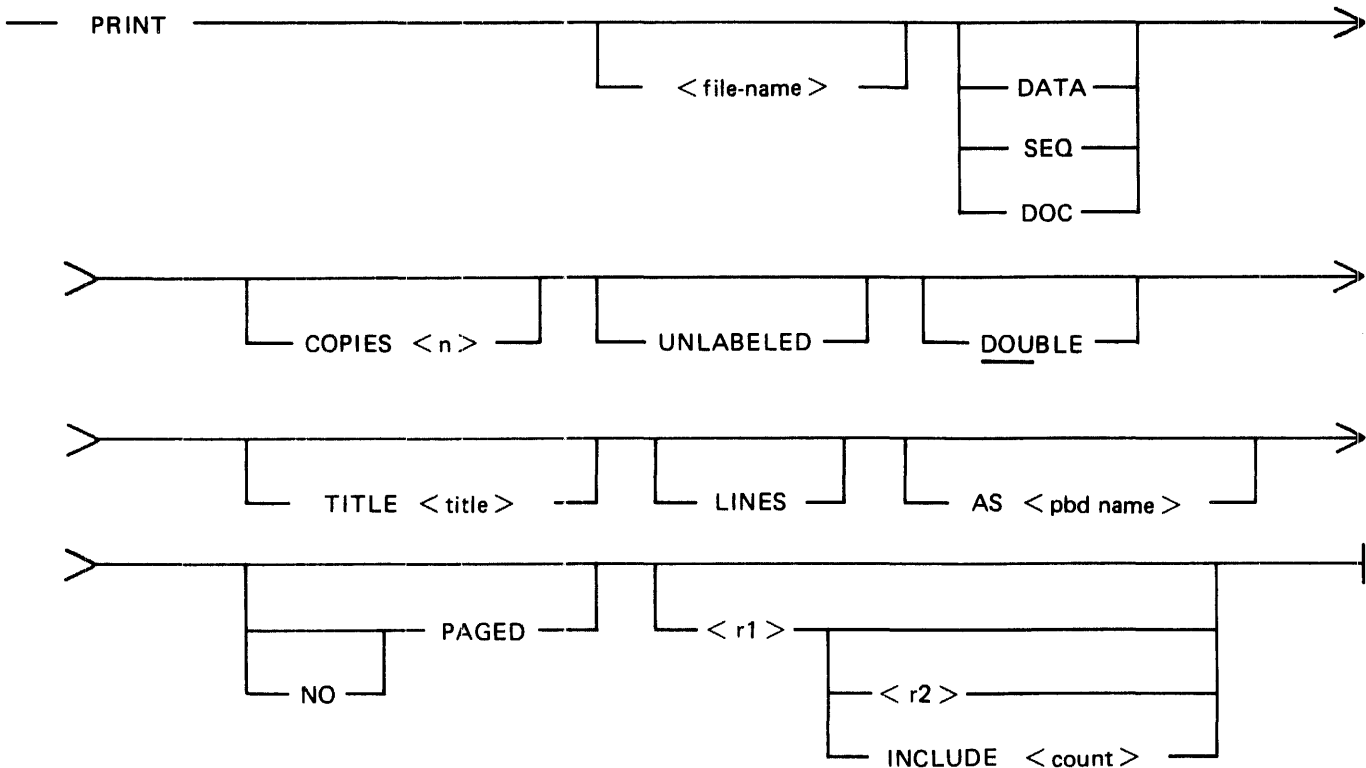
END

This option causes the last page to be displayed.

PRINT

The PRINT command causes the current input file or a specified file to be printed on a line printer. If the HEX option is on, the PRINT command causes the file to be printed in hexadecimal format.

Syntax:



Semantics:

`<file-name>`

The `<file-name>` option specifies the file name to be printed. If it is not specified, the current input file is printed. The `<file-name>` option may be specified only when there is no current input file.

DATA

The DATA option causes the file to be printed with no record numbers. The DATA option is the default for printer backup files.

SEQ

The SEQ option causes the file to be printed in the following format: 6-digit record number, blanks, 80 characters of data, 10-character patch id, blanks, and the 10-character file id. The SEQ option is the default for data files.

DOC

The DOC option causes the file to be printed in the following format: 6-digit record number, blanks, 72 characters of data, blanks, an 8-digit sequence number, blanks, 10-character patch id, blanks, 10-character file id.

COPIES <n>

The COPIES options causes <n> copies to be printed. The default is one copy.

UNLABELED

The UNLABELED option produces an unlabeled printer backup file. The default is a labeled printer backup file.

DOUBLE

The DOUBLE option produces a double-spaced listing. The default is single-spacing.

TITLE <title>

If the PAGED option is specified, the TITLE option causes <title>, a string of up to 60 characters, to be printed at the top of each page except the first.

LINES

The LINES option, valid only for the DOC output format, causes reference numbers to be printed for every line.

AS <pbid name>

The AS option creates a user-named backup file labeled <pbid name>.

PAGED

The PAGED option causes page numbers to be printed on every page except the first. If this option is set, any of the tokens & PAGE, &PAGE, & page, or &page in columns 1 through 6 of any input record causes the REMOTE/LISTER program to skip to the next page. PAGED is default for the DOC printing option, but may be specified for any printing option.

NO

The NO option turns off the PAGED option for files being printed with the DOC printing option.

<r1>

The <r1> option causes the REMOTE/LISTER program to begin printing at the record specified by <r1>. The default is the first record.

<r2>

The <r2> option causes the REMOTE/LISTER program to stop printing at the record specified by <r2>. The default is the end of file.

INCLUDE <count>

The INCLUDE option causes the REMOTE/LISTER program to print <count> number of records. The default is printing to the end of file.

REMOVE

The REMOVE command closes the current input file and allows switching to a new file. The repetition of the REMOVE keyword causes the purge of the file if it is a backup file. The REMOVE command can only be specified if a workfile is present.

Syntax:



Semantics:

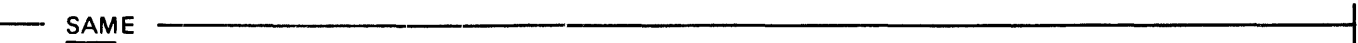
REMOVE

The REMOVE option causes the purge of the input file if it is a backup file.

SAME

The SAME command causes the same page to be displayed on the terminal screen. The SAME command can only be specified if a workfile is present.

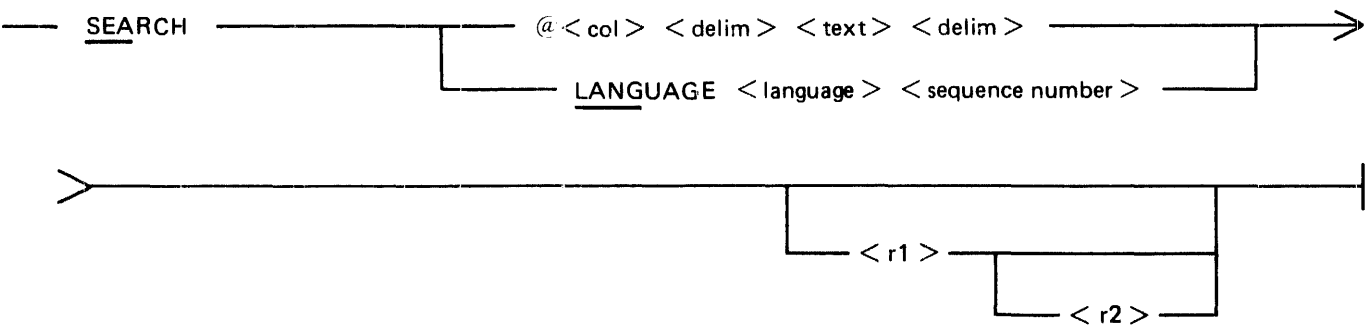
Syntax:



SEARCH

The SEARCH command causes the REMOTE/LISTER program to search the input file for a text string. The SEARCH command can only be specified if a workfile is present.

Syntax:



Semantics:

@ < col >

The @ < col > option causes the search to begin in column < col > of the file.

<delim>

Any nonblank character except an alphanumeric, period (.), or underscore (__) that does not appear in the text.

<text>

The target text to be sought. It can be up to 40 characters in length.

LANGUAGE <language>

The LANGUAGE option specifies the compiler or program that the the compile listing was produced by. Refer to the GET command for a list of allowable languages.

<sequence number>

The <sequence number> specifies the sequence number to be searched for. An exact match must be made. This is different from the SEQ command which searches for a sequence number that is greater than or equal to the target sequence number.

<r1>

An integer representing the starting record number (not sequence number) where the search is to begin.

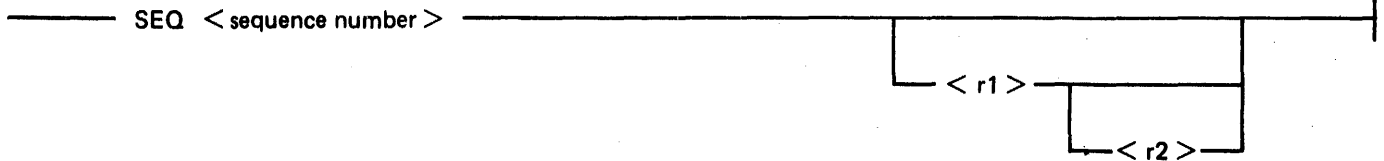
<r2>

An integer representing the ending record number where the search is to end.

SEQ

The SEQ command causes the REMOTE/LISTER program to search for a given sequence number in the file, stopping when it encounters a sequence number that is greater than or equal to the target sequence number. The SEQ command can only be specified if a workfile is present.

Syntax:



Semantics:

<sequence number>

The target sequence number that is to be searched for.

<r1>

An integer representing the starting record number (not sequence number) where the search is to begin.

<r2>

An integer representing the ending record number where the search is to end.

TEACH

The TEACH command lets a user obtain explanations of REMOTE/LISTER commands and program switches.

Syntax:



Semantics:

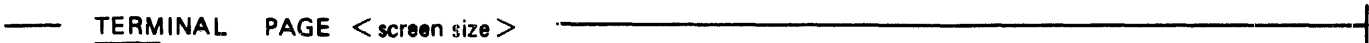
<command>
Specifies any valid REMOTE/LISTER command.

SWITCHES
Requests an explanation of the program switches.

TERMINAL

The TERMINAL command causes the REMOTE/LISTER program to change the screen size of a terminal from the default of 1920 to the value entered. After the new screen size has been entered, the REMOTE/LISTER program asks the user if the correct screen size has been entered.

Syntax:



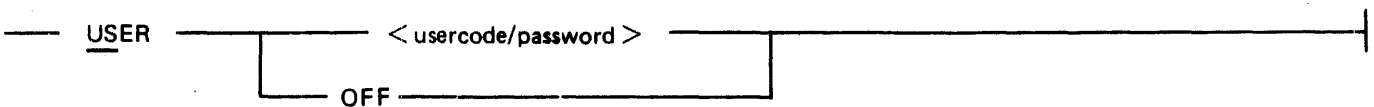
Semantics:

<screen size>
The <screen size> is the size of the display in characters that is to be displayed on the screen. The value of <screen size> must not be greater than 1920 characters.

USER

The USER command allows the user to log on to the REMOTE/LISTER program. The usercode/password combination entered is used on all file opens.

Syntax:



Semantics:

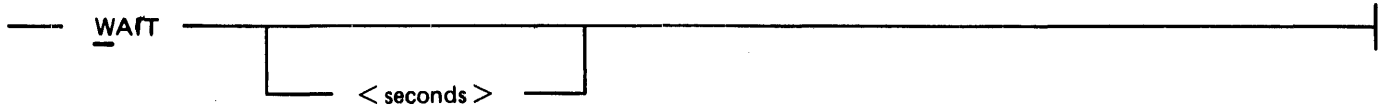
<usercode/password>
Any valid usercode and password.

OFF
The OFF option logs the usercode and password given off the REMOTE/LISTER program.

WAIT

The WAIT command causes the REMOTE/LISTER program to enter a continuous output mode, waiting the specified time between successive displays of each output screen. Entering WAIT 0 or any non-blank input turns the WAIT command off. The WAIT command can only be specified if a workfile is present.

Syntax:



Semantics:

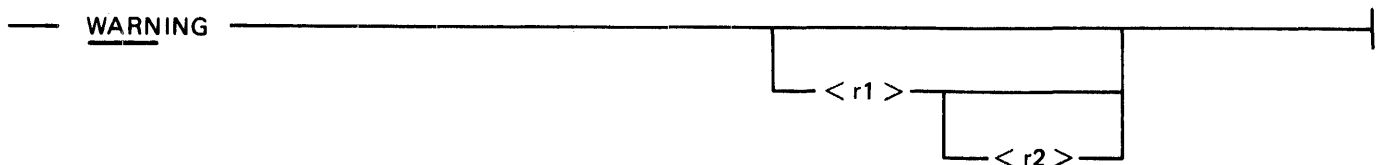
<seconds>

The <seconds> options specifies the number of seconds that the REMOTE/LISTER program waits before displaying the next output screen. The maximum time is 60 seconds and the default is 3 seconds.

WARNING

The WARNING command searches for warning messages in a printer backup compile listing. The file must have been loaded with the appropriate language specified in the GET command. The WARNING command displays five records prior to any error message that is located. The WARNING command is continued by the entry of blank input. Any other input terminates the command. The WARNING command can only be specified if a workfile is present.

Syntax:



Semantics:

<r1>

An integer representing the starting record number (not sequence number) where the search is to begin.

<r2>

An integer representing the ending record number where the search is to end. If <r2> is specified, then <r1> must also be specified.

WHAT

The WHAT command displays the program-level, usercode (if the station is logged on), file information, and option settings of the REMOTE/LISTER program.

Syntax:

— WHAT —————|

SECTION 24

REMOTE/PRINT

The REMOTE/PRINT program reads a B 1000 printer backup file and converts the records to a format that is acceptable to any of a number of types of remote printers as well as the system line printer. The printer backup files must have been created by the Mark 9.0 or later System Software release.

The following documents provide information on the applicable printers.

AP 300 Equipment (Data Communications Interface) Reference Manual, form number 1119153.

Model 630 Printer/Terminals, HPR05 Interface, form number, 901441-01, Rev A. (The Model 630 is the Diablo C version of the AP 1301. This document is available from Diablo Systems, Inc., Box 5003, Hayward, California 94545).

AP 1340 Equipment Capabilities and Features, form number 1137684.

AP 1340 Installation, Operation, and Maintenance Guide, form number 2030268.

B 9246-3/6 Line Printer Operator's Guide, form number 1137221.

B 9949-10 Remote Printer Interface Reference Manual, form number 1136884. (The B 9949 Interface Kit is used to convert the B 9246 system line printer into a remote printer.

The following documents provide information on B 1000 software:

B 1000 Systems System Software Operation Guide, Volume 1, form number 1169000.

B 1000 Work Flow Language Installation, Operation, and Functional Description Manual, form number 1168994.

OVERVIEW AND FEATURES

The REMOTE/PRINT program can interface with the following remote printer types:

AP 310, AP 1301, AP 1340, B 9246-3/6.

(B 9246-3/6 is a series of band printers, referred to as B 9246 throughout this document.)

The program can also interface with any other remote printer that appears functionally identical to any of the above-listed printers. It also interfaces with system (non-remote) line printers.

Program Command Summary

Program commands are entered by means of the MCP ACCEPT mechanism or through the Work Flow Language (WFL) compiler.

Program commands are summarized next. Syntax and details are presented in Program Commands.

DEBUG	Provides debug facilities.
DEVICES	Queries status of program and printers.
DUMP	Produces a program dump.
EOJ	Terminates the program.
HELP	Provides current information, including explanations and syntax.
OK	Resumes printing after a STOP.
PRINT	Initiates printing. (See Print Option Summary.)
SKIP	Allows selection of record sequence for printing.
STATUS	Queries for status of current print request.
STOP	Suspends printing.
TEST	Tests printer functions; prints test pattern.
WHAT	Identical to STATUS.

Print Option Summary

The PRINT program command includes options to enable, specify parameters for, and disable the functions summarized in the following paragraphs. Option syntax, semantics, and examples are presented in PRINT Command Options.

BANNERS	Controls the printing of leading and trailing banners.
CHARSPERINCH	Specifies the characters per inch (horizontal measurement) for the AP 310, AP 1301, and AP 1340 remote printers.
DOUBLESPACE	Controls the double spacing of the output.
EXTRACT	Prints a portion of the file.
FONT	Specifies the font to be used for the AP 310 remote printer.
INDENT	Indents the text on the left with blanks.
LINESPERINCH	Specifies the number of lines per inch (vertical measurement) for the AP 310, AP 1301, and AP 1340 remote printers.
PAGESIZE	Specifies the page size of the printer backup file.
PAGEWIDTH	Specifies the page width of the printer form.
PRINTCOPIES	Specifies the number of copies of the backup file to be printed.
SAVEFILE	Controls the saving or removing of the backup file after printing.
SUPPRESS	Suppresses the carriage controls in the printer backup file.
TOPMARGIN	Specifies the number of blank lines to be printed as a top margin on the first page.
UPPERCASE	Controls the translation of the printed output to upper-case characters.
USER	Allows a usercode to be specified on whose behalf the file is to be opened.

Default Options Summary

The site manager can permanently modify the defaults of certain functions of the REMOTE/PRINT program by means of program switches or file attributes or both. The functions and mechanisms are described in detail later, in Default Options (Attributes and Switches). In the summaries that follow, the mechanisms (PFA for PRINTER file attribute, BFA for BACKUP file attribute, SW for program switch) are indicated in parentheses.

Output Message Blocking

Whether the output messages to the remote printer are to be blocked in a buffer (SW), and the size of the buffer (PFA). (SW, PFA)

Banners

The printing of banners (backup file name, creation date) on pages preceding and following the output if the page width is 132 characters. (BFA)

Page Length and Width

The default page length (lines) and width (characters) of the printer form in tenths of an inch. (BFA)

Save/Purge

The automatic saving or purging of the backup file upon the completion of a print request. (BFA).

Output Device and Remote File Title

The output device for the program, PRINTER if system line printer and REMOTE if remote printer (PFA), and the title of the remote file if the device is REMOTE. (PFA)

Printer Switching

In the case where more than one remote printer is available, the ability to automatically switch to another remote printer if one is busy. (PFA)

Output Translation

The translation of printer and ODT output to upper-case letters. (SW)

Program Termination

Whether the REMOTE/PRINT program is to remain in the mix indefinitely or is to terminate after completing one print request. (SW)

Error Messages

All error messages and messages that require operator intervention are highlighted on the ODT using bright video. Error messages are comprehensive and self explanatory.

Printer Busy Retries

When the REMOTE/PRINT program finds the remote printer busy (all remote printers busy if automatic printer switching is in effect), it retries until a printer becomes available or until 15 minutes elapse.

Backup File Names

The REMOTE/PRINT program uses standard file naming conventions for usercoded files. In standard file names, the slash (/) character does not follow the usercode: For example:

(USER)FILE not (USER)/FILE

If the file name entered is numeric and seven characters or less in length, the REMOTE/PRINT program assumes that it is a system-created backup file of the format: BACKUP/PRT<n>, where <n> is the backup number. The REMOTE/PRINT program searches the backup designated disk (BD) for system-created backup files unless a family name is explicitly entered.

If the REMOTE/PRINT program is executed with a usercode or if a usercode is specified using the USER option and a numeric file name is entered, the REMOTE/PRINT assumes that the file is a system-created backup file of the format: (<usercode>)PRT<n>.

If a usercode is specified in the print request and a family name is not specified, the MCP attempts to open the file on the default user disk assigned to the usercode.

If the file name is not a system-created backup file, any valid file name is accepted.

File Security Considerations

If the REMOTE/PRINT program is executed with a usercode, files are accessed in the default directory for that usercode. A different usercode and password can be specified with the USER option, overriding the usercode and password that REMOTE/PRINT was executed with.

Examples:

EX REMOTE/PRINT;AX PRINT BACKUP

The file, "BACKUP", is on the system disk as *BACKUP.

USER A/B EX REMOTE/PRINT;AX PRINT BACKUP

EX REMOTE/PRINT;AX PRINT BACKUP USER A/B

The file, "BACKUP", is on user A's default pack as (A)BACKUP. The two examples are functionally identical.

USER A/B EX REMOTE/PRINT;AX PRINT BACKUP USER C/D

The file, "BACKUP", is on user C's default pack as (C)BACKUP.

Remote Printer Control Characters

Table 24-1 lists, for each remote printer, the control characters that are used to invoke pertinent control functions. Any remote printer for use with the REMOTE/PRINTER program must be installed by the Burroughs Field Engineer to use the options listed.

In the table, the term User RP implies a remote printer that is functionally identical to an AP 1301, but to which reset, cancel, and format control codes are not sent.

Table 24-1. Remote Printer Control Characters

Printer	Line Feed	Overprint	Channel Skip	New Page	Reset
B 9246	CRLF	CR	VT <chan>	FF CR	none
AP 310	VT <nn>	VT00	n/a	FF	CAN
AP 1301	CRLF	CR	n/a	FF	ESC CR P
AP 1340	LF	CR	n/a	FF	CAN
User RP	CRLF	CR	n/a	FF	none

NOTES

CAN	(EBCDIC @18@)	cancel
CR	(EBCDIC @0D@)	carriage return
ESC	(EBCDIC @27@)	escape
FF	(EBCDIC @0C@)	form feed
LF	(EBCDIC @25@)	line feed
VT	(EBCDIC @0B@)	vertical tab

<nn> Number of lines to be skipped.
 (Unpacked numeric format.)

<chan> Channel number: 1-12.
 (Packed numeric format.)

n/a Not available.

Printer Buffer Sizes

Table 24-2 shows the maximum buffer sizes (suggested network controller buffer sizes) for printers of interest.

Table 24-2. Printer and Network Controller Buffer Sizes

Printer	Buffer Size	Notes
AP 310 DL/1	256	AP 310 DL/1 has 512-character buffer, but only 256 characters can be sent at once.
AP 310 DL/2	2048	A maximum of 2100 characters is supported by the REMOTE/PRINT program.
AP 1301	2688	
AP 1340	2048	
B 9246	950	
User RP	2100	

Channel Skips

Because the AP 300, AP 1301, and AP 1340 remote printers have no carriage control format ribbons, a skip to channel <n>, where <n> is greater than channel 1 (top of form), is converted to a one-line skip.

For the B 9246 remote printer, all channel skips except a skip to channel 12 are performed. When the REMOTE/PRINT program encounters a skip to channel 12 for one of these printers, a one-line skip is performed instead.

NETWORK CONTROLLER INSTALLATION

In order to use a remote printer on a B 1000 system, the network controller must be set up correctly. The installation of a network controller for use with the various remote printers is described next, under the headings Declaration, Terminal, Station, Line, and File. Following these is an example that will compile and execute as a network controller but is primarily intended to be added to an existing network controller.

The variable names used throughout this material are based on an AP 310 remote printer.

Declaration Section

No special declarations need to be made in the Declaration section.

Terminal Section

The standard CANDEPOLTD and CANDESELTD request procedures are used. The CANDEFSLTD output request procedure can be used in place of the CANDESELTD request procedure. Both input and output request procedures must be declared or a network controller run error (180) can occur.

Although transmission numbers are not required, their use is recommended to minimize transmission errors, especially on switched lines, leased lines, or lines of poor quality.

NOTE

The standard CANDE request and control procedures use alternating 0 and 1 transmission numbers for remote printer devices. (In previous releases, modulo 10 transmission numbers were used.)

The terminal types are set as follows:

Remote Printer	Terminal Type
B 9246	70
AP 310	71
AP 1301	72
AP 1340	73
User RP	79

If the output to the remote printer is to be blocked, the BUFFERSIZE parameter must be declared equal to the maximum buffer size in which the messages are to be blocked, plus 50 characters for the message header. Table 24-2, earlier, provides buffer sizes for these printers.

Example:

TERMINAL AP310:

```
TRANSMISSION = 1.  
ADDRESS      = 2.  
TYPE        = 71.  
BUFFERSIZE  = 2000.  
REQUEST     = CANDEPOLTD: RECEIVE,  
              CANDESELTD: TRANSMIT.
```

Station Section

The NDL MYUSE attribute must be set to INPUT, OUTPUT. The address is set to the address of the remote printer.

Example:

```
STATION APTERYX:
TERMINAL      = AP310.
RETRY         = 10.
FREQUENCY     = 0,0.
MYUSE         = INPUT, OUTPUT.
ADDRESS       = "AP".
```

Line Section

The remote printer can be declared on a line with other terminals using the same line discipline. The line address is set to the address of the line containing the remote printer.

Example:

```
LINE APLINE:

ADDRESS       = 2:0:2.
CONTROL       = CANDETDCTL.
STATION       = APTERYX.
```

File Section

The remote file for the remote printer must be declared separately from the SMCS remote file, MCSREMOTE. Only one remote printer can be declared per remote file.

The TITLE attribute of the REMOTE/PRINT file PRINTER should be modified to have the title of the remote file declared in the network controller.

Example:

```
FILE APFILE:

FAMILY        = APTERYX.
```

Example Network Controller

TERMINAL AP310:

TRANSMISSION = 1.
ADDRESS = 2.
TYPE = 71.
BUFFERSIZE = 2000.
REQUEST = CANDEPOLTD: RECEIVE,
CANDESELTD: TRANSMIT.

STATION APTERYX:

TERMINAL = AP310.
RETRY = 10.
FREQUENCY = 0,0.
MYUSE = INPUT, OUTPUT.
ADDRESS = "AP".

LINE APLINE:

ADDRESS = 2:0:2.
CONTROL = CANDETDCTL.
STATION = APTERYX,
PLATYPUS.

FILE APFILE1:

FAMILY = APTERYX.

FILE APFILE2:

FAMILY = PLATYPUS.

DEFAULT OPTIONS (ATTRIBUTES AND SWITCHES)

Various program defaults are controlled by modifying the file attributes of the the REMOTE/PRINT files BACKUP and PRINTER and by the setting of program switches. In the text that follows, control card attribute names are used. (The Burroughs standard attribute name is included in parentheses with the first use of a control card name.)

Output Message Blocking

The blocking of output messages to the remote printer is controlled by program switch 1 (SW1) and the RECORD.SIZE (MAXRECSIZE) file attribute of the REMOTE/PRINT file PRINTER. Blocking of output messages is not valid for the system line printer.

The desirability of blocking the output messages depends on various factors, including remote printer type, line type, line speed, and number of terminals on the line. Blocking can be advantageous. For example, if the remote printer is on a line that is used by other terminals.

If program switch 1 is set (SW1 = 1), output messages are blocked in a buffer equal to the MAXRECSIZE file attribute for the file PRINTER plus 50 characters for the message header, as follows:

MAXRECSIZE = printer buffer size = 50;

For example, if MAXRECSIZE = is 1950 characters, the printer buffer size must be 2000 characters.

If SW1 = 0, output messages are not blocked.

Examples:

```
WFL MODIFY REMOTE/PRINT SW1 = FALSE;  
MODIFY REMOTE/PRINT FILE PRINTER (MAXRECSIZE = 2000);  
Output messages are blocked in a 1950-character buffer.
```

```
WFL MODIFY REMOTE/PRINT SW1 = TRUE;  
Output message blocking is disabled.
```

Notes

If the specified printer buffer size is too large for a given type of remote printer or for the network controller buffer size for the remote printer, a warning message is displayed and the buffer size is changed to an acceptable value.

The REMOTE/PRINT program cannot differentiate between an AP 310 DL/1 with a maximum buffer size of 256 and an AP 310 DL/2 with a maximum buffer size of 2048. Consequently, specifying a printer buffer size greater than 256 for an AP 310 DL/1 results in a REMOTE/PRINT timeout exception.

Banners

The REMOTE/PRINT program can cause the printing of banner pages with the file name and creation date of the backup file in block letters (10 characters by 10 characters). One banner page precedes and another follows the printer output text. Banner printing is suppressed if the printer does not have a default page width of 132.

The printing of banners is controlled by the setting of the LABEL (LABEL.TYPE) file attribute for the REMOTE/PRINT file BACKUP. If LABEL is set to STANDARD, banners are printed by default on all printer types that have a default page width of 132. If the LABEL file attribute is set to OMITTED, no banner pages are printed.

Banner printing can also be specified with the PRINT command option BANNERS.

Examples:

```
WFL MODIFY REMOTE/PRINT FILE BACKUP (LABEL=OMITTED);  
Banner printing is suppressed.
```

```
WFL MODIFY REMOTE/PRINT FILE BACKUP (LABEL=STANDARD);  
Banner printing takes place if page width is 132.
```

Page Width

The page width is controlled by the RECORD.SIZE (MAXRECSIZE) file attribute for the REMOTE/PRINT file BACKUP.

Page width is specified in tenths of an inch and is used to calculate the number of characters per inch to use when printing on an AP 310, AP 1301, or AP 1340 remote printer. The default page width is 85 characters for the AP 310 and 132 characters for all other printer types.

Examples:

```
WFL MODIFY REMOTE/PRINT FILE BACKUP (MAXRECSIZE=85);  
Specifies a default page width of 85.
```

```
WFL MODIFY REMOTE/PRINT FILE BACKUP (MAXRECSIZE=120);  
Specifies a default page width of 120.
```

Page Size

The page size (number of lines) is controlled by the PAGE.SIZE (PAGESIZE) file attribute for the REMOTE/PRINT file BACKUP. The value specified for the page size is downloaded to the AP 310, AP 1301, and AP 1340 remote printers. The default page size is 66 lines (an 11-inch form at 6 lines per inch).

The default page size is overridden by the PRINT command option PAGESIZE or by creating a backup file with a specific page size. The PRINT option takes precedence over the backup file and default page sizes.

Examples:

```
WFL MODIFY REMOTE/PRINT FILE BACKUP (PAGESIZE = 66);  
  Specifies a default page size of 66.
```

```
WFL MODIFY REMOTE/PRINT FILE BACKUP (PAGESIZE = 88);  
  Specifies a default page size of 88.
```

Saving/Purging of the Backup File

The backup file can be saved or purged by modifying the SAVE (SAVEFACTOR) file attribute of the REMOTE/PRINT file BACKUP. If the SAVE file attribute is zero (0), the backup file is purged upon completion of the print request; otherwise, it is saved. The default is to save the backup file.

The SAVE file attribute setting is overridden by the PRINT command option SAVEFILE.

Examples:

```
MODIFY REMOTE/PRINT FILE BACKUP SAVE = 1;  
  Causes a backup file to be saved upon completion of a print request.
```

```
MODIFY REMOTE/PRINT FILE BACKUP SAVE = 0;  
  Causes a backup file to be purged upon completion of a print request.
```

Output Device

The output device for the REMOTE/PRINT program is specified by the device attribute for the REMOTE/PRINT file PRINTER. (No control card attribute name is needed; the Burroughs standard attribute name is KIND.) If device attribute is specified as PRINTER, the output is directed to the system line printer. If the attribute is specified as REMOTE (the default), the output is directed to the remote printer file specified by the TITLE attribute for the file PRINTER. (See Remote File Title, next.)

Examples:

```
WFL MODIFY REMOTE/PRINT FILE PRINTER (KIND = PRINTER);  
  Output is directed to the system line printer.
```

```
MODIFY REMOTE/PRINT FILE PRINTER (KIND = REMOTE);  
  Output is directed to a remote printer.
```

Remote File Title

The title of the remote file is controlled by the TITLE file attribute for the file PRINTER. If the device attribute is REMOTE, the TITLE file attribute specifies the name of the file in the network controller that contains the remote printer to which output is to be directed.

The default external file name for the remote file is PRINTER.

Example:

```
WFL MODIFY REMOTE/PRINT FILE PRINTER (TITLE=PRINTER);
```

The REMOTE/PRINTER program prints on the remote printer that is in a file titled PRINTER.

Printer Switching

Printer switching, the capability to select the first available remote printer from a defined group of remote printers, is controlled by the NUMBER.STATIONS (STATIONSALLOWED) file attribute for the REMOTE/PRINT file PRINTER. If the STATIONSALLOWED file attribute is set to 0 or 1, one remote printer is assumed; otherwise, the value assigned to the file attribute specifies the number of remote printers and the REMOTE/PRINT program attempts to switch among them until it finds one that is not in use.

The remote printer files must be declared separately in the network controller using the following format

```
<basename> <number>
```

where <basename> has eight or fewer characters and <number> is one of a consecutive series that uniquely identifies the file for each printer. Thus, the files PRINTER1, PRINTER2, and PRINTER3 could be used in this manner for printer switching. The default is no printer switching.

Examples:

```
WFL MODIFY REMOTE/PRINT FILE PRINTER (STATIONSALLOWED=5);
```

Specifies that five remote printers are available.

```
WFL MODIFY REMOTE/PRINT FILE PRINTER (STATIONSALLOWED=1);
```

Specifies that only one remote printer is available.

Backup Output Translation

The translation of backup file characters to upper-case printer output characters is controlled by program switch 4. If program switch 4 is set, the output is translated to upper-case letters; otherwise, the output is not translated. The default condition is no translation.

Output translation for a print request may be invoked by the UPPERCASE print option.

Examples:

```
MODIFY REMOTE/PRINT SW4=TRUE;  
    Backup file characters are translated to upper-case printer output characters.
```

```
MODIFY REMOTE/PRINT SW4=FALSE;  
    Backup file characters are not translated for printing.
```

ODT Output Translation

The translation of ODT output to upper case is controlled by program switch 6. If program switch 6 is set, ODT output is translated to upper case; otherwise, ODT output is not translated. The default condition is no translation.

Examples:

```
MODIFY REMOTE/PRINT SW6=TRUE;  
    ODT output is translated to upper case.
```

```
MODIFY REMOTE/PRINT SW6=FALSE;  
    ODT output is not translated.
```

Program Termination

Whether the REMOTE/PRINT program remains in the mix or goes to end of job after printing one file is controlled by the setting of program switch 8. If program switch 8 is set, the REMOTE/PRINT program remains in the mix; otherwise, the REMOTE/PRINT program terminates after processing one print request.

If the REMOTE/PRINT program is initiated by the Work Flow Language (WFL) compiler, the REMOTE/PRINT program automatically goes to end of job after processing one print request.

By default, the REMOTE/PRINT program goes to end of job after processing one print request.

Examples:

```
MODIFY REMOTE/PRINT SW8=TRUE1;  
    The REMOTE/PRINT program remains in the mix indefinitely.
```

```
MODIFY REMOTE/PRINT SW8=FALSE;  
    The REMOTE/PRINT program terminates after processing one print request.
```


Summary of Default Program Attributes

Table 24-3 summarizes the attributes used to modify the various functions of the REMOTE/PRINT program. For program switches, an attribute value of TRUE is equivalent to 1; FALSE is equivalent to 0.

Table 24-3. File and Program Switch Attributes

Function and Desired Result	File	Switch or Attribute	Value
OUTPUT MESSAGE BLOCKING Disable: Enable:	----	SW1 SW1	= TRUE = FALSE
If SW1 = FALSE, printer buffer size + 50 = <integer>:	PRINTER	MAXRECSIXE	= <integer>
BANNERS Do not print: Print:	BACKUP BACKUP	LABEL LABEL	= OMITTED = STANDARD
PAGE WIDTH Character count = <integer>:	BACKUP	MAXRECSIZE	= <integer>
PAGE LENGTH Line count = <integer>:	BACKUP	PAGESIZE	= <integer>
SAVE FILE AFTER PRINTING? Save: Purge:	BACKUP BACKUP	SAVEFACTOR SAVEFACTOR	= 1 = 0
OUTPUT DEVICE Line Printer: Remote Printer:	PRINTER PRINTER	KIND KIND	= PRINTER = REMOTE
FILE TITLE If KIND = REMOTE, remote filename = <title>:	PRINTER	TITLE	= <title>
PRINTER SWITCHING If KIND = REMOTE, available printers = <integer>:	PRINTER	MAXSTATIONS	= <integer>
BACKUP OUTPUT TRANSLATION Translate to upper case: Do not translate:	---- ----	SW4 SW4	= FALSE = TRUE
ODT OUTPUT TRANSLATION Translate to upper case: Do not translate:	---- ----	SW6 SW6	= FALSE = TRUE
TERMINATE PROGRAM? Prints one file, terminates: Remains in mix:	---- ----	SW8 SW8	= FALSE = TRUE

OPERATING INSTRUCTIONS

The two methods for executing the REMOTE/PRINT program, ODT execution and WFL execution, are described next.

ODT Execution

PRINTER, the remote file of the REMOTE/PRINT program, must be file-equated to the remote file in the network controller that contains the remote printer. (See Network Controller Installation, ear- The default external file name for the remote file is PRINTER.

Program execution is initiated with either of the following statements:

(1) EXECUTE REMOTE/PRINT; AX PRINT MYFILE

or

(2) EXECUTE REMOTE/PRINT FILE PRINTER TITLE AP__FILE; followed by

<mix number>AX PRINT MYFILE

Both forms of the command are parsed to a a percent sign (%) character, if any, and the rest of the string is treated as a comment and ignored. The maximum length of the entry is 120 characters. Upper-case or lower-case letters may be used.

If program switch 8 is reset (SW8=0), only one AC or AX message, specifying a file to be printed, is allowed before the program goes to end of job. If SW8=1, multiple AC or AX messages are allowed and the EOJ message is required to terminate execution.

WFL Execution

The REMOTE/PRINT program may be executed using the Work Flow Language (WFL) compiler by file equating the title of the file to be printed to the REMOTE/PRINT file BACKUP.

Table 24-4 lists the file attributes that control the manner in which the REMOTE/PRINT program is executed when using the WFL compiler. The TITLE attribute for the file BACKUP must be specified.

Refer to the *B 1000 System Work Flow Language (WFL) Installation Operation, and Functional Description Manual* for more information on WFL.

Example:

```
BEGIN JOB PRINTIT;  
  RUN REMOTE/PRINT;  
    FILE BACKUP (TITLE = MYFILE);  
    FILE PRINTER (TITLE = AARDVARK);  
END JOB;
```

This causes the backup file MYFILE to be printed on the remote printer described in the remote file AARDVARK.

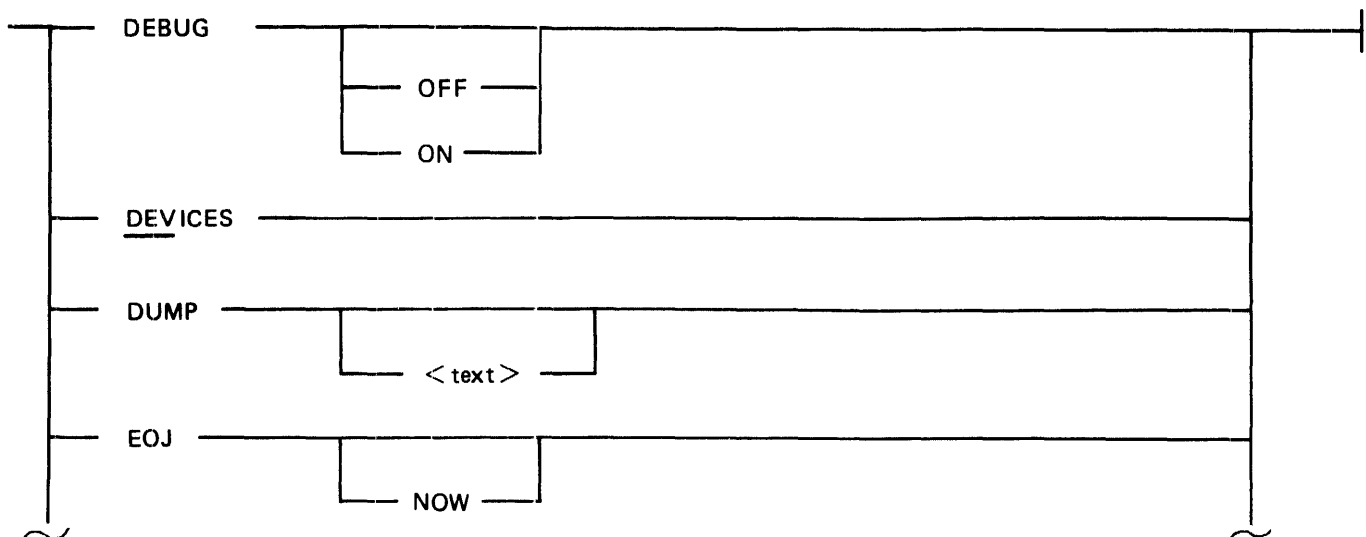
Table 24-4. File Attributes for WFL Execution

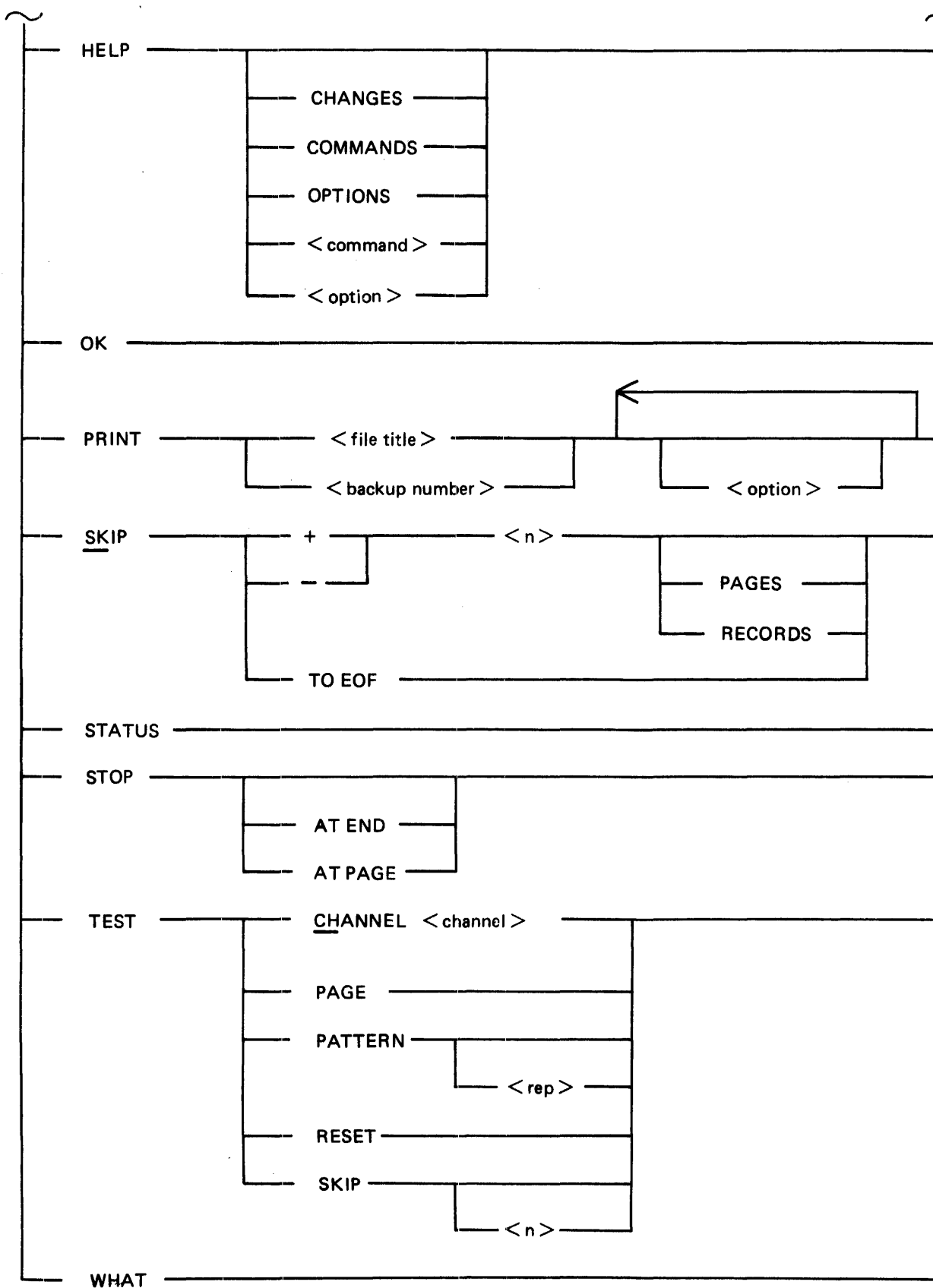
REMOTE/PRINT File	File Attribute	Attribute Value	Print Option or Function performed
BACKUP	LABEL	OMITTED STANDARD	BANNERS = FALSE BANNERS = TRUE
BACKUP	MAXRECSIZE	<n>	PAGEWIDTH = <n>
BACKUP	MINRECSIZE	<n>	EXTRACT COLUMNS 1 - <
BACKUP	PAGESIZE	<n>	PAGESIZE = <n>
BACKUP PRINTCOPIES = <n>	PRINTCOPIES	<n>	
BACKUP	SAVEFACTOR	0 1	SAVEFILE = FALSE SAVEFILE = TRUE
BACKUP PRINTER	TITLE KIND	<title> PRINTER REMOTE	PRINT <title> Output to system printer Output to remote printer contained in the remote file specified by KIND.
PRINTER	TITLE	<title>	Remote file name in the network controller if KIND = REMOTE.

PROGRAM COMMANDS

The following diagram is an overview of the syntax of all the REMOTE/PRINT program commands. Each command is described in the paragraphs that follow the diagram. The options for the PRINT command are described separately after all of the program command descriptions.

Syntax:





Semantics:

DEBUG

Used for debugging the REMOTE/PRINT program. Off is default.

DEVICES

Produces a display of the current status and release level of the REMOTE/PRINT program and, if the device is a remote printer, a display of the printer device type and the station name.

DUMP

Causes the program to create a program dump. If <text> is specified, it is included in the dump.

EOJ

Causes the program to terminate after printing the current copy of the print request. EOJ NOW causes immediate termination.

HELP

Displays information on using the REMOTE/PRINT program:

HELP Option	Description Provided
CHANGES	the program.
COMMANDS	all the program commands.
OPTIONS	all the options for the PRINT command.
<command>	a specific program command.
<option>	a specific PRINT command option.

OK

Resumes printing after a STOP. (See STOP.)

PRINT

<file title> is any valid B 1000 file title. <backup number> is the number of the system-created backup file. <option> is any of the fifteen PRINT command options described following the program command descriptions. (See PRINT Command Options.)

SKIP

Moves the pointer to the backup file being printed. **SKIP + <n> PAGES (RECORDS)** skips <n> pages (records); **SKIP - <n> PAGES (RECORDS)** repeats <n> pages (records). **RECORDS** is the default in both cases. **SKIP TO EOF** terminates printing of the current file and all remaining copies.

A page skip takes place whenever the the **SKIP** command is invoked unless the **SUPPRESS** option of the **PRINT** command has been specified.

NOTES

If the page size of the current form is different from the default page size or the backup file page size, the **PAGESIZE** option of the print command must be specified or errors may occur in calculations of page breaks.

The control records that precede the text in the the printer backup file are never printed as text.

For printers with buffers, the **SKIP** command takes effect on text that has not yet reached the buffer; therefore the action appears to be delayed.

STATUS

Displays the status of the current print request. Identical to **WHAT**.

STOP

Causes an immediate pause in printing. If there is no print activity and **SW8=1**, the program goes idle; otherwise, it terminates.

STOP AT END delays the pause until printing of the current copy is complete. **print request**, and **STOP AT PAGE** delays it until the next page.

NOTES

If the page size of the current form differs from the default page size or the backup file page size, the **PAGESIZE** option of the **PRINT** command must be specified or calculations of page breaks may be erroneous for the **STOP AT PAGE** command.

For printers with buffers, printing continues until the buffer is empty.

TEST

The REMOTE/PRINT program must be in an idle state or the STOP command must be entered prior to invoking TEST.

TEST CHANNEL or TEST CH tests with a skip to channel <channel>. <channel> is an integer in the range 1 through 12. This option is valid for the system line printer and the B 9246 remote printer.

TEST PAGE tests with one a page skip.

TEST PATTERN prints a ripple pattern of test characters and repeats <rep> times.

TEST RESET tests the remote reset capabilities of the printer. (Valid for remote printers only.)

TEST SKIP causes the program to skip <n> lines. If <n> is omitted or zero, one line is skipped.

WHAT

Identical to STATUS.

PRINT COMMAND OPTIONS

On entry of the PRINT command, if the REMOTE/PRINT program ascertains that the file is a printer backup file, the file is queued for printing.

Record Size

The maximum record size allowed for a printer backup file is 212 characters for the AP 1340 remote printer, 197 characters for the AP 1340, and 134 characters for all other printer types. The maximum record size includes two characters of control information.

NOTE

The maximum record sizes listed above are only valid if a form of the correct size is installed in the printer; otherwise truncation might occur.

File Specification

If the REMOTE/PRINT program is executed without a usercode and if no usercode is specified (PRINT command option USER), the format of the backup file is BACKUP/PRT<n>, where <n> is the backup number. The REMOTE/PRINT program searches the disk designated as the backup disk (BD) for system-created backup files unless a disk file identifier is explicitly entered.

If the REMOTE/PRINT program is executed with a usercode or if a usercode is specified with the USER option, the format of the backup file is (<usercode>)PRT<n>.

Refer to Volume 1 of *the B 1000 Systems System Software Operation Guide* for information on backup files and file titles.

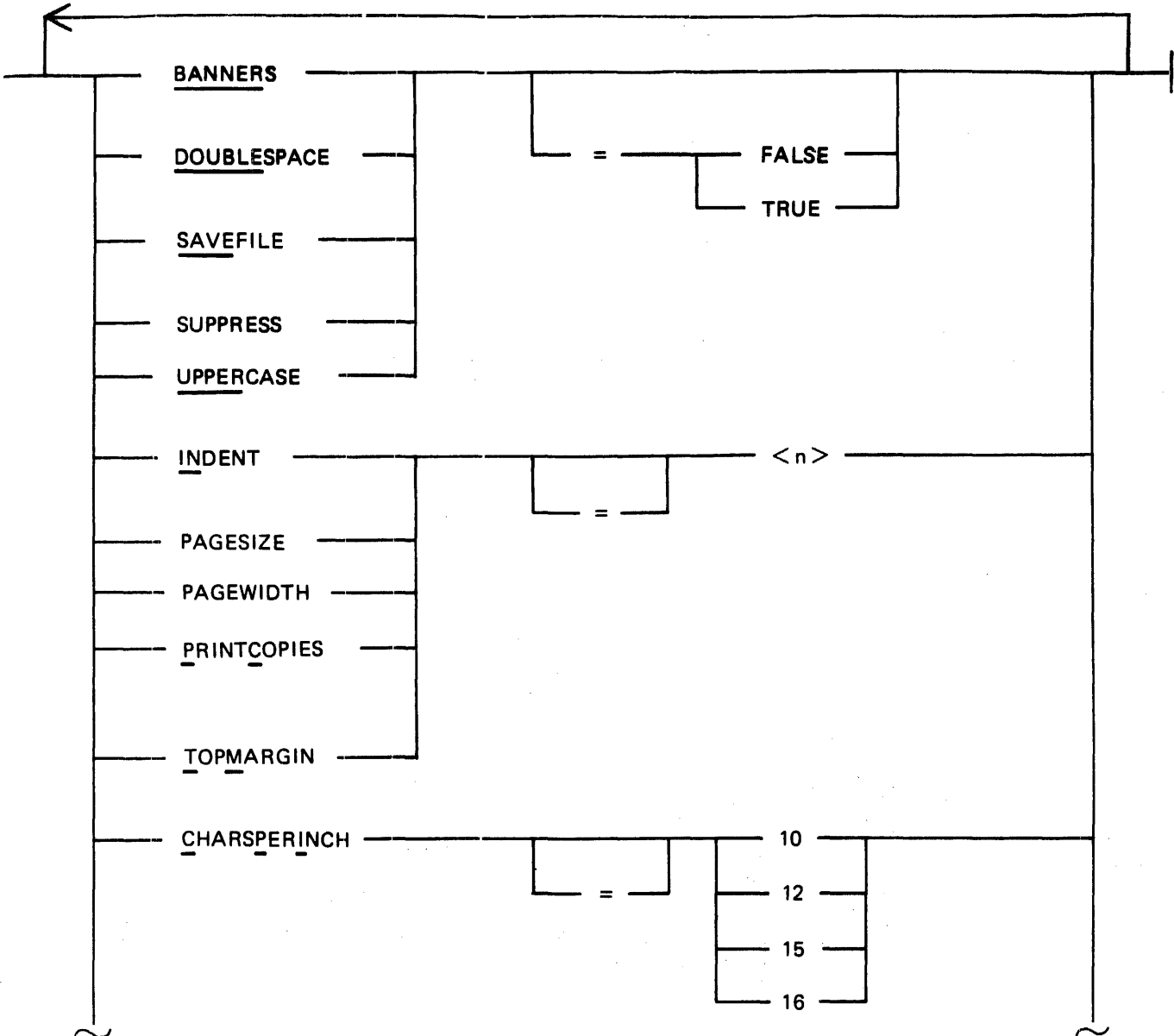
Examples:

```
PRINT MYFILE  
PRINT PRT123 BANNERS PRINTCOPIES 3  
PRINT (USER)BACKUP ON P
```

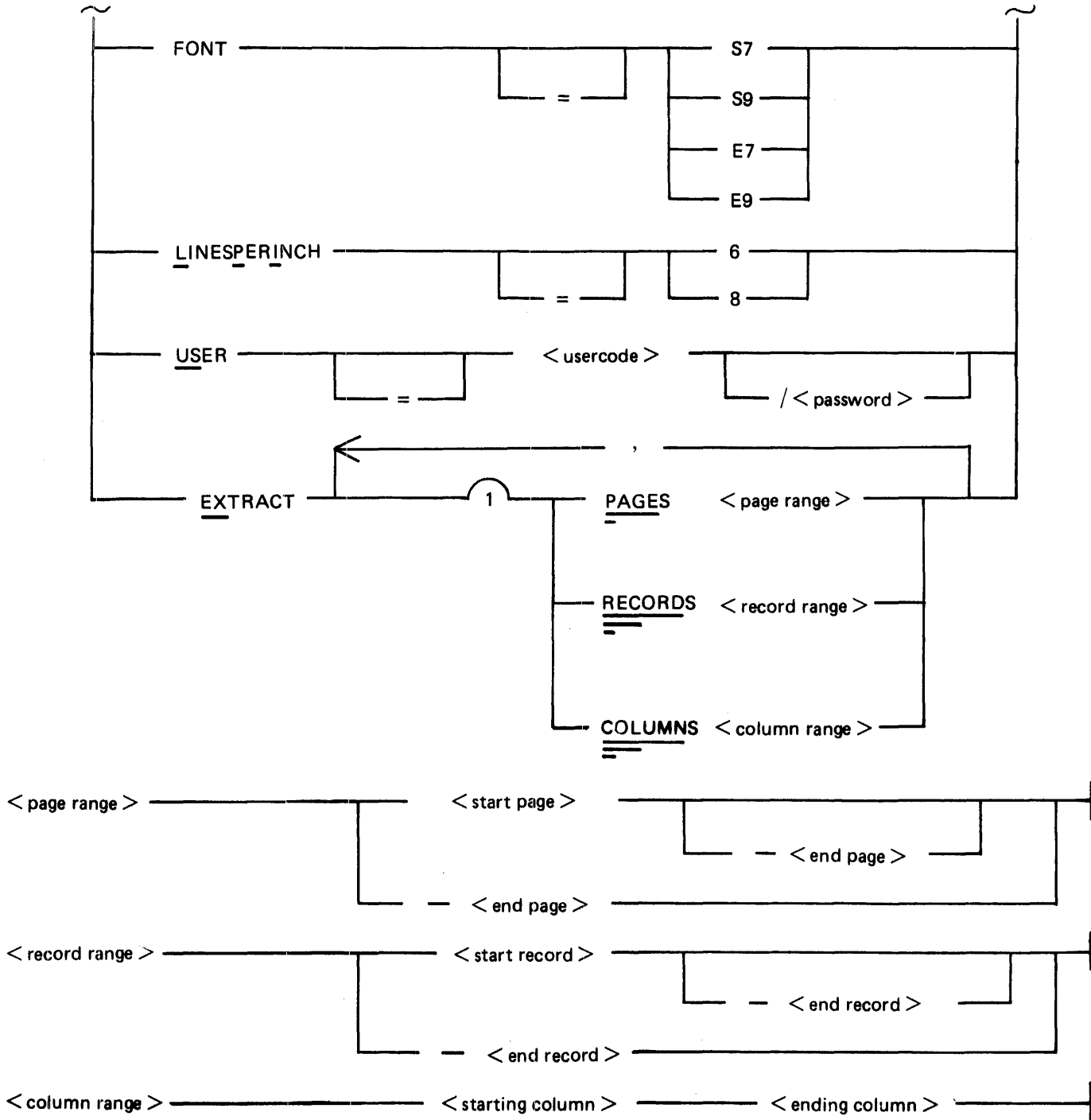

Syntax Diagram

The following diagram is an overview of the syntax of all the options for the PRINT program command. The diagram is organized for clarity and compactness. The semantics of each option are presented in alphabetical order following the diagram.

Syntax:



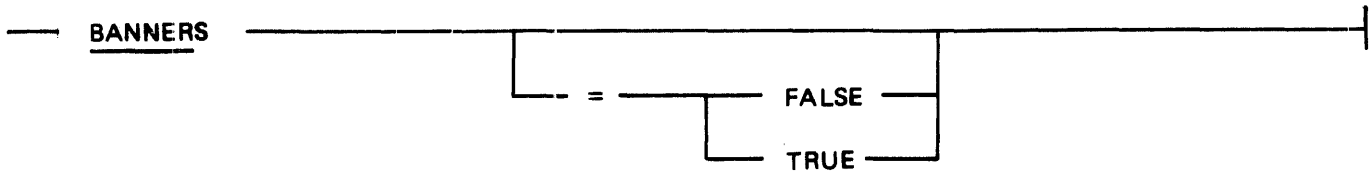
B 1000 Systems System Software Operation Guide, Vol. 2
 REMOTE/PRINT



BANNERS

The BANNERS print option allows the user to specify that a banner consisting of the file name and the creation date of the file is to be printed on a header page preceding and a trailer page following the backup file.

Syntax:



If neither TRUE nor FALSE is specified, banners are printed.

The BANNERS option is TRUE by default for printer devices that have a default page width of 132.

NOTE

Specifying BANNERS with a page width of less than 132 characters for printers such as the B 9246 and the system line printer can cause truncation of the title and/or the creation date.

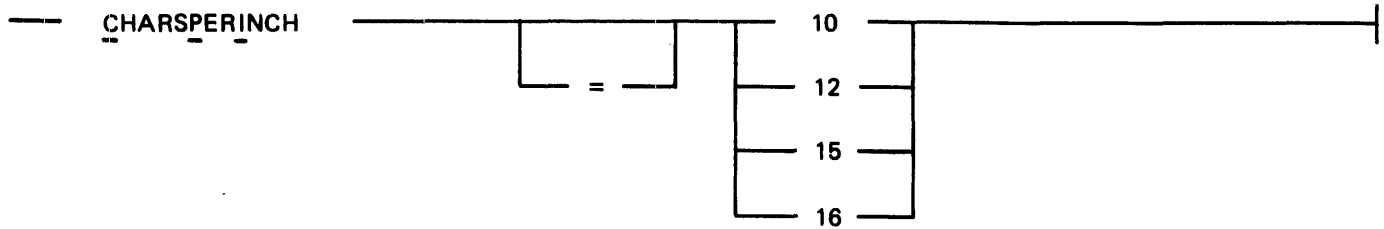
Examples:

```
BANNERS
BANNERS = TRUE
BANNERS = FALSE
BANNER
BANNER = TRUE
BANNER = FALSE
```

CHARSPERINCH

The CHARSPERINCH print option is used to specify the horizontal spacing (characters per inch) to be used when printing on the AP 310, AP 1301, and AP 1340 remote printers.

Syntax:



For the AP 310 and AP 1340 remote printers, valid values are 10, 12, and 16. For the AP 1301 remote printer, the values are 10, 15, and 16.

If CHARSPERINCH is specified for printers other than the AP 310, AP 1301, and AP 1340 remote printers, an error message is generated and the print request is rejected.

If CHARSPERINCH is not specified, the REMOTE/PRINT program calculates a value based on the text width and the page width for the AP 310, AP 1301, and AP 1340 remote printers.

Examples:

CHARSPERINCH = 10

Valid for AP 310, AP 1301, and AP 1340 remote printers.

CHARSPERINCH 12

Valid for AP 310, AP 1301, and AP 1340 remote printers.

CPI = 15

Valid for AP 1301 remote printer only.

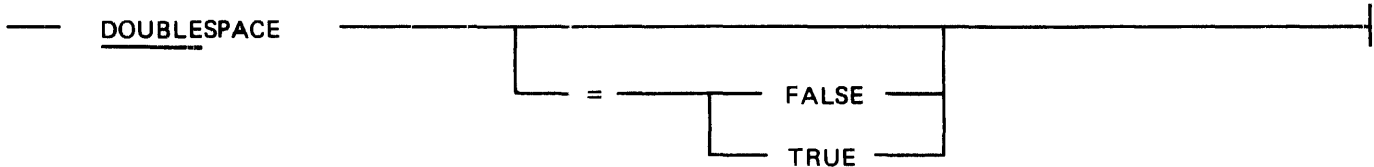
CPI 16

Valid for AP 310 and AP 1340 remote printers only.

DOUBLESPACE

The DOUBLESPACE print option is used to double all carriage controlled line spacing values except overprints and channel skips. For example, a single space is converted to a double space and a double space is converted to four spaces.

Syntax:



If neither TRUE nor FALSE is specified, the file is double spaced.

The default value for the DOUBLESPACE option is FALSE.

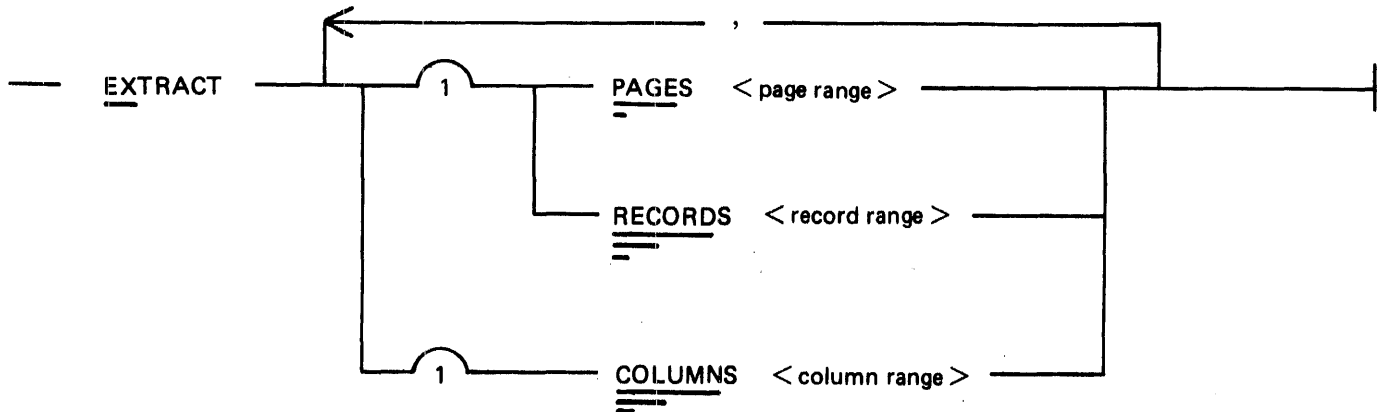
Examples:

```
DOUBLESPACE
DOUBLESPACE = TRUE
DOUBLESPACE = FALSE
DOUBLE
DOUBLE = TRUE
DOUBLE = FALSE
```

EXTRACT

The **EXTRACT** print option is used to request printing of a portion of a file rather than an entire file. Extraction may be done on the basis of pages, records, or columns.

Syntax:



If **EXTRACT PAGES** is specified, the record range specified by <start page> and <end page> is printed. If <start page> is omitted, the first page of actual text is used as the starting page. If <end page> is omitted or is greater than the last page of the file, printing is to the end of the file.

If **EXTRACT RECORDS** is specified, the record range specified by <start record> and <end record> is printed. If <start record> is omitted, the first record of actual text is used as the starting record. If <end record> is omitted or is greater than the end-of-file pointer, the end-of-file pointer is used as the ending record.

The print options **EXTRACT PAGES** and **EXTRACT RECORDS** cannot be specified for the same execution of the Remote/Print program.

If **EXTRACT COLUMNS** is specified, the specified column range is printed.

The default is to print the entire file.

NOTES

If the page size of the current form is different from the default page size or the page size that the backup file was created with, the **PAGESIZE** print option must be specified or calculations of page breaks may be erroneous.

The control records that precede the text in the the printer backup file are never printed as text.

Examples:

EXTRACT PAGES 5 - 10
EXTRACT PAGE 5 - 10
EXTRACT P 5 - 10
Print pages 5 through 10.

EXTRACT PAGES - 10
EXTRACT PAGE - 10
EXTRACT P - 10
Print pages 1 through 10.

EXTRACT PAGES 5
EXTRACT PAGE 5
EXTRACT P 5
Print pages 5 through the end of file.

EXTRACT RECORDS 25 - 50
EXTRACT RECORD 25 - 50
EXTRACT REC 25 - 50
EXTRACT R 25 - 50
Print records 25 through 50.

EXTRACT RECORDS - 50
EXTRACT RECORD - 50
EXTRACT REC - 50
EXTRACT R - 50
Print records 1 through 50.

EXTRACT RECORDS 25
EXTRACT RECORD 25
EXTRACT REC 25
EXTRACT R 25
Print records 25 through the end of file.

EXTRACT COLUMNS 10 - 85
EXTRACT COLUMN 10 - 85
EXTRACT COL 10 - 85
EXTRACT C 10 - 85
Print columns 10 through 85.

FONT

The FONT print option is used to specify a type font for printing on an AP 310 remote printer.

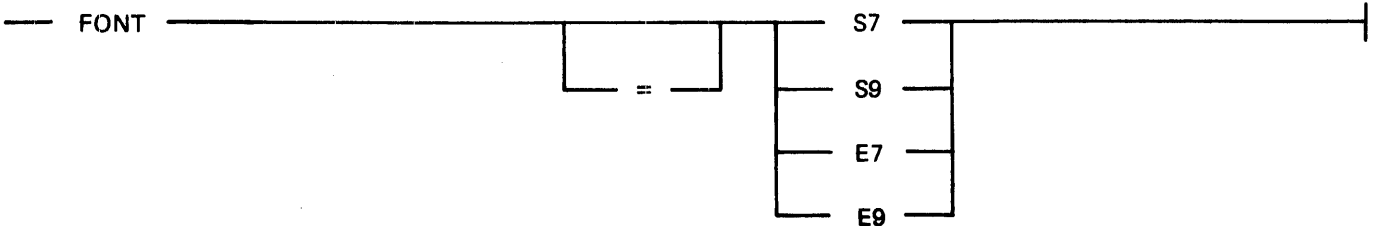
Four fonts are available:

- S7 Standard seven-pin font
- S9 Standard nine-pin font
- E7 Expanded seven-pin font),
- E9 Expanded nine-pin font

Refer to the *AP 310 Equipment (Data Communications Interface) Reference Manual*, form 1119153, for more information on fonts that are available for the AP 310 series of remote printers.

The FONT option is valid only for the AP 310 remote printer. If it is specified for another printer type, an error message is generated and the print request is rejected.

Syntax:



The default font is the standard seven-pin font (S7) for the AP 310 remote printer.

Examples:

```
FONT = S7  
FONT S9  
FONT E7  
FONT = E9
```


INDENT

The INDENT print option is used to specify that the printed text is to be indented <n> blank characters from the left.

The default is no indentation.



Examples:

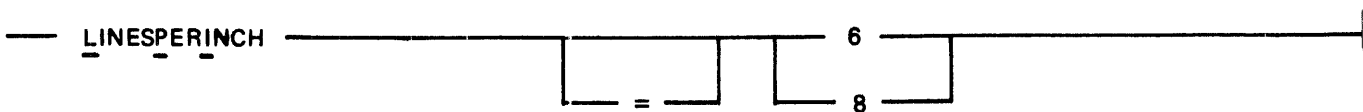
```
INDENT = 10
INDENT 10
IN = 10
IN 10
```

LINESPERINCH

The LINESPERINCH print option is used to select the vertical lines-per-inch value for printing on the AP 310, AP 1301, and AP 1340 remote printers. Possible values are 6 or 8.

If the LINESPERINCH option is specified for a printer other than the three named, an error message is generated and the print request is rejected.

The default value is 6.



Examples:

```
LINESPERINCH = 6
LINESPERINCH 8
LPI 6
LPI = 8
```

PAGESIZE

The PAGESIZE print option is used to specify the number of lines per page of printing.

The default is to print 66 lines per page.

The specified page size is downloaded to the AP 310, AP 1301, and AP 1340 remote printers. An incorrect page size specification results in incorrect page skips.

The PAGESIZE print option is also important when PAGES is specified in the SKIP program command, when AT PAGES is specified in the STOP program command, and when PAGES, PAGE, or situations results in erroneous calculations of the number of pages to skip.

— PAGESIZE ————— <n> —————
 ┌ ┐
 └ ┘
 =

Examples:

PAGESIZE = 88
PAGESIZE 88

PAGEWIDTH

The PAGEWIDTH print option is used to specify the page width in tenths of inches of the printer form.

Defaults are 85 for the AP 310 remote printer and 132 for all other remote printer types. If a value other than 85 is specified for the AP 310 remote printer, an error message is generated and the entry is rejected.

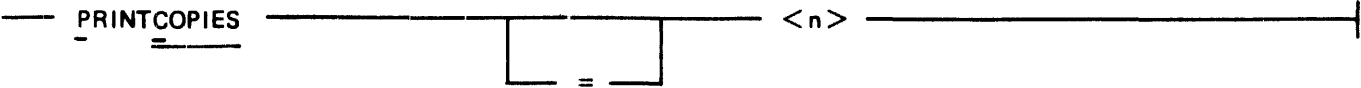
— PAGEWIDTH ————— <n> —————
 ┌ ┐
 └ ┘
 =

Examples:

PAGEWIDTH = 85
PAGEWIDTH 120

PRINTCOPIES

The PRINTCOPIES print option causes <n> copies of the printer backup file to be printed. The default is to print the number of copies that was specified in the PRINTCOPIES file attribute when the backup file was created. The PRINTCOPIES option overrides the value of the PRINTCOPIES file attribute.

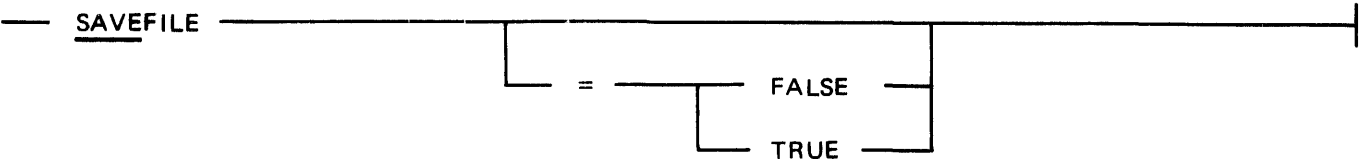


Examples:

```
PRINTCOPIES = 5
PRINTCOPIES 10
PC = 5
PC 10
COPIES = 3
COPIES 3
```

SAVEFILE

The SAVEFILE print option is used to specify whether the printer backup file is to be saved or purged upon completion of printing.



If the SAVEFILE option is specified as TRUE, or if no boolean value is specified, the file is saved upon completion of the print request. If the SAVEFILE option is specified as FALSE, then the file is purged from the disk directory.

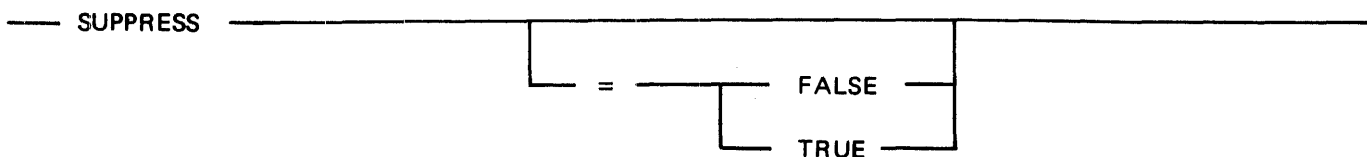
The default for this option is controlled by the setting of the SAVEFACTOR file attribute for the REMOTE/PRINT file BACKUP. If the SAVEFACTOR file attribute is zero, the backup file is automatically purged on completion of the print request; otherwise, the backup file is saved.

Examples:

```
SAVEFILE
SAVEFILE = TRUE
SAVEFILE = FALSE
SAVE
SAVE = TRUE
SAVE = FALSE
```

SUPPRESS

The SUPPRESS print option is used to override the printer control characters in a printer backup file, thereby causing the output to be printed single spaced if the DOUBLESPEACE option is FALSE or double spaced if the DOUBLESPEACE option is TRUE. Multiple occurrences of blank lines are also suppressed. Records under overprint control are not affected.



If neither TRUE nor FALSE is specified, TRUE is assumed and the printer control characters are suppressed.

The SUPPRESS option is FALSE by default.

Examples:

```
SUPPRESS  
SUPPRESS = TRUE  
SUPPRESS = FALSE
```

TOPMARGIN

The TOPMARGIN print option is used to specify that a top margin of <n> blank lines is to be inserted at the top of the first page only. The default is no top margin.



Examples:

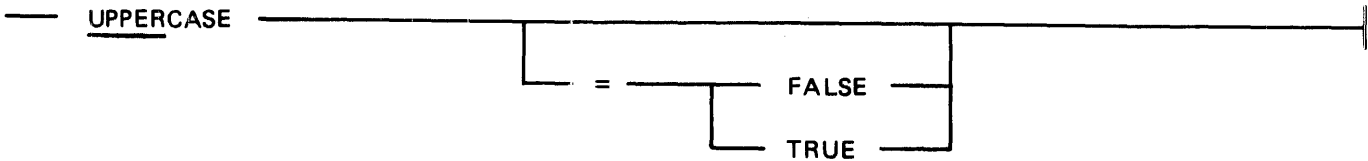
```
TOPMARGIN = 5  
TOPMARGIN 10  
TM 5  
TM = 10
```

UPPERCASE

The UPPERCASE print option is used to controls the translation of lower-case to upper-case letters in the output text.

If TRUE or no value is specified, translation to upper-case letters is performed. If FALSE is specified, no translation is performed.

The default for translating the text to upper case is controlled by the setting of program switch 4. If program switch 4 is set (= 1), the text is translated to upper-case by default; otherwise no translation is performed.



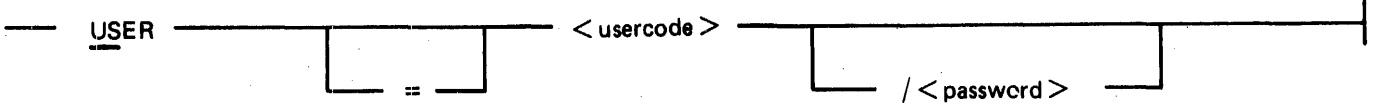
Examples:

```
UPPERCASE
UPPERCASE = TRUE
UPPERCASE = FALSE
UPPER
UPPER = TRUE
UPPER = FALSE
```

USER

The USER print option is used to specify the usercode and password on whose behalf the file is to be opened.

The default is to open the file using the usercode and password, if any, that are specified with the EXECUTE or RUN commands.



Semantics:

<usercode>

The <usercode> keyword specifies the usercode for the USER option. It must not exceed eight characters and it must begin with a letter.

<password>

The <password> keyword specifies the password for the USER option. It must not exceed ten characters and it may be blank if the user's password is null.

Examples:

```
USER A/B  
USER = A/B  
USER A
```


SECTION 25 SDL/SSC

The SDL/SSC (SDL stack size changer) is an SDL support program that enables an operator to display or change the sizes of the various stacks in an SDL/UPL program code file without recompiling the program. Because the SDL/SSC program deals exclusively with the scratchpad format expected by the SDL interpreter, it cannot be used to display or modify the scratchpad of any programs other than those compiled by the SDL or UPL compilers.

The SDL/SSC program operates on the original copy of the scratchpad contained in the object code file. This means that it is not possible to change the stack sizes of a program in either the schedule or the mix because a working copy of the scratchpad is being used rather than the original copy.

OPERATING INSTRUCTIONS

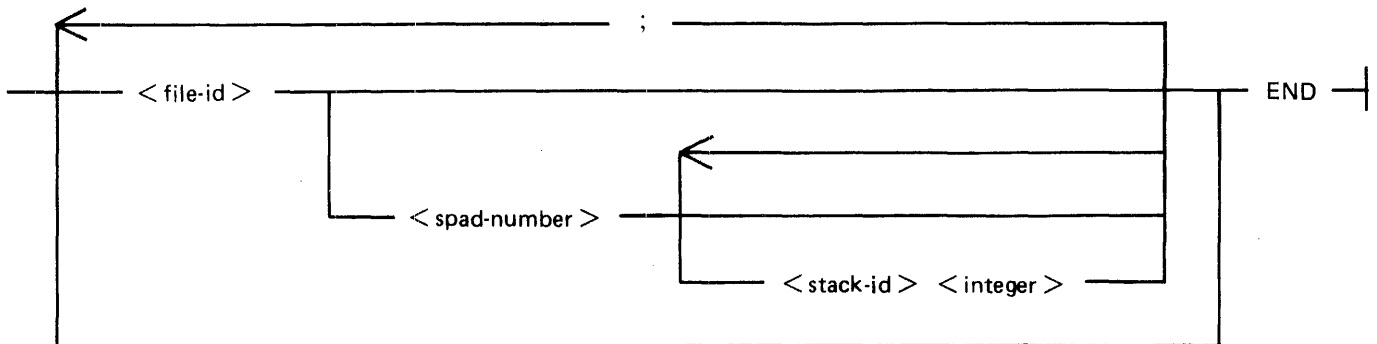
The SDL/SSC program is executed by entering the following command:

EX SDL/SSC SW1 <0 or 1>

If SW1 = 0, input specifications are expected as responses to ACCEPT messages. If SW1 = 1, input messages are expected from a card file labeled CARDS.

In either case, specifications are input to the program in accordance with the following syntax.

Syntax:



Semantics:

The <file-id>, which must be the file name of an SDL/UPL code file, is the first parameter to be entered in response to the SDL/SSC program's AX message, or is the first item in the card file.

If <file-id> is entered without parameters and transmitted, the system responds with a display of the current stack sizes, the size of dynamic memory, and the total core size. (Refer to the example at the end of this section.)

<spad-number> identifies the scratchpad. Zero is the usual entry.

<stack-id> <integer>, a letter-integer pair, identifies the stack to be changed and the desired size. Stack names, identifiers, and size units are listed in table 25-1.

If more than one size is to be changed in one pass, the <spad-number> may be followed by multiple <stack-id> <integer> pairs.

Changes to stacks in more than one code file may be specified in a single execution of the SDL/SSC program. Each new codefile sequence must be preceded with a semicolon.

The program is terminated by the entry of a blank or the keyword END.

Program prompts provide guidance.

NOTE

Only SDL/UPL programs that use the CHANGE__STACK__SIZES statement have more than one scratchpad. The COBOL compiler is one example of an SDL program that changes its stack sizes between passes.

Table 25-1. Identifiers and Size Units of SDL Stacks

Stack Name	<stack-id>	<integer>
Value Stack	V	bits
Name Stack	N	entries (48 bits each)
Display Stack	X	bits
Control Stack	C	entries (48 bits each)
Evaluation Stack	E	entries (48 bits each)
Program Pointer Stack	P	entries (32 bits each)
Dynamic Memory	D	bits

Restrictions

The following two restrictions apply to the size of the SDL program stacks:

1. Evaluation Stack plus Program Pointer Stack total size may not exceed 65,536 bits.
2. Total stack size excluding Dynamic Memory may not exceed 1,048,576 bits (128KB).

EXAMPLE

An example of an ODT session following execution of the program is provided below. System responses are indented.

```

EX SDL/SSC SW1 0
  SDL/SSC =1234 BOJ. ...
  % SDL/SSC =1234 SDL/SSC, MARK <version> OF <compile date/time>
  % SDL/SSC =1234
  % SDL/SSC =1234 ENTER FILENAME OR BLANK TO TERMINATE
  SDL/SSC =1234 ACCEPT.
1234AX TEST/PROGRAM                                % <file-id>
  % SDL/SSC =1234 ***** FILE NAME : TEST/PROGRAM
  % SDL/SSC =1234 ENTER SCRATCHPAD NUMBER
  SDL/SSC =1234 ACCEPT.
1234AXO                                             % <spad-number>
  % SDL/SSC =1234
  % SDL/SSC =1234 ***** SCRATCHPAD # 0
  % SDL/SSC =1234 VALUE..... (V) :13000
  % SDL/SSC =1234 NAME..... (N) :100
  % SDL/SSC =1234 DISPLAY..... (D) :160
  % SDL/SSC =1234 CONTROL..... (C) :15
  % SDL/SSC =1234 EVALUATION..... (E) :20
  % SDL/SSC =1234 PROGRAM POINTER. (P) :25
  % SDL/SSC =1234 DYNAMIC..... (D) :15000
  % SDL/SSC =1234 ***** TOTAL CORE: 35840
  % SDL/SSC =1234 ENTER STACK ID

  SDL/SSC =1234 ACCEPT.
1234AX V 14000                                     % <stack-id> <integer>
  % SDL/SSC =1234 ENTER ' ' TO END CHANGES
  SDL/SSC =1234 ACCEPT.
1234AX N 50 ;                                       % <stack-id> <integer>
  % SDL/SSC =1234
  % SDL/SSC =1234 NEW STACK SIZES
  % SDL/SSC =1234
  % SDL/SSC =1234
  % SDL/SSC =1234 ***** SCRATCHPAD: 0
  % SDL/SSC =1234 VALUE..... (V) :14000
  % SDL/SSC =1234 NAME..... (N) :50
  % SDL/SSC =1234 DISPLAY..... (D) :160
  % SDL/SSC =1234 CONTROL..... (C) :15
  % SDL/SSC =1234 EVALUATION..... (E) :20
  % SDL/SSC =1234 PROGRAM POINTER. (P) :25
  % SDL/SSC =1234 DYNAMIC..... (D) :15000
  % SDL/SSC =1234 ***** TOTAL CORE : 34440
  % SDL/SSC =1234 ENTER FILENAME OR BLANK TO TERMINATE
  SDL/SSC =1234 ACCEPT.
1234AXEND
  SDL/SSC =1234 EOJ. ...
  
```


SECTION 25A

SDL2INTRIN/REMOVER

The SDL2INTRIN/REMOVER program is a special purpose utility program that is used during the changing of the SDL2 Intrinsic file. The purpose of the program is to mark an old version of the SDL2INTRIN/AGGREGATE file as unrestricted so that the file can be removed from the system. It is also used to mark the new version of the SDL2INTRIN/AGGREGATE file as restricted to prevent its removal from the system.

OPERATING INSTRUCTIONS

When loading a new SDL2 Intrinsic file, execute the SDL2INTRIN/REMOVER program to mark the current SDL2INTRIN/AGGREGATE file as an unrestricted file, as follows:

EXECUTE SDL2INTRIN/REMOVER;

Remove the current SDL2INTRIN/AGGREGATE file and change the name of the new SDL2INTRIN/AGGREGATE file to the proper name, as follows:

REMOVE SDL2INTRIN/AGGREGATE;
CHANGE SDL2NEW/AGGREGATE TO SDL2INTRIN/AGGREGATE;

Execute the SDL2INTRIN/REMOVER program to mark the new SDL2INTRIN/AGGREGATE file as a restricted system file, as follows:

EXECUTE SDL2INTRIN/REMOVER;

SECTION 26

SSLOAD/MAKCAS

The SSLOAD/MAKCAS program is a normal-state utility program that creates CLEAR/START and stand-alone utility cassettes. (Cassettes can be created only on B 1000 systems with magnetic tape cassette I/O controls.) The cassette system must be in I/O mode when the SSLOAD/MAKCAS program is executed.

OPERATING INSTRUCTIONS

Although the SSLOAD/MAKCAS program creates an unlabeled cassette, the MCP will not attach a cassette file to the program unless a labeled, scratch cassette is available. The operator provides the serial number by using the SN system command.

Example:

```
SN CSA A
```

Program Switches

Program switch settings that govern the execution of the SSLOAD/MAKCAS program are described in table 26-1.

Table 26-1. SSLOAD/MAKCAS Program Switches (Defaults = 0)

Switch	Value	Meaning
0	0	File names are input from the ODT.
0	1	File names are from the file CARDS. CARDS is declared as a card reader file, but it may be modified to be a disk file.
2	0	Cassettes are created from a MIL code file; for example, CLEAR/START.
2	1	Cassettes are created from a stand-alone SDL code file; for example, PACK/INIT.
7	0	The <u>SYSTEM/MLFIRMWARE</u> file is copied to the cassette. This switch setting must be used if SW 2 = 1 (SDL code file) and the cassette might be used on a B 1990 system. The cassette is usable on all B 1000 systems.
7	1	The <u>SYSTEM/MLFIRMWARE</u> file is not copied to the cassette. This setting must be used if SW 2 = 0 (MIL code file) or if it is known that the cassette never will be used on a B 1990 system.
8	0	Normal cassette record breaks performed.
8	1	A cassette record break will occur when a pseudo micro-op of @00F0@ is decoded. This feature is provided for diagnostic programs.
9	0	The cassette is created but not verified.
9	1	The cassette is created and verified, but is not remade on error. This is the recommended setting for this switch.
9	2	The cassette is created, verified, and, on errors, is remade and reverified.

Modes

The program operates in two modes: (1) from MIL code files and (2) from SDL code files.

Creating MIL Program Cassettes

To create a cassette from a MIL code file (CLEAR/START, for example), the file name of the MIL code file is provided either through the ODT (AC or AX command) or from the CARDS file. (See switch 0 in Program Switches.) One cassette is created for each file name that is provided. The SSLOAD/MAKCAS program is terminated by a blank or null input.

Example:

```
EX SSLOAD/MAKCAS; SW 7 1; SW 9 1;
  SSLOAD/MAKCAS=1257 BOJ...
  % SSLOAD/MAKCAS=1257 ENTER FILE IDENTIFIER
  SSLOAD/MAKCAS=1257 ACCEPT.
1257AXCLEAR/START
1257AX
  SSLOAD/MAKCAS=1257 EOJ.
```

NOTE

Use SW 7 = 1 for MIL programs. SW 9 = 1 is recommended for all runs

Creating SDL Program Cassettes

To create a cassette from an SDL code file (PACK/INIT, for example), execute SSLOAD/MAKCAS with program switch 2 = 1. The cassette will contain the SDL program, a loader, an SDL interpreter, and a stand-alone GISMO. Also, if program switch 7 = 0, the cassette will contain a firmware file for loading into the multiline control on a B 1990 if this is needed to provide an ODT interface for the SDL program.

The file name of the SDL code file must be provided through the ODT or the CARDS file. The names of the remaining standard files may optionally be provided by default. When the file names of the loader (L), the SDL interpreter (I), GISMO (G), and the SDL code file (S) are provided through the ODT or the CARDS file, they must be entered in the following format -- free-form, any order, one file name per AX record:

```
L <loader file name>
G <stand-alone GISMO file name>
I <SDL interpreter file name>
S <SDL code file name>
```

How do you get a
SYSTEM/MCFIRMWARE from
a different disk?
File equate ???

Any of the file names may be provided by default; for example:

```
AXS PACK/INIT
AX
```

provides the following default file names:

```
L CASSETTE/LOADER
G GISMO/SA
I SDL/INTERPIU
```

These are preferred because they are standard system software files.

Example:

```
EX SSLOAD/MAKCAS; SWO 1; SW2 1; SW9 1; FI CARDS NAME MYNAMES/A DISK;
SSLOAD/MAKCAS=134 BOJ...
% SSLOAD/MAKCAS=134 LOADER FILE NAME NOT CHANGED - WILL USE
% CASSETTE/LOADER
% SSLOAD/MAKCAS=134 GISMO FILE NAME NOT CHANGED - WILL USE
% GISMO/SA
% SSLOAD/MAKCAS=134 INTERP FILE NAME NOT CHANGED - WILL USE
  SDL/INTERPIU
1257AXCLEAR/START
.
.
%SSLOAD/MAKCAS=134 CASSETTE OF SDL PROGRAM: PACK/INIT CREATED
  FOR B 1000 SYSTEM.
SSLOAD/MAKCAS=134 EOJ....
```

ERROR CORRECTION

If any one of the input L, G, I, or S specification cards is missing or names a non-existent disk file, the SSLOAD/MAKCAS program notifies the operator and asks for the correct file identifier to be entered. The requested specification card, followed by an ?END card, must then be read through the card reader from which the SSLOAD/MAKCAS program is receiving its data.

ERROR MESSAGES

INVALID FILE TYPE <type>

The <type> specified in column 1 of an input specification card is not L, G, I, or S. Correct and re-enter the card.

FILE NOT ON DISK-<file identifier>

The file title could not be located on disk. Make this file available, or request another file, and re-enter the corresponding specification card through the card reader assigned to the SSLOAD/MAKCAS program.

MISSING <file type> FILE

The specification card for the file-type indicated was not found. Read the correct card through the card reader assigned to the SSLOAD/MAKCAS program.

<file identifier> IS NOT A VALID CODE FILE

The specified SDL program is segmented and thus cannot be used to make a stand-alone SDL cassette. Stand-alone programs cannot be segmented.

DISK PARITY ERROR – UNABLE TO VERIFY

The temporary disk file used during verification has a parity error.

VERIFY ERROR REC# <record number>

An error was encountered during verification of the record indicated.

REMAKE NOT REQUESTED

A verification error was encountered, but the cassette was not remade because switch 9 = 1.

INVALID REMAKE – DISK PARITY ERROR ENCOUNTERED

The temporary disk file used for remaking the cassette is bad.

SECTION 27

STANDALONE/DISK-DUMP

The STANDALONE/DISK-DUMP program copies data file-by-file from one disk cartridge or disk pack to another of the same or greater capacity. The program can also copy a larger system or user disk pack to multiple smaller user disk packs, provided that no single file is larger than the capacity of a single output disk pack. The input disk pack may be a single system disk (type S) that is not a continuation part of a multiple system pack, or it may be a user disk pack (type U). Disk cartridges, however, may be user disks (type U) only.

The entire input disk is copied file-by-file to produce a squashed output disk. The original type (S or U) of the output disk does not matter, it will be forced to the same type as that of the input disk with the following exceptions:

1. When the capacity of the input disk is greater than that of the output disk, the output disk is always a type U (user) disk.
2. When the input disk is a fixed (non-removable) type S system disk that does not have a greater capacity than the output disk, the operator is given a choice of a type U or S output disk.

A system disk that has not been CLEAR/STARTed is not allowed as input. Multi-pack files are copied only if the input and output disk serial numbers are the same.

The STANDALONE/DISK-DUMP program corrects certain errors that it detects. This capability can be used to try to recover some types of corrupted disks. An irrecoverable error in a disk directory or file header causes the associated file or files to be skipped, but the other files on the disk are copied. Any missing disk areas on the input disk are ignored when writing to the output disk. Thus, the output disk is fully usable. When copying a system disk, some checking and correction of the disk label and MCP structures is done; however, this should be regarded as a partial fix, not as a replacement for a full COLDSTART.

A STANDALONE/DISK-DUMP program option compares the output disk file-by-file to the input disk so that comparison errors can be identified.

NOTE

When copying a system disk, AB (Auto Backup) coldstart variables are not duplicated on the output disk. A newly copied disk may be taken to another system with different port and channel values for line printers; therefore, the user may want to enter the new values after the CLEAR/START of the newly copied system disk.

Two sets of procedures for program loading follow. The first is for B 1990 systems and the second is for all other B 1000 systems.

Program Loading on B 1990 Systems

To load the program on B 1990 systems:

1. Halt the system by entering the MCP command HALT via the ODT. For dual-processor systems, be sure both processors are halted.

NOTE

Instructions for using the B 1990 Soft Panel in this fashion are hereafter presented in either of the following forms:

Transmit <keyword-1> <keyword-2> ...

<keyword-1> <keyword-2> ... [XMT]

A looping condition could deny access to the ODT. In such cases, press the INTRPT push button. If the system still does not halt, press the HALT/RUN push button.

2. Place the cassette into the cassette reader and ensure that the tape rewinds to BOT.
3. Transmit CLEAR MTR GO.

The cassette automatically rewinds on B 1990 systems.

Program Loading on Other B 1000 Systems

To load the program on all other B 1000 systems:

1. Halt the system using either the INTRPT push button or the console INTERRUPT switch. If the INTERRUPT switch is used, return it to its initial position following the halt. If the system continues to run, press the HALT push button. If the system still continues to run, press the HALT and CLEAR push buttons simultaneously. For dual-processor systems be sure both processors are halted.
2. Place the cassette into the cassette drive and ensure that the tape rewinds to BOT.
3. Press (1) the CLEAR push button, (2) the MODE push button to obtain the MTR mode, and (3) the START push button.
4. After the bootstrap loader is read from the cassette tape, the processor halts. The L register must contain @AAAAAA@ at this time; if not, the cassette must be rewound and the procedure restarted from step 3, above.
5. Press the START push button. Reading of the cassette resumes. When the program is completely loaded, program operation begins.
6. Rewind the cassette tape by pressing the REWIND button. (Rewind is automatic on B 1900 systems.)

NOTES

If the tape does not automatically rewind when it is inserted into the console cassette drive, press the REWIND button. If the tape is wound completely onto the right spool, remove the cassette and manually rewind it until the clear leader is no longer at the read gap.

If the system halts with @000004@ or @0D0053@ in the L register and with the PERP register equal to 1, a cassette error during program loading is indicated. Restart the procedure from step 1.

PROGRAM OPERATING INSTRUCTIONS

After the program is loaded from the cassette, the following messages are displayed on the Operator Display Terminal (ODT):

```
STANDALONE/DISK-DUMP - <mark-and-patch-level> (<compile-date/time>)  
-- PLEASE PUSH SPCFY IF YOU SEE AN ETX ON THIS SCREEN --  
ENTER INPUT DRIVE - <DC? OR DP? OR BLANK FOR EOJ>
```

If there is an ETX character anywhere on the screen, press the SPCFY key once. This is required because of the variety of ODT device types and firmware settings available on the B 1000 family.

After a valid response to ENTER INPUT DRIVE, the input drive <pack-id>, serial number, and type are read from the specified disk and displayed, followed by the message:

```
ENTER OUTPUT DRIVE - <DC? OR DP?>
```

After a valid response, the output drive <pack-id>, serial number, and type are read from the specified disk and displayed, followed by the message:

```
TO VERIFY - IS COPY OF <input-drive> TO <output-drive> DESIRED?
```

A NO response causes the prompt sequence to begin again. A YES response results in one of the following messages or message sequences when the input disk capacity is not greater than the output disk capacity:

1. When the input disk is a user disk:

```
USER PACK WILL BE CREATED
```

2. the input disk is a removable system disk:

```
SYSTEM PACK WILL BE CREATED
```

3. When the input disk is a fixed (non-removable) system disk:

```
DO YOU WANT THE OUTPUT PACK TO BE A 7USER7 PACK? <YES OR NO>
```

A YES response elicits USER PACK WILL BE CREATED, while a NO response results in SYSTEM PACK WILL BE CREATED.

A YES response to the TO VERIFY ... question when the input capacity is greater than the output disk capacity results in one of the following message sequences:

1. When the input disk is a user disk:

```
WARNING: <input-drive> INPUT DISK HAS A GREATER  
CAPACITY THAN OUTPUT DISK ON <output-drive>  
USER PACK WILL BE CREATED
```

2. When the input disk is a system disk:

WARNING: <input-drive> INPUT DISK HAS A GREATER
CAPACITY THAN OUTPUT DISK ON <output-drive>
USER PACK WILL BE CREATED - IS THIS DESIRED?
<YES OR NO>

A YES response elicits USER PACK WILL BE CREATED, while a NO response causes the entire prompt sequence to begin again.

Later, while the dump is being performed, the following messages are displayed when a smaller output disk pack is filled:

NOT ENOUGH OUTPUT DISK ON <output-drive> TO COPY <file-name>
ENTER OUTPUT DRIVE - <DC? OR DP?>

To continue, the operator must make another output disk pack available and answer the prompt with its drive mnemonic.

The prompt sequence now returns to its main path with the following question:

IS DATA COMPARISON DESIRED? <YES OR NO>

A NO response results in a dump with no comparison of output data to input data. A YES response causes the output data to be read after being written and then compared to the input data buffer still in memory. Only data comparisons are made; disk structures such as the directory or file headers are not compared when this option is requested. The following message is displayed following the comparison question:

IS DOUBLE CHECK COMPARISON DESIRED? <YES OR NO>

A YES response causes the input data to be read again and then compared to the output data. This ensures that the correct sector was read when the data was copied. A NO response causes the dump to begin without this function. The following message is displayed and the copy process begins:

DUMPING FILES FROM <input-drive> TO <output-drive>

Upon dump completion without errors, the following messages are displayed:

DUMP COMPLETE <input-drive> TO <output-drive>
ENTER INPUT DRIVE - <DC? OR DP? OR BLANK FOR EOJ>

At this time, the entire prompt sequence can be repeated. Entering a blank input message causes the STANDALONE/DISK-DUMP program to end with the following message:

END STANDALONE/DISK-DUMP

STATUS Command

Once the copy process begins, the operator can interrogate the program status by entering STATUS or ST from the ODT. In response the program displays a message that describes the file currently being copied and the percentage of the disk dumped at that point. The format of this message is as follows:

CURRENT FILE: <file-name> APPROX: <percentage> % COMPLETE

ERROR MESSAGES

The following error messages are returned by this program.

NOTE

If a B 1990 system halts with an irrecoverable exception, the following error messages are displayed on page 2 of the ODT and the Maintenance Card switches the ODT display to page 1 and allows command line input. The operator should look to page 2 for the program error message and, for those halts that specify "press the START push button", transmit GO on the command line of page 1. This is equivalent to pushing the START push button on other B 1000 systems.

INVALID RESPONSE – PLEASE TRY AGAIN

The input message entered was in error. Enter a valid response.

BAD DISK FILE HEADER ADDRESS IN INPUT DIRECTORY ON <input-drive> AT <disk address>

An invalid disk file header was detected for the specified input file. The file is not copied to the output disk, but the dump continues.

NOT ENOUGH OUTPUT DISK ON <output-drive> TO COPY FILE <file-name>

The output disk does not have enough remaining available space to contain the specified file. The STANDALONE/DISK-DUMP program must be restarted using an output disk with a larger capacity.

NOT ENOUGH OUTPUT DISK ON <output-drive> FOR MCP DISK STRUCTURES

The output disk has either a bad Master Available Table or too many XDed sectors to allow MCP required structures to be built. The output disk must be reinitialized (cylinder 0 initialization may be sufficient) or a different output pack must be used.

NO DISK DEVICE ON SYSTEM

No disk cartridge or pack control could be located on the system.

INVALID SUB DIRECTORY SECTOR ON <drive> INPUT DISK – <disk address>

WARNING: SOME OR ALL <directory-name> DIRECTORY FILES

WILL NOT BE COPIED

An invalid subdirectory was detected for the specified <directory-name>. Some or all of the files having that <directory-name> will not be copied.

WARNING: <file-name> IS A MULTI PACK FILE – – CANNOT COPY

The serial numbers of the input and output packs are not the same; therefore, copying a multi-pack file is not allowed.

INPUT DISK ERROR (SECTOR <disk-address> FOR <number of sectors>) FILE <file-name>

OUTPUT DISK ERROR(SECTOR <disk-address> FOR <number of sectors>) FILE <file-name>

A portion of the specified file had irrecoverable errors during the copy process. The copy is completed for the file, but the output data needs to be manually verified to insure integrity. If no file name is displayed, the error occurred during the building of MCP disk structures.

<input-drive> INPUT DISK NOT READY

<output-drive> OUTPUT DISK NOT READY

A not-ready exception was detected on the specified disk unit. The operator must insure that the disk is ready, then press the START push button to retry the operation.

<output-drive> OUTPUT DISK WRITE LOCKOUT

A write-lockout exception was detected on the output disk. The operator must insure that the disk is write enabled, then press the START push button to retry the operation.

DISK NOT PRESENT <drive>

The disk drive specified in the input message could not be located.

PACK LABEL BAD - CANNOT USE <input-drive> INPUT PACK

PACK LABEL BAD - CANNOT USE <output-drive> OUTPUT PACK

The label on the specified pack or cartridge is invalid. The operator must reload and restart the copy using a pack or cartridge with a valid label.

WARNING: IAD FILE <file-name> HAS BEEN COPIED AS A NON-IAD FILE

This message warns that an Installation Allocated Disk (IAD) file on the input drive is copied to the specified disk drive, but that the output copy is no longer marked as an IAD file.

WARNING: <input-drive> INPUT DISK BAD NAME TABLE --

OUTPUT DISK WILL BE COLDSTARTED

This message indicates that the MCF Name Table on the input system disk is invalid or could not be located. The STANDALONE/DISK-DUMP program automatically attempts to perform a disk-to-disk coldstart operation (similar to the COLDSTART/DISK program), in order to create a usable output system disk.

WARNING: COMPARISON ERROR IN FILE <file-name>

- <input-drive> INPUT SECTOR <disk-address> ,

<output-drive> OUTPUT <disk-address> FOR <number> SECTORS

The specified file failed to compare following the copy operation. The output copy of this file should be manually verified to assure integrity.

MISSING REQUIRED SYSTEM FILE <file-name>

PLEASE ENTER THE CORRECT FILE NAME FOR <file-name>

This message occurs when copying a system disk if the specified required system file is not found on the input disk under the default file name, or if the file cannot be copied, or has already been copied as another required system file. The operator must input the name of a file to be used as a replacement.

MISSING REQUIRED SYSTEM FILE GISMO3 OR GISMO2

PLEASE ENTER THE CORRECT FILE NAME FOR GISMO3 OR GISMO2

This message occurs when copying a system disk if the GISMO2 or GISMO3 required system file is not found on the input disk under the default file name, or if it cannot be copied. The operator must input the name of a file to be used as a replacement.

WARNING: <input-drive> INPUT DISK HAS A BAD PACK LABEL -- USER PACK ASSUMED

The label on the input disk is invalid. The STANDALONE/DISK-DUMP program assumes that the disk type is U and continues the copy operation.

<input-drive> INPUT SYSTEM DISK--INVALID OR MISSING COLDSTART VARIABLES--CANNOT USE

The input disk is the continuation part of a multiple system disk configuration, or is a system disk that has not had a CLEAR/START operation performed on it of the same release level as this STANDALONE/DISK-DUMP program.

INVALID MAIN DIRECTORY SECTOR ON <input-drive> INPUT DISK <disk address>

WARNING: NO FILES LOCATED BY DIRECTORY SECTOR <disk address> WILL BE COPIED

An invalid main directory entry has been encountered on the input pack. The dump continues to copy as many of the remaining files as it can.

WARNING: <file-name> IS A PSEUDO-DECK--CANNOT COPY

This file, if copied, would not be usable on the output pack.

<drive> INPUT DISK IS A PRE 9.0 DISK -- CANNOT COPY

The input disk is not a 9.0 or later disk; the dump terminates.

INVALID DISK TYPE

A disk drive unit has responded to a test operation with an invalid unit identification. Try another unit.

**DISK IO TIMEOUT -- IO FAILED TO COMPLETE
DISK IO DISPATCH DENIED(LOCKED OR MISSING DEVICE)**

A catastrophic I/O error has occurred. The entire program must be reloaded and restarted to continue.

- ERROR ON <drive> (PORT <port>, CHANNEL <channel>, UNIT <unit>) RESULT = <disk result descriptor>

This is additional information displayed after some disk I/O errors.

DUMP ABORTED -- THE OUTPUT PACK ON <output-drive> IS NOT A VALID PACK

THE ENTIRE PROGRAM MUST BE RELOADED AND RESTARTED

A fatal error has occurred and the STANDALONE/DISK-DUMP program cannot recover. The program must be loaded again from cassette.

STANDALONE/DISK-DUMP HALTS

The following paragraphs describe possible STANDALONE/DISK-DUMP halts. All these halts, except ODT halts, are preceded by an explanatory display on the ODT. These halts are indicated by lights on the system console or by the the Soft Panel register display on a B1990 system. The L register must contain the value @000011@, and the T register contains one of the following values:

@AAAAAA@

The STANDALONE/DISK-DUMP completed successfully.

@F0F0F0@

The STANDALONE/DISK-DUMP program has aborted. Refer to the ODT for the reason.

FC0000@

There was no ODT found on the system.

@FC1Dxx@

The dispatch to the ODT was denied. xx contains the port and channel number.

@FC1Exx@

Data error encountered on the ODT. xx contains the ODT port and channel number.

@FC20xx@

Could not load the multiline control ODT firmware. xx contains the unit control number indicating the type of failure with the MLC 4/5 firmware load.

@FC2Dxx@

Remote ODT dispatch was denied. xx contains the ODT port and channel number.

@FD0000@

No disk was found on the system.

@FD0100@

Not enough output disk space available.

@FD0200@

An invalid Master Available Table sector has been encountered on the output pack.

SECTION 28

SYSTEM/BACKUP

The SYSTEM/BACKUP program is an MCP utility that enables a user to order the writing (printing and punching) of backup files. A backup file is a disk or tape file whose device type is printer, punch, or backup.

Printing of the WFL Job Log and the backup files associated with it may also be specified through the SYSTEM/BACKUP program. The WFL Job Log is a log file generated by a Work Flow Language (WFL) program. The WFL Job Log and its associated files are on disk.

REFERENCES

For detailed information on the SYSTEM/BACKUP program, see the PB, AB, and AP system commands in Volume 1, section 5 of the *B 1000 Systems System Software Operational Guide (SOG)*. For information on the WFL Job Log, see the B 1000 Systems Work Flow Language Information, Operation, and Functional Description Manual.

Brief descriptions of the files used by the SYSTEM/BACKUP program are provided at the end of this subsection.

Appendix C of this volume contains detailed information on printer/punch backup files.

OPERATING INSTRUCTIONS

The SYSTEM/BACKUP program runs on B 1000 series systems under MCP II control. The program may be executed directly by means of the following command:

```
EXECUTE SYSTEM/BACKUP FILE BACKFILE NAME <backup-filename>
```

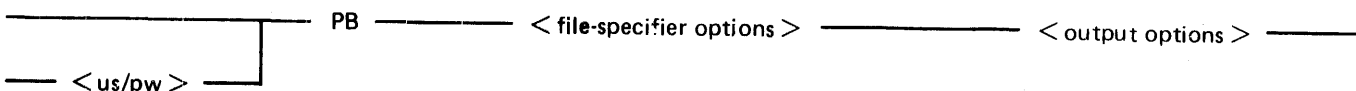
However, execution is ordinarily invoked by means of either of two system commands: PB (Print/Punch Backup) or AB (Auto Backup). All three methods of execution access the same code file.

The SYSTEM/BACKUP program contends for system resources like any other normal-state program, but it has special privileges in obtaining output devices when it is invoked by the AB system command.

PB System Command

The PB system command invokes the writing (printing, punching, or both) of existing backup files or the printing of WFL log files.

Overall Syntax:

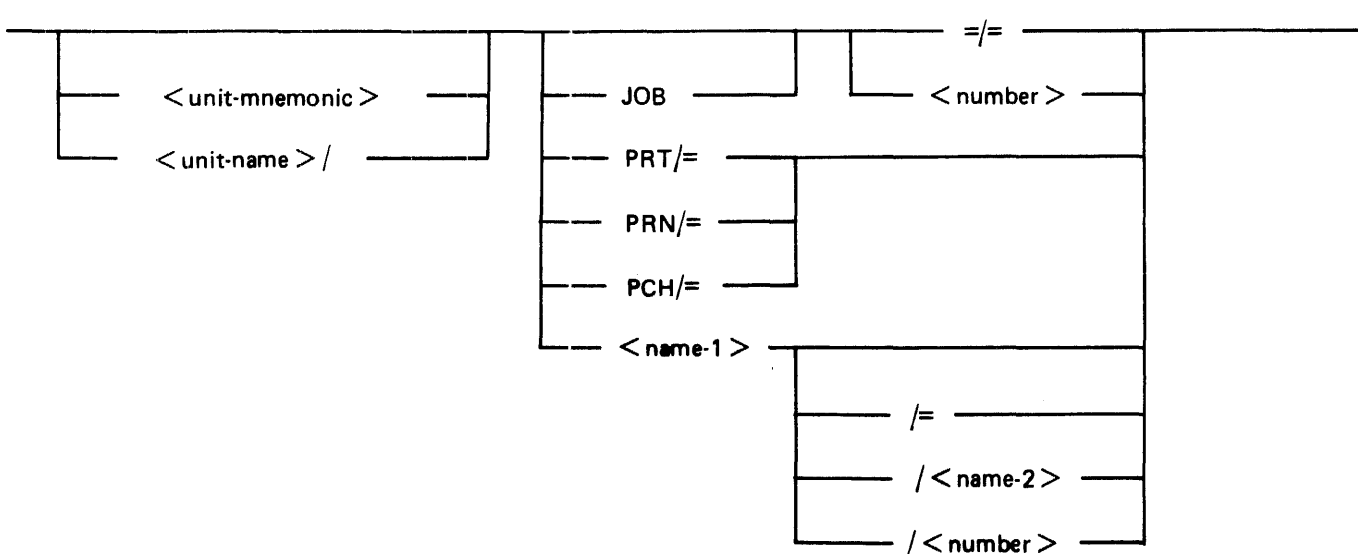


The command includes two types of options:

1. File specifier options for specifying source units as well as source files.
2. Output options for specifying number of copies, line spacing, and similar output parameters.

File Specifier Options

Syntax:



An /= file specifier option causes the writing of all backup files from a specified disk or tape unit. In this case, if both printer and punch files exist on the unit, two copies of the SYSTEM/BACKUP program are executed, one for the print files and the other for the punch files. Output options are not allowed with /=.

PRT/= and PCH/= cause, respectively, the printing and punching of all backup files on a specified unit. (PRN/= is identical to PRT/=.) Output options may be used with these file specifiers.

If JOB is specified, the entire contents of the WFL Job Log (the JOBPRT, summary, and backup files) are printed. If <name-1> is JOBLOG, the summary of the WFL job is printed.

Output Options

Details of the output options are provided in the discussion of the PB command in section 5 of Volume 1 of the B 1000 Systems System Software Operation Guide. Also see Options in this section.

AB System Command

The AB (Auto Backup) system command invokes the MCP autoprint mechanism. This causes a backup file, upon release, to be routed automatically to a line printer. An individual copy of the SYSTEM/BACKUP program is put into execution for each line printer that is assigned to the autoprint facility. Thus, if four printers are assigned, four copies of the SYSTEM/BACKUP program may be in the mix.

The names of the existing printer backup files are entered into a queue file labeled AUTOPRINT. Processing is the same as with a PB system command. The backup files are removed after printing.

The autoprint mechanism can be overridden when a particular backup file is created or whenever it is open by specifying

```
FILE <file-name> NO AUTOPRINT (or NO ATP)
```

A backup file that was created with NO ATP may be printed by means of the AP (Auto Print) system command, or, if a line printer is available that is not assigned to the autoprint facility, by a PB system command.

OPTIONS

Execution-time options for the SYSTEM/BACKUP program include the program switches and the output options available with the PB system command. These options may be preset to take effect when execution begins. Certain other options may be invoked through the entry of AX (or AC) commands during program execution. These are called the run-time control options. The program switches and the run-time control options are described below. The output options are described with the PB system command in in section 5 of Volume 1 of the *B 1000 Systems System Software Operation Guide*.

The following example shows the structure of a PB command that includes an output option (COPIES) and a program switch setting:

```
PB <backup = file> COPIES 4; SWITCH 3 = 1
```

Program switches also may be set or reset by modifying the object code of the SYSTEM/BACKUP program as follows:

```
MODIFY SYSTEM/BACKUP SWITCH 3 = 1;
```

Program Switches

The SYSTEM/BACKUP program recognizes program switches 1, 2, 3, 5, 6, 7, and 8. Reset (0) is the default. The options are described in table 28-1.

Table 28-1. SYSTEM/BACKUP Program Switches

Switch	Value	Meaning
1	0	Carriage control information in the backup file is recognized.
1	1	Single spacing overrides any carriage control in the backup file.
2	0	User-named backup files are not removed after they are printed or punched.
2	1-15	User-named backup files are removed after they are printed or punched.
3	0	All backup files are removed after printing or punching.
3	1-15	All files are saved.
5	0	Lower-case character printing is enabled. The UPPER and LOWER options of the PB command override this setting.
6	0	A simple page-skip is performed when end of page is reached while printing the file.
6	1-15	The end-of-file action specified for the file is performed when end of page is reached.
7	0	When autoprint (AB) is set, SYSTEM/BACKUP goes to EOJ as soon as all backup files in the autoprint queue are printed.
7	1-14	When autoprint (AB) is set, SYSTEM/BACKUP waits up to five minutes for a new backup file to be released before going to EOJ. If a new file is released, the waiting period is reset to five minutes. After five consecutive minutes pass without a new file release, the program goes to EOJ.
7	15	When autoprint (AB) is set, the SYSTEM/BACKUP program waits indefinitely for a new backup file to be released.
8	0	Banners are omitted.
8	1-15	Banners are printed if the file is labeled and the FORM option is not set. The banner includes the usercode (if present) under which the program was executed, the external file name in the creating program, and the actual file name on disk, if different from the external file name.

Run-Time Control Options

The run-time control options enable an operator to make certain inquiries or to enter specifications during program execution. This may be done by entering a job number and an AC or AX command through the ODT. The options available affect only the file currently being printed and override any related options specified with the original execute statement or the PB command.

In the case of a WFL Job Log, the processing of any AC or AX commands is deferred until the printing of the summary is complete.

If more than one copy of the SYSTEM/BACKUP program has been initiated, only the copy which accepts the message (determined by the job number in the AX input message) is affected by the run-time option.

The available run-time control options are described next.

COPIES

Provides a count of the number of copies of a backup file remaining to be printed or punched.

HELP

See TEACH, below.

QUIT or QT

Terminates writing of the current backup file. If any backup files remain, writing resumes with the next file in the queue.

QUIT END or QT END

Terminates the SYSTEM/BACKUP program after writing of the current file is complete. This overrides the five-minute wait option that can be set by program switch 7.

REMOVE or RE

Causes the backup file to be removed from disk after it is written.

SAVE

Causes the backup file to be retained after it is written.

SINGLE, DOUBLE

SINGLE causes single spacing to override any carriage control specified in the backup file. DOUBLE causes double spacing to override any carriage control specified in the backup file.

TEACH or HELP

Causes the display of explanations of all AX input messages and switch settings.

UPPER, LOWER

UPPER causes all lower-case characters to be translated to upper case for printing. LOWER resets the UPPER option to re-enable the printing of lower-case characters.

WHERE

Displays the name of the file, the number of records written thus far, and the end-of-file (EOF) count for the file currently being written.

+ or - <integer>

Causes the spacing of <integer> records forward (+) or backward (-) in the file. A warning is printed on the printer listing. This option is valid only with printer output.

ERROR MESSAGES

Following are all the error messages that may be returned by the SYSTEM/BACKUP program.

<file-id> – EOF REACHED, NO RECORDS PROCESSED.

Either the file had no records or the selection options specified could not be satisfied.

<file-id> – FILE NOT PRESENT.

The specified file is not on disk. Removing a backup file before it is printed or failing to specify file-security information for secured backup files can cause this error.

<file-id> – COLUMN OF ZERO NOT ALLOWED

KEY column is base 1 relative. (See KEY output option.)

<file-id> – KEY LENGTH OF ZERO NOT ALLOWED

KEY option, if used, must be greater than zero.

<file-id> – KEY LENGTH SPECIFIED TOO LARGE

KEY option must not be greater than 10. (Maximum of 9 characters.)

<file-id> – RECORD START OF ZERO NOT ALLOWED

Record count is base 1 relative.

EXCEPTION ON READ – <file-id>

Exceptions are reported until a good read occurs or until the block is exhausted. Bad blocks are skipped but processing continues.

MISSING HOST – <hostname>

Reported when attempting to print or punch a backup file on another system that was not available at open time.

INVALID JOB LOG FILE <file-id> – PROCESSING DISCONTINUED

Reported when attempting to print a WFL log file which contains invalid header records.

CONTINUATION JOB LOG FILE <file-id> MISSING – PROCESSING
DISCONTINUED

Reported when an attempt is made to open a WFL continuation log file but the file cannot be found. (Already removed, for example.)

FILES AND INTERNAL STRUCTURES

Following are brief descriptions of the files used by the SYSTEM/BACKUP program.

NUMBERS

The NUMBERS file is a disk file that is known only to SYSTEM/BACKUP and the MCP. It has a record size of 10 bytes and is blocked 18 records per block. The first record contains two numbers, bits 8-31 and bits 32-55. The first number indicates how many option records follow, and the second number specifies in which option record the <hostname> begins. Each following record contains either a 10-character FILE.ID or 80 bits of encoded option specifications. If the HOSTNAME option is specified, then the given <hostname> spans two option records, and the related options SAVE and NOLIST are specified as the last 2 bits of the second record.

BACKFILE

The file BACKFILE contains control information on the backup disk or tape file to be processed, plus the actual file itself. File equation is included in the MCP-generated execute statement to identify (1) the name(s) of the backup file(s) to be processed, and (2) the input source (tape or disk).

Backup files contain header records and the program-generated backup file. Header records contain control information for the SYSTEM /BACKUP, program, the original FPB, and the output label of the file. Following the header records are the actual user records, each prefixed with carriage control information.

AUTOPRINT

The AUTOPRINT file is a queue file with 31-byte records, two buffers, and a maximum of 150 messages. This file is shared by all copies of the SYSTEM/BACKUP program that are initiated by the AB system command. Each record contains a 30-character backup file name (PACK.ID, MULTI.FILE.ID, FILE.ID) and eight bits of control information.

OUTFILE

The file OUTFILE is the eventual output printer or punch file whose FPB has been written from the FPB contained in the header records of BACKFILE.

LOGFILE

The file LOGFILE contains the WFL Job Log file to be printed. The WFL Job Log is a file that contains a description of the life of the WFL job. The records included in LOGFILE are: (1) BOJ and EOJ records, (2) Open records for printer backup files, (3) Input and Output text records, (4) a Transfer record if the log overflows to a continuation log file, and (5) a Comment record at the beginning of each log file.

SECTION 29

SYSTEM/BUILDTRAIN

The SYSTEM/BUILDTRAIN program is a normal-state utility program that creates translation tables for the B 1247-4 Train Printer Control (control-id @3E@). The generated translation tables are contained in a system disk file labeled SYSTEM/TRAINABLE. Each installation usually needs a SYSTEM/TRAINABLE file that contains the character sets used at that installation.

Refer to volume 1 of the *B 1000 Systems System Software Operation Guide* for information on specifying train printer translation tables.

TRANSLATE TABLE FORMAT

Each translation table generated by the SYSTEM/BUILDTRAIN program consists of 256 one-byte elements. Each element contains the hexadecimal value of the train module link position for the corresponding graphic; thus, byte 193 (zero-relative) in a translator contains the link position for the graphic A (internal EBCDIC representation @C1@ = decimal 193).

As a real example, the 96-character EBCDIC translator (train-id number 016) contains link position values from @00@ through @60@ in the byte positions corresponding to the internal representations of the graphics on the train module. All other positions in the translator contain codes representing the invalid character graphic.

The invalid character code is represented by a hexadecimal character that corresponds to @80@ plus the hexadecimal value of the link position of the graphic that is selected to represent the invalid character in printing. The graphic most often printed as the invalid character is the question mark (?) character. On the 96-character EBCDIC print module, the question mark graphic is located in link position 17 (@11@); therefore, @91@ (@11@ + @80@) is the invalid character code in the 96-character EBCDIC translator.

If a link position value is greater than any link existing on the train module, a PRINT CHECK exception condition occurs. In the 96-character EBCDIC train module, link positions @61@-@7F@ and @E1@-@FF@ result in PRINT CHECK exceptions and, thus, should not be specified in the translator.

OPERATING INSTRUCTIONS

The SYSTEM/BUILDTRAIN program is executed with a standard EXECUTE statement and is controlled by program switches 1, 3, and 4. By default, the file INPUT/PC5-TABLES is used by the SYSTEM/BUILDTRAIN program as an input file. This file must be on disk when the program is executed unless the external identifier of the internal file INPUT is changed by means of a FILE statement. Refer to the *B 1000 Systems System Software Operation Guide, Volume 1*, for additional information.

PROGRAM SWITCHES

The SYSTEM/BUILDTRAIN program recognizes program switches 1, 3, and 4. The function of each switch is described in table 29-1.

Table 29-1. SYSTEM/BUILDTRAIN Program Switches

Switch	Value	Function
1	0	(Default) Creates a new SYSTEM/TRAINABLE file and generates a summary printer listing of the translate tables.
1	1	Generates a summary printer listing of the translate tables, but does not create a new SYSTEM/TRAINABLE file.
3	0	No action.
3	1	Lists the input file on the line printer. Valid only when SWITCH 1 = 0.
4	0	No action.
4	1	Punches the input file as a card deck. Valid only when SWITCH 1 = 0.

INPUT RECORD FORMAT

When building new translation tables, the SYSTEM/BUILDTRAIN program expects the input file labeled INPUT/PC5-TABLES to be located on the system disk. No additional input specification needs to be entered if the standard tables are to be created. The internal name for this translate table specification file is INPUT.

For each translate table to be included in the SYSTEM/TRAINABLE file a set of sixteen (16) input records is required as input to the SYSTEM/BUILDTRAIN program. The format of each of these records is described next.

Columns	Contents
1-20	Train name
22-24	Train-id number
26-28	Character set size
30=31	Printer type (00=400/750 LPM, 01=1100/1500 LPM)
33-64	Link positions (16 per record)
66-67	Sequence number (01-16)
70-80	Optional date (format: 01 JAN 1979)

The train name can be any identifier. The train-id number must be the identification generated by the train module for 1100/1500 LPM printers and is always less than 128. For 400/750 LPM printers, the train-id number can be any value that is greater than 127 and less than 256.

Each link position entry (columns 33 through 64) contains two hexadecimal characters that describe the location on the print train module of the graphic representing the internal EBCDIC configuration. Thus, record #01 gives the link positions for EBCDIC characters @00@ through @0F@, record #02 gives the link positions for EBCDIC characters @10@ through @1F@, and so forth.

Link position entries equal to or greater than @80@ are used to specify unprintable characters. If an internal EBCDIC character translates to a link position equal to or greater than @80@, an INVALID CHARACTER exception result descriptor is returned from the print operation. The actual graphic printed is the specified link position minus @80@.

It is possible to print a graphic other than the question mark as the invalid character by specifying the value of the link position of the desired graphic plus @80@ as the invalid character code. For example, to print the space graphic (which has a link position of @00@) as the invalid character on the 96-character EBCDIC train module, a link position of @80@ (@00@ + @80@) is substituted for every occurrence of the @91@ link position in the input record set.

It is also possible to print a legitimate character in place of the invalid character. This suppresses the generation of the exception result descriptor. For example, to print the question mark graphic as the invalid character (and suppress the reporting of the INVALID CHARACTER result descriptor), substitute @11@ for every occurrence of @91@ in the input record set.

If the resulting link position is greater than the number of characters on the train module, a PRINT CHECK exception result descriptor is returned from the print operation.

STANDARD TRANSLATE TABLES

A set of specifications for the standard printer translate tables is supplied with the SYSTEM/BUILDTRAIN program in a disk file labeled INPUT/PC5-TABLES. The SYSTEM/TRAINABLE file can be generated directly from these standard tables by executing the SYSTEM/BUILDTRAIN program with no modifications, or individual tables can be modified to suit installation requirements.

The five versions of the 96-character EBCDIC translate table are included here as examples of the manner in which specific tables can be generated and tailored to individual installation requirements. EBCDIC96 is the standard 96-character EBCDIC translator, having both upper and lower case graphics. It prints the question mark (?) graphic for the invalid character. EBCDIC96__UPPER__CASE and EBCDIC96__LOWER__CASE also print the question mark graphic for the invalid character; however, the EBCDIC96__UPPER__CASE option prints the alphabet in upper case and the EBCDIC96__LOWER__CASE option prints it in lower case. The EBCDIC96__UPPER__CASEB and EBCDIC96__LOWER__CASEB options function in a similar manner; however, they both print the space graphic (" ") for the invalid character.

Multiple translation tables with the same train-id number (but unique train names) can be contained in the same SYSTEM/TRAINABLE file. The most commonly used version of the translation table should be the first one specified in the input file because it is the table loaded automatically by the MCP II when the printer first becomes ready following a CLEAR/START operation; it is also the table loaded when the train-id number is specified in the LT system command. (See section 5 of Volume 1 of the *B 1000 Systems System Software Operation Guide*.)

Examples:

LT LPA 016

A specific version of such multiple translation tables may be specified by including the train name in the LT system command:

LT LPA EBCDIC96__UPPER__CASEB

Such a translation table remains loaded until the next CLEAR/START operation or until it is explicitly changed by another LT system command.

NOTE

It is not possible to designate a translate table for the 1100/1500 LPM printers in which the train-id number does not match the identification number contained in the train module.

The 400/750 LPM train printers do not have automatic train identification. The proper translation table must be explicitly specified with an LT system command.

The standard printer translation tables supplied are named in tables 29-2 and 29-3.

Table 29-2. 1100/1500 LPM Train Printer

Train Name	ID Number	Description
EBCDIC18	001	18-character EBCDIC
FORTRAN48_NONSTD	002	48-character FORTRAN
B300_B50048	003	48-character B300/B500
EBCDIC48	004	48-character EBCDIC
EBCDIC72	005	72-character EBCDIC
UKB3500_72	006	72-character EBCDIC (UK B3500)
UKB6500_72	007	72-character EBCDIC (UK B6500)
PORTUGAL_72	008	72-character EBCDIC (Latin Portugal)
SPAIN_72	009	72-character EBCDIC (Latin/Spain)
FINLAND_72	010	72-character EBCDIC (Finland)
DENMARK_72	011	72-character EBCDIC (Denmark/OCR-B)
BCL72	012	72-character BCL
TURKEY_72	013	72-character EBCDIC (Turkey)
SWEDEN_72	014	72-character EBCDIC (Sweden)
ASCII72	015	72-character ASCII
ASCII96	055	96-character ASCII
EBCDIC96	016	96-character EBCDIC
EBCDIC96_UPPER_CASE	016	96-character EBCDIC
EBCDIC96_UPPER_CASEB	016	96-character EBCDIC
EBCDIC96_LOWER_CASE	016	96-character EBCDIC
EBCDIC96_LOWER_CASEB	016	96-character EBCDIC
KATAKANA	017	96-character KATAKANA (Japan)
EBCDIC_A72	018	72-character Alphabetized EBCDIC
EBCDIC_N72	019	72-character Numericized EBCDIC
RPG48	020	48-character RPGII
OCR_A72	021	72-character OCR-A
OCR_B72	022	72-character OCR-B
FORTRAN48	036	48-character FORTRAN
THAI144	052	144-character EBCDIC (Thailand)
ASCII96	055	96-character ASCII

Table 29-3. 400/750 LPM Train Printer

Train Name	ID number	Description
FORTTRAN48	130	48-character FORTTRAN
FORTTRAN48__NONSTD	130	48-character FORTTRAN non- standard
B300/B500__48	131	48-character B300/B500
EBCDIC3__48	132	48-character EBCDIC-3
RPG48	140	48-character RPGII
EBCDIC96	144	96-character EBCDIC
KATAKANA	145	96-character KATAKANA (Japan)
EBCDIC3__16	254	16-character EBCDIC-3
EBCDIC3__64	255	64-character EBCDIC-3

SECTION 30

SYSTEM/COMPARE

The SYSTEM/COMPARE program is a normal-state utility program that compares pairs of files that exist on cards, tape, or disk. This program prints and identifies by record number all non-matching record pairs. Such non-identical records are printed in EBCDIC and hexadecimal representations with all differing four-bit hexadecimal digits flagged with asterisks.

The files to be compared may be on any combination of disk, card, or tape media. Should both files be on disk, a special mode of operation can be selected wherein both files are compared sector-by-sector rather than record-by-record, allowing maximum speed.

Other features include automatic termination of comparison after encountering a specified number of errors, interrogation for the number of records currently processed, and explicit termination of a comparison operation at any time.

If the files being compared have differing record lengths, the SYSTEM/COMPARE program emits a warning message and pads the shorter records with blanks.

OPERATING INSTRUCTIONS

Input to the SYSTEM/COMPARE program can be provided in either of two ways. An operator desiring to compare only one pair of files can execute SYSTEM/COMPARE and file-equate the desired files to internal file identifiers A and B. The specified comparison is performed and SYSTEM/COMPARE goes to end of job (EOJ).

Example:

```
EXECUTE SYSTEM/COMPARE FILE A NAME=COBOL;FILE B NAME USERPACK/COBOL/;
```

Use of the SYSTEM/COMPARE program in this single compare mode precludes the ability to use the sector-by-sector mode of operation or the ability to specify the maximum number of comparison errors before termination. A default maximum error count of 15 is used.

If files A and B are not changed by the NAME file attribute of the FILE statement, the SYSTEM/COMPARE program expects input, entered with AC or AX commands. Input is requested after each pair of files has been processed until the program is terminated by the entry of a blank AC or AX command. To simplify use of the SYSTEM/COMPARE program, the input syntax has been made similar to that of the DMPALL program.

PROGRAM SWITCHES

The SYSTEM/COMPARE program recognizes the program switches 0 and 9. Switch 0 specifies the number of characters on the printer chain. Switch 9 changes result information from zero-relative to one-relative quantities.

Switch 0 specifies the number of characters on the printer train. This affects the EBCDIC error listing because characters not printable based on this switch setting are translated to blanks. Users with one printer or multiple printers with the same character set will find it convenient to permanently assign the appropriate value to switch 0, based on the following:

Switch 0 Value	Character Set
0	48-character (default)
1	64-character
2	72-character
3	96-character

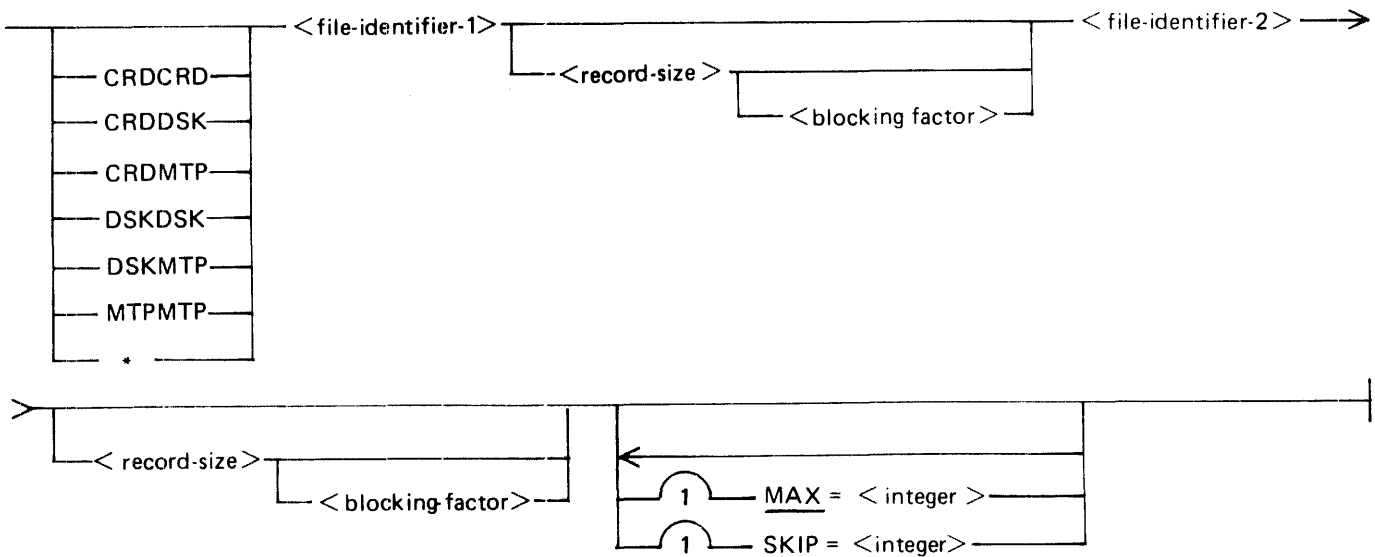
If the value of switch 0 exceeds 3, a 48-character set is assumed.

By default, the SYSTEM/COMPARE program expresses results as zero-relative quantities. This means that the first bit of the first record is defined as bit zero of record zero. However, if switch 9 is set to 1, quantities are expressed in a one-relative manner; that is, the first bit of the first record is now defined as bit one of record one. The appropriate origin is printed as part of the error listing heading.

COMMAND SPECIFICATIONS

Commands must conform to the following syntax:

Syntax:



The shorthand notation for the routine type uses the following abbreviations: DSK (Disk), CRD (Card), or MTP (Magnetic Tape/Cassette). If the routine-type specification is omitted, DSKDSK is assumed. The asterisk is used to specify comparison of two disk files sector-by-sector. A warning message is emitted if a record size or blocking factor is specified for either file in the sector-by-sector mode.

The file-identifiers must be valid file names such as ANNUAL, (CRUCIAL)/DOCUMENT, or USERPACK/"First Try"/ TABULATION. A file identifier whose first component is all numeric, such as 1234/TEST, or is equivalent to a <routine-type> (when none is specified), must be enclosed in quotes to distinguish it as a <file-identifier>. For example, the file-id above must be specified as "1234"/TEST.

Both <record-size> and <blocking-factor> are integer values. If a <blocking-factor> is specified, a <record-size> must also be specified to avoid ambiguity. The <blocking-factor> is normally used only for unlabeled tapes or for blocked card files.

The integer appearing after the keyword MAX specifies the maximum number of errors to list before terminating the current file comparison. The default value is 15. Any value specified must be non-zero.

The integer appearing after the keyword SKIP causes the SYSTEM/COMPARE program to space over the first <integer> records or sectors in each file before beginning the file comparison.

Sample execution strings:

MTPDSK SALES/AUGUST MASTERFILE

* (SOURCE)/QWIKLOG BACKUPCOPY/QWKLOGSRCE

MTPMTP UL_1 80 9 UL_2 80 4 MAX=5 SKIP 150

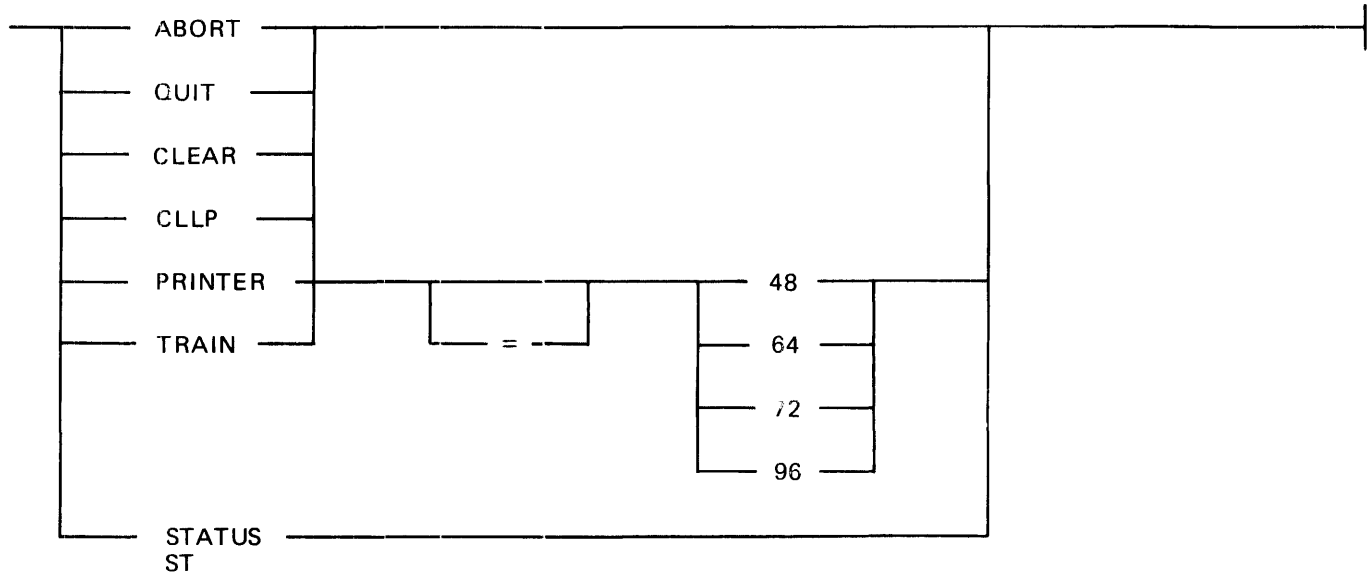
DSKDSK FIRST/USER BACKUP/COPY MAX=1

FORTRAN RELEASE/FORTRAN/ SKIP = 1

RUN-TIME INTERROGATION

Requests for additional comparisons may be made at any time with AC or AX commands through the ODT. Commands are queued if a comparison is currently in progress.

The following run-time interrogation requests can also be provided at any time during execution of the SYSTEM/COMPARE program; these are processed immediately, whether or not additional comparison specifications have been provided.



Semantics

QUIT or ABORT

Terminates current comparison.

CLEAR or CLLP

Closes the printer file LINE to free it for other users.

ST or STATUS

Displays current record and error count.

TRAIN or PRINTER [n]

Displays the value, or changes that value dynamically, of the character set to which mismatched records are translated. Characters not appearing on the specified print train are translated to blanks.

The occurrence of a parity error on either record of a pair being compared causes an informative message to be emitted on the printer file and the comparison of that pair of records to be suppressed.

OUTPUT MESSAGES

The SYSTEM/COMPARE program provides a comprehensive set of error, warning, and informative messages. Error messages indicate invalid operator input and generally include the syntactic token which evoked the error message.

Fatal Errors

Fatal errors represent irrecoverable problems with the execution environment of the SYSTEM/COMPARE program and result in program termination. The fatal error messages and related information follow.

INSUFFICIENT DYNAMIC MEMORY TO EXECUTE SYSTEM/COMPARE; RERUN WITH ME GEQ 4320 BITS

The SYSTEM/COMPARE program makes substantial use of dynamic memory; 4320 bits are necessary as a minimum value. However, there is no increase in efficiency if it is run with a higher value unless the program specifically requests a greater amount.

SOFTWARE INCOMPATIBILITY: <level> SYSTEM/COMPARE; <level> MCP...EOJ

The SYSTEM/COMPARE program is concerned with certain MCP II structures that may change between software releases. In order to ensure correct execution, the SYSTEM/COMPARE program verifies that it is being run under control of the proper MCP II and terminates with this message in case of a mismatch.

SYSTEM ERROR: VARIABLE RANDOM FILES - DS OR DP.

Take a program dump and submit the dump with KA listings of the two input files to a Burroughs technical representative.

Errors

The following error messages identify incorrect operator input and cause the SYSTEM/COMPARE program to ignore that input message.

"48", "64", "72", OR "96" EXPECTED HERE

The value specified for the PRINTER (or TRAIN) option is not one of the four valid integers. Examples:

TRAIN = 56

PRINTER = 74

INSUFFICIENT DYNAMIC MEMORY TO COMPARE <file-name> TO <file-name>; RERUN WITH ME GEQ <integer> BITS.

The SYSTEM/COMPARE program builds its input buffers in dynamic memory; comparison of files with unusually large record sizes results in a need for more than the default 4320 bits. The value needed to perform a given comparison is 16 times the larger record size in bytes. The default value thus suffices to compare files up to 270 bytes (4320 divided by 16) in record length. Users who have frequent need to compare files with longer records are advised to modify the SYSTEM/COMPARE program to the minimum appropriate value for dynamic memory using the MEMORY program attribute.

INTEGER EXPECTED FOR <error limit>

Following the keyword MAX, and the optional equal sign, the SYSTEM/COMPARE program scanned a non-numeric token. Example:

CRDCRD PUNCHED INPUT MAX = TEN

INTEGER EXPECTED FOR <skip value>

Following the keyword SKIP, and the optional equal sign, the SYSTEM/COMPARE program scanned a non-numeric token. Example:

* WITH.INTRO DOCUMENT SKIP MIX = 5

INTEGER EXPECTED FOR TRANSLATION ID

A non-numeric token was scanned after the keyword PRINTER (or TRAIN), aside from the optional equal sign (=) character. Example:

TRAIN = EBCDIC

<file-id> IS AN EMPTY DISK FILE

The SYSTEM/COMPARE program does not process a disk file having an end-of-file (EOF) record of zero.

"MAX", "SKIP", OR <EMPTY> EXPECTED HERE

The SYSTEM/COMPARE program parsed the second file name, and a record size and blocking factor if present, but then scanned an invalid keyword. Example:

MTPMTP DATA BACKUP 45 8 MSX = 1

MISSING FILE ID'S

No input beyond the <routine type> was provided. Example:

DSKDSK

MISSING FINAL QUOTE IN FILE NAME

An unmatched pair of quotation marks was found, which presumably was intended to delimit a <file identifier> with special characters. Example:

DSKMTP PASSBOOK/ACCOUNTS "5%INTEREST MAX = 3

MISSING 2ND FILE ID

The input command specifications contained only one <file identifier>. Example:

* ONLY/ONE

NAME > 10 CHARACTERS

The MCP II permits a maximum of 10 characters in each component of a <file identifier>; the SYSTEM/COMPARE program enforces this limit. Example:

CRDDSK BANKBOOK PASSBOOK__SAVINGS/ACCOUNT SKIP 10 MAX = 1

NAME CAN'T START WITH SLASH

A <file identifier> erroneously started with the slash (/) character. Example:

CRDDSK CARDDECK /DISK/COMPARISON

NO DISK FILE <file-id>

The disk file specified was not present. Note that this message terminates a comparison attempt for a disk file only. A NO FILE condition on card or tape causes the MCP II to request the appropriate file.

NON-CONGRUENT AREAS ON RANDOM FILES

This message can appear only for random disk files, meaning files with interspersed empty areas. Either the records per area value differed for the two files, or the pattern of allocated and unallocated areas did not match.

**RANDOM DISK FILE <file-id> CAN BE COMPARED ONLY TO ANOTHER
RANDOM DISK FILE**

It is meaningless to attempt to compare a disk file having interspersed unused areas to a serial type file, disk or otherwise.

**VARIABLE-RECORD DISK FILE <file-id> CAN BE COMPARED ONLY
TO ANOTHER VARIABLE-RECORD DISK FILE**

The specialized format of variable records makes it illogical to attempt comparison to a file with fixed-length records.

Warning Messages

The following warning messages alert the operator to an action or decision by the SYSTEM/COMPARE program.

COMMAND QUEUE FULL...SPECS IGNORED

More than 10 commands queued while a comparison is in progress. In the unlikely event that this tank size is insufficient, the file TANK can be file-equated or modified to a higher value with the QUEUE.MAX.MESSAGES file attribute. Example:

EXECUTE SYSTEM/COMPARE FILE TANK QMX = 15;

EXTRANEIOUS GARBAGE IGNORED

Unrecognized input occurred in a valid query (non-command) ACCEPT input message. Example:

STATUS PLEASE

PREVIOUS SKIP/MAX CLAUSE IGNORED

A duplicate SKIP or MAX clause was scanned. The SYSTEM/COMPARE program uses the last clause entered. Example:

* STATISTICS CURRENT/(SYSTEM)/DATA MAX 5 SKIP 400 MAX = 3

RECORD AND/OR BLOCK SIZE SPECS IGNORED IN SECTOR MODE

When processing disk files sector-by-sector, the actual record and block sizes of those files are superfluous and irrelevant. Example:

* FIRST/FILE 50 3 SECOND/FILE/COMPARED

"SKIP" SPEC IGNORED FOR RANDOM DISK FILES

The meaning of skipping records in a file with empty disk areas cannot be satisfactory to all users. The SYSTEM/COMPARE program avoids the problem by not using a SKIP specification.

UNEQUAL RECORD SIZES: <file=identifier=A> = <record=size=A>;
<file=identifier=B> = <record=size=B>

The files to be compared had unequal record sizes. The SYSTEM/COMPARE program pads the shorter records with blanks.

VALUE OF ZERO IGNORED IN (MAX/SKIP) CLAUSE

The integer provided with these two keywords must be positive; a SKIP of zero records is the default, whereas a maximum error count of zero has no logical validity.

Informative Messages

<integer> CHARACTER TRANSLATION ENABLED

This message results from either a query or dynamic translation change using the PRINTER (or TRAIN) command.

Examples:

```
TRAIN
```

```
% SYSTEM.COMPARE =3744 48 CHARACTER TRANSLATION ENABLED
```

```
TRAIN 72
```

```
% SYSTEM/COMPARE =3744 72 CHARACTER TRANSLATION ENABLED
```

COMPARISON TERMINATED

This message is the acknowledgement of a QUIT (or ABORT) request.

EOF ON <file-id>

The two files being compared had differing numbers of records. The SYSTEM/COMPARE program terminates the comparison when EOF on the shorter file is reached.

*** ERROR LIMIT REACHED ***

The specified (or default) number of comparison errors occurred before EOF was reached.

(<integer>/NO) ERRORS IN <integer> (RECORDS/SECTORS)

The results of each comparison are displayed in this message. This is also the response to a STATUS (or ST) query. Example:

```
NO ERRORS IN 814 RECORDS
```

NO COMPARISON IN PROGRESS

ST, STATUS, QUIT, or ABORT was entered while the SYSTEM/COMPARE program was waiting for operator input with no comparison being performed.

"SKIP" SPEC POSITIONED (<file-id>/BOTH FILES) PAST EOF

The given SKIP specification exceeded the number of records or sectors (as appropriate) in one or both files being compared.

SPECS

The SYSTEM/COMPARE program is idling, waiting for operator input.

SECTION 31

SYSTEM/COPY

The SYSTEM/COPY program is a generalized library maintenance program for the copying and/or comparing of disk and tape files.

OVERVIEW

The SYSTEM/COPY program can ADD files from disk and tape sources to disk destinations, COPY files from disk and tape sources to tape as well as disk destinations, and COMPARE files on either or both media. (Output tapes are Burroughs standard library tapes, called COPY tapes.) The program also includes a DIR command that may be used to create a printer file of the directory of any library tape. ADD, COPY, COMPARE, and DIR are MCPII library maintenance commands that implicitly invoke the execution of the SYSTEM/COPY program.

Multiple copy requests can be included in a single COPY statement. This allows the files on a tape to be spread across several packs.

The program provides error messages and terminates if it finds fatal syntax errors in the command specification string. Also, it notifies the operator if the command fails to execute completely as specified, and it specifies recovery options when I/O errors occur.

Program switches provide for (1) operator control of message routing, (2) special character recognition, (3) duplicate file checking, (4) comparison error report generation, and (5) enabling/disabling of the streamer tape mode.

The program may be executed from the ODT, a card reader, or a remote terminal. When executed from the ODT without a usercode, it is a privileged program with access to all files. When executed from a card reader or a terminal without file security, the program has access to public files only. When executed with a usercode from any input device, the program has the same privileges as the usercode under which it is executed, imposes file security on all files that it accesses, and automatically prefixes the usercode in parentheses to the identifiers of files that it creates. (Refer to Appendix D of Volume 1 of the *B 1000 Systems System Software Operation Guide* for detailed information on file security.)

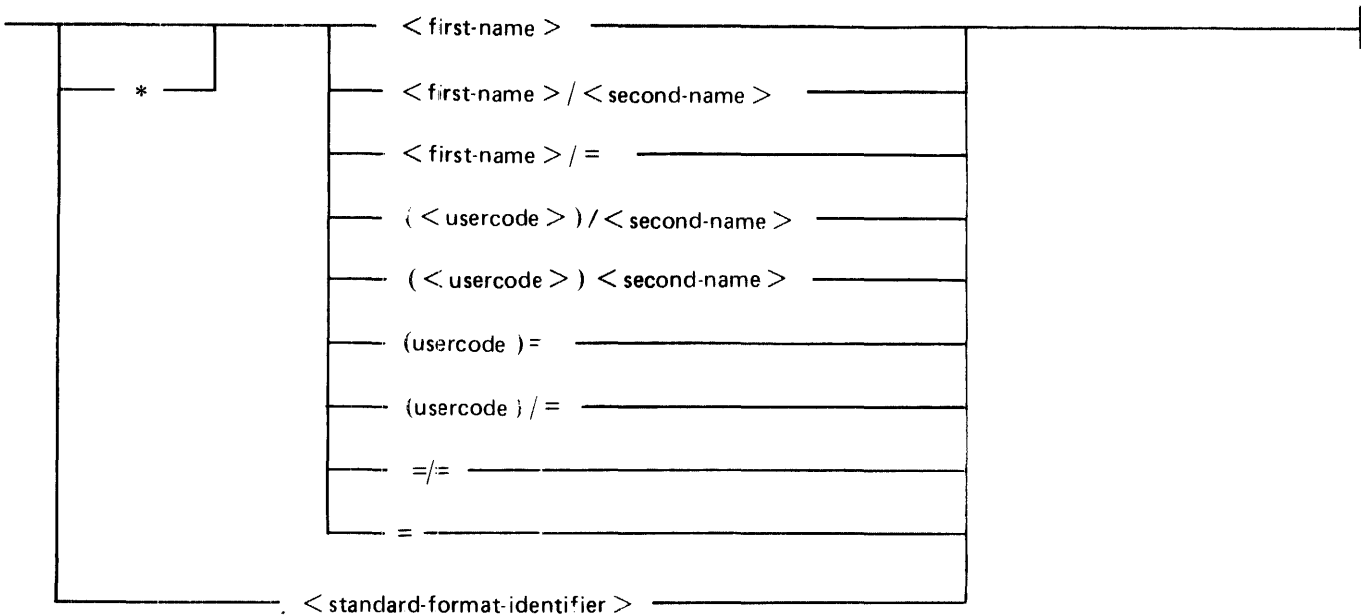
TERMINOLOGY

The following information is pertinent to the proper construction of a SYSTEM/COPY program command string and is particularly applicable to the file-specifications portion of the command syntax. (See Command Syntax.)

File Identifiers

A file-identifier identifies and references one or more files on disk or on tape.

File Identifier Syntax:



Semantics:

*

The asterisk (*) may be used as the first character in a file-identifier to override a usercode and/or the disk associated with that usercode.

<first-name>

A <first-name> is the first part or only part of a <file-identifier>. A <first-name> consists of one to ten characters. Characters accepted by the SYSTEM/COPY program are A-Z, 0-9, and the hyphen (-). When the asterisk syntax is used, the asterisk character (*) counts as one of the ten characters in the <first-name>.

<second-name>

A <second-name> is the second part of a <file-identifier>. It must follow the virgule (/) character. A <second-name> consists of one to ten characters. Characters accepted by the SYSTEM/COPY program are A-Z, 0-9, and the hyphen (-).

<usercode>

A **<usercode>** is a name assigned to a user to secure system and file access. When a **<usercode>** in parentheses is the first part of a **<file-identifier>**, the file belongs to that user. When the command string is executed under the usercode (no parenthesis), all file-identifiers in the string are assumed to belong to that user. The **<usercode>** consists of one to eight characters. Characters accepted by the SYSTEM/COPY program are A-Z, 0-9, and the hyphen (-).

=/=

=/= specifies every file that carries the first name or usercode specified in the first field. **=/=** specifies every file under the family name or volume name specified in a FROM clause. (See FROM Clause, TO Clause later in this section.) Expansion of the last equals sign (**=**) yields **second=names**. Expansion of the first equals sign yields **first-names** or **usercodes**.

<standard-format-identifier>

Identifiers of files on tape must conform to the Burroughs standard format: one to thirteen fields of up to seventeen characters each. Characters accepted by the SYSTEM/COPY program are A-Z, 0-9, and the hyphen (-) when program switch 2 is set (1), but all special characters are accepted when switch 2 is reset (0). Each field is separated from the next by a virgule (/). The field following the last virgule may consist of a lone equals sign.

When any file is copied from tape to disk, the file-identifier format of the copied file must conform to the disk file-identifier format depicted in the upper part of the syntax diagram and described in the preceding semantics paragraphs. The AS Option is used to create a conforming identifier. (See AS **<file-id-2>** option under File Specifications.)

SYSTEM and User Disks

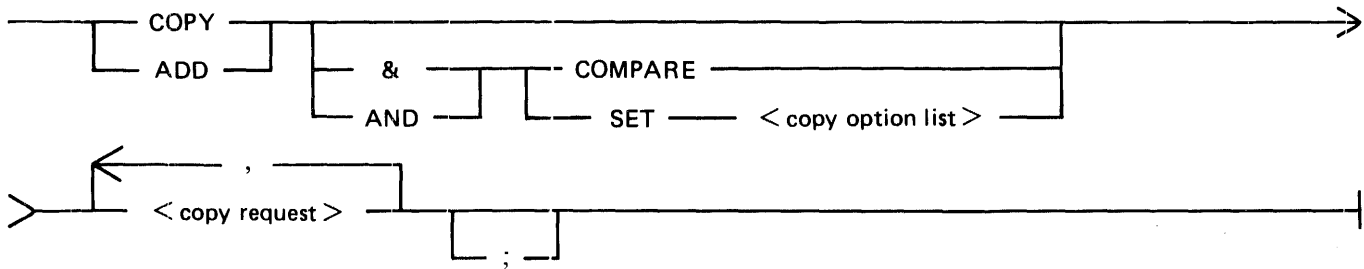
The term "system disk" refers to a disk (either disk pack or head-per-track) that has been initialized for system (MCP) purposes. A system disk may not be removed during system operation. (Refer to the PACK/INIT or SYSTEM/DISK-INIT utility programs in this volume for a complete description of the initialization of a disk pack a system disk.) The system disk is the default disk for unsecured files.

The term "user disk" refers to a disk (usually a disk pack or disk cartridge) that has been initialized for user (rather than system) purposes. A user disk may be removed during system operation; that is, it need not be present for the MCP to function. The default disk for a secured file is the user disk assigned to the usercode.

SYSTEM/COPY PROGRAM COMMANDS

The four basic SYSTEM/COPY program commands are ADD, COPY, COMPARE, and DIR. The syntax and semantics for ADD, COPY, and COMPARE, including the options clause, the file specifications clause, and the FROM and TO clauses are described next. The DIR command is described last.

ADD Command



The ADD command directs the SYSTEM/COPY program to copy files from tape to disk or from disk to disk. (ADD does not copy to tape.) The files are identified through one or more <file-specifications> statements. Copy sequence is as specified for the COPY command (next subsection). File identifiers that have duplicates on the destination disk are skipped. All other files specified are copied.

The AND COMPARE (or & COMPARE) phrase sets the COMPARE option to TRUE. For more information about the COMPARE option, refer to <options> Clause under Command String Clauses in this section.

Examples:

(Only tape sources are shown; disk sources may be substituted to create additional examples.)

ADD A FROM T1 (KIND=TAPE) TO D1;

Copies file A from tape T1 to disk D1 if there is no file A on D1.

ADD A, B FROM T1 TO D1;

Copies files A and B from tape T1 to disk D1 if they are not on D1.

ADD =/= FROM T1 (KIND=TAPE) TO D1;

Copies all the files from tape T1 to disk D1. File names that already exist on D1 are not copied.

ADD =/= FROM T1(KIND=TAPE), A FROM T2(KIND=TAPE) TO D1, TO D2;

Copies all the files from tape T1 plus file A from from tape T2 to disks D1 and D2. File names that exist on either disk are not copied to that disk.

COPY Command

Syntax:

See ADD and COPY Syntax under ADD Command.

The COPY command directs the SYSTEM/COPY program to copy files from disk or tape to tape or disk. When the source is disk, files are copied in the sequence that their names are encountered in the <file-specifications>. When the source is tape, files are copied in their sequence on the source tape.

File identifiers that have duplicates on the destination disk are not skipped. These files replace the files with the same identifiers on the destination disk, and their file directory information is copied into the destination disk directory.

The AND COMPARE (or & COMPARE) phrase sets the COMPARE option to TRUE. For more information about the COMPARE option, refer to <options> Clause under Command String Clauses in this section.

Examples:

(The examples given above for ADD apply to COPY. In the examples below, tape may be substituted for disk as source, destination, or both.)

COPY A FROM D1 TO D2;

Copies file A from disk D1 to disk D2.

COPY A, B FROM D1 TO D2;

Copies files A and B from disk D1 to disk D2.

COPY A FROM D1, B FROM D2 TO D3, TO D4;

Copies file A from disk D1 and file B from disk D2 to disks D3 and D4.

COPY =/= FROM T1 (KIND=TAPE) TO T1 (KIND=TAPE);

Copy a tape with the same name and contents as the source tape. Both tape volume names are T1 at the end of the operation.

**COPY A, B FROM T1 (KIND=TAPE), C, E FROM T2 (KIND=TAPE)
TO P1 (KIND=DISK), D, F FROM T2 (KIND=TAPE) TO P2 (KIND=DISK);**

Note that two of the FROM clauses specify the same input tape volume.

Files A and B are copied from the tape T1.

Files C, D, E, and F are copied from the tape T2.

Files A, B, C, and E are copied to the disk P1.

Files D and F are copied to the disk P2.

COPY I, J TO P1 (KIND=DISK), K, L FROM T1 (KIND=TAPE)
TO P1 (KIND=DISK), TO T2 (KIND=TAPE);

Files I and J are copied from the default disk family. (The FROM T1 does not apply to the file I and J because of the intervening TO P1 clause.)

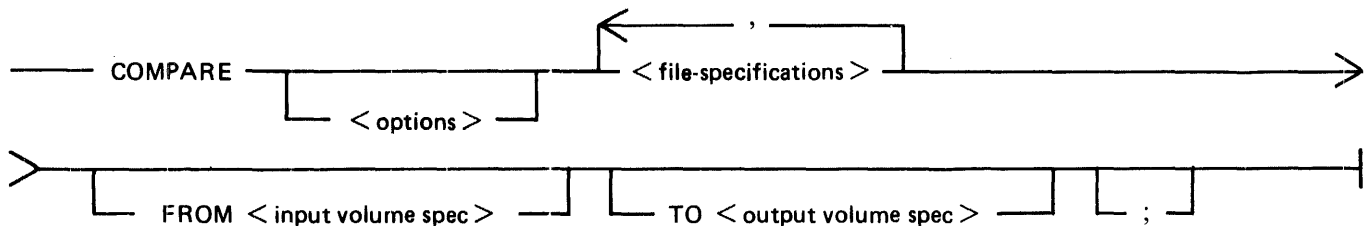
Files K and L are copied from the tape T1.

Files I and J are copied to the disk P1.

Files K and L are copied to both the disk P1 and the tape T2.

COMPARE Command

COMPARE Syntax:



The COMPARE command causes the SYSTEM/COPY program to verify that two files or two sets of files are identical. The headers and data of the source and destination files are compared and a listing is produced with error flags as appropriate.

The output listing (comparison summary) shows the source and copy family or volume name and the hardware type and a list of each file identifier (both identifiers if they are different). If the comparison is error-free, the file type, record and block size, end-of-file (EOF) location, and creation date are provided. For files with comparison errors, detailed information, including byte-by-byte listings of the error areas for both files, is shown.

The operation of the COMPARE command is similar to the operation of the COMPARE option (see Options Clause), with three exceptions:

1. Only one input and one output unit may be specified when using the COMPARE command.
To verify the results of an ADD or COPY from more than one input and/or output unit, either the COMPARE option is used or multiple COMPARE commands are required.
2. The COMPARE command (not the option) must be used to compare tape files that cross reel boundaries. The attribute VOLUMEINDEX (see FROM clause syntax diagram) is used to specify which reel of a multi-reel volume is being used. Thus, the COMPARE command with the VOLUMEINDEX attribute is used if the source file originates on other than the first reel of a multi-reel library tape.
3. The COMPARE command always produces an output listing of the details of the comparison; the option produces the listing only if program switch 4 is set.

Examples:

COMPARE A FROM T1 (KIND=TAPE,VOLUMEINDEX=3) TO D1;

Compares file A on the third reel of the multi-reel input tape T1 to a file with the same name on disk D1.

COMPARE XYZ/= TO D2;

Compares all the files with the first name XYZ on the default disk (see FROM Clause and TO Clause) with files with the same names on disk D2.

COMPARE A AS B, XYZ/= FROM T1 (KIND=TAPE) TO T2 (KIND=TAPE);

Compares file A on tape T1 with file B on tape T2, and compares all the files with the first name of XYZ on tape T1 with files with the same names on tape T2.

Figure 31-1 is a sample listing obtained by means of the following command string:

```
US SOG2/AL COMPARE FROTO AS FRO, SINIT AS SIN, SCOPY AS SCO;
```

The goal is to compare three pairs of files: FROTO and FRO, SINIT and SIN, and SCOPY and SCO, all under the same usercode (SOG2). The FROM and TO clauses are omitted, which causes the SYSTEM/COPY program to go to the default disk. Because the command was invoked under a usercode, the default disk is the user disk (S in this case).

Additional information on the COMPARE function is presented in the description of the COMPARE option. (See COMPARE syntax and semantics under <options> Clause; next.)

Command String Clauses

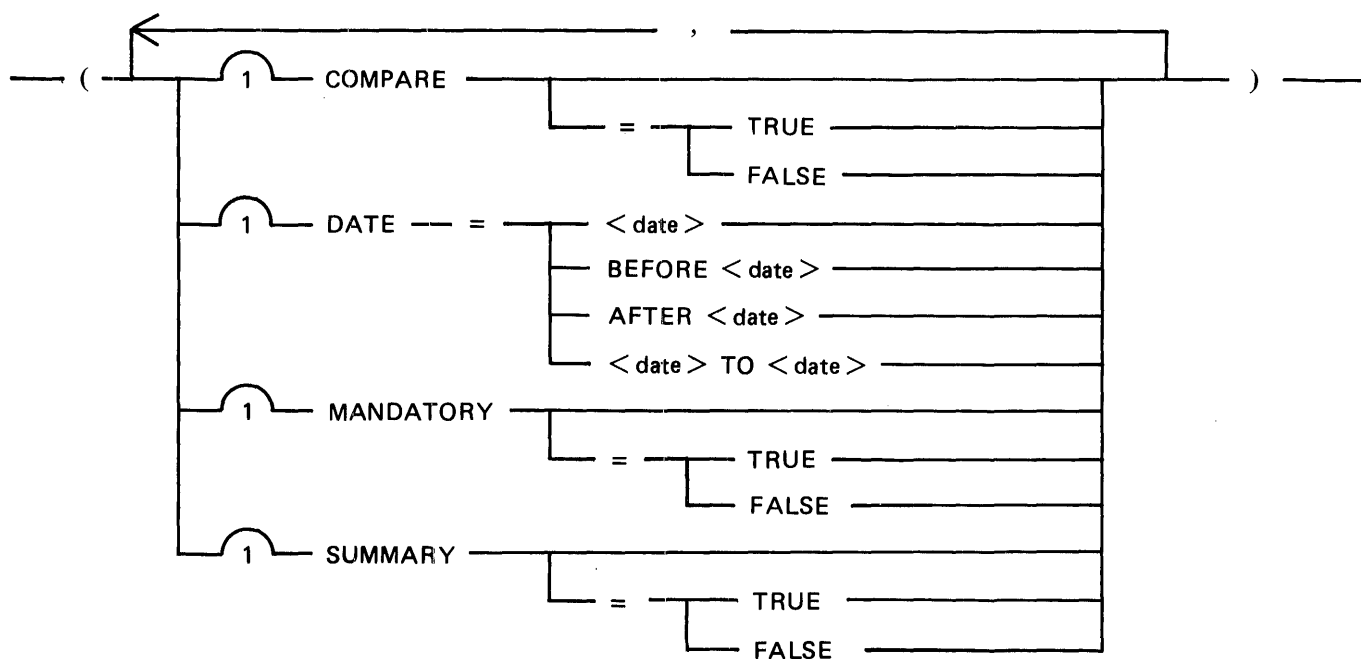
The semantics of the <options>, <file-specifications>, FROM, and TO clauses are described in the paragraphs that follow.

<options> Clause

Available options are COMPARE, DATE, MANDATORY, and SUMMARY. The overall options clause, if included, must precede any file specifications in the command string. The AND (or &) SET conjunction must precede any option, and each option must be enclosed in parentheses.

In all three cases where the option has a TRUE or FALSE condition, TRUE is the default if the option is specified.

Syntax:



Semantics:

COMPARE

The COMPARE option, if included and not FALSE, causes the SYSTEM/COPY program to verify that files have been compared correctly. Comparing takes place immediately after copying each file. The headers and data of source and copied files are read and compared. If program switch 4 = 1, a listing of the compared files with error files flagged is produced unless there are no errors. (This listing is similar to the one shown in figure 31-1.) If program switch 4 = 0 (default), no listing is produced. If MANDATORY = FALSE and a file is missing, the file is not compared and a message is displayed stating that the file was not found. If a comparison error is detected, the operator is notified and given the option to CONTINUE to compare this file, ABORT all operations to this unit, or QUIT the comparison of this file and continue with the rest of the statement. If MANDATORY = TRUE and a file is missing, the operator is notified but must either make the file present or discontinue (DS system command) the program. If a comparison error is detected, SYSTEM/COPY abnormally terminates with the following message:

```
COMPARISON ERROR ON FILE <file title>.  
MANDATORY = TRUE.  
DS OR DP.
```

~~To compare a tape file that crosses a reel boundary, the COMPARE command rather than the option must be used. (See COMPARE command.)~~

Examples:

```
DD & SET(COMPARE) XYZ/= TO D1;  
COPY & SET(COMPARE) XYZ/= TO T1 (KIND=TAPE);
```

Two similar operations. The first copies all files with the XYZ first name on the system disk to disk D1. The second copies the same files to tape T1. In both cases, a file is copied and compared before the next file is copied.

```
COPY AND SET (COMPARE) A AS B, XYZ/= FROM T1 (KIND=TAPE) T  
T2 (KIND=TAPE);
```

Copies an individual file (giving it another name) and all the files with the XYZ first name from one tape volume to another. Compares each file immediately after it is copied.

DATE

The DATE option permits a file date to be used as the selection criterion for copying or comparing. A date or date range is specified and only files with update dates that conform to the specification are selected. (Zeroed update dates mean that the last update occurred before the Mark 9.0 release; in these cases, the file creation date is used.)

Dates must be in the form mm/dd/yy, and the equal sign (=) is required after the word DATE. The following values may be specified:

Value	Files that are selected
<date>	Files updated on the specified date.
BEFORE <date>	Files updated before the specified date.
AFTER <date>	Files updated after the specified date.
<date1> TO <date2>	Files updated between (inclusive) the specified dates.

Example:

```
ADD & SET(DATE = BEFORE 6/1/81) XYZ/= TO D1;
```

From the set of files with the first name XYZ on the system disk; the ones that were created before 6/1/81 are copied to disk D1.

MANDATORY

The MANDATORY option in a COPY, ADD, or COMPARE statement means that the SYSTEM/COPY program either successfully completes the entire statement or is abnormally terminated. For errors such as missing files or packs that an operator can correct, the SYSTEM/COPY program requests operator assistance. The operator may either correct the situation so that the request can continue or may abnormally terminate (DS or DP) the program. For errors that the operator cannot correct (for example, hard I/O errors or comparison errors), the SYSTEM/COPY program is abnormally terminated.

By default (MANDATORY option not included or FALSE), the SYSTEM/COPY program terminates normally even though the entire statement does not complete successfully. Examples include missing files or packs and I/O errors. For file identifiers prefixed with equals signs, at least one file must be found on the specified disk. (See Missing Files, under Error Handling, for more information.)

A COPY AND COMPARE with MANDATORY = TRUE completes successfully even when a file crosses a reel boundary and, therefore, is not compared.

Example:

```
ADD & SET(DATE = BEFORE 6/1/81,MANDATORY) XYZ/= TO D1;
```

This is the previous example (DATE option) with MANDATORY added. If the copy request cannot be completed the program waits for operator intervention. The operator must either correct the situation and enter AX OK or abnormally terminate the program by entering DS or DP.

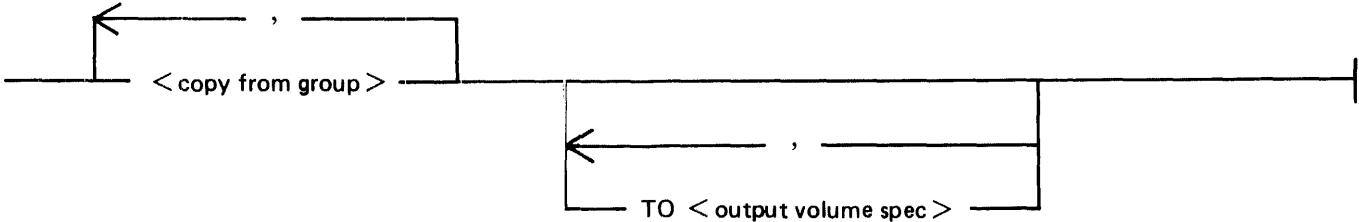
SUMMARY

The SUMMARY option, if included and not FALSE, produces a backup print file of the copy messages and the results of the program run.

<copy request>

A <copy request> is used to identify the files (including the volume on which they reside) that are to be copied and the volume or volumes to which they are to be copied.

Syntax:

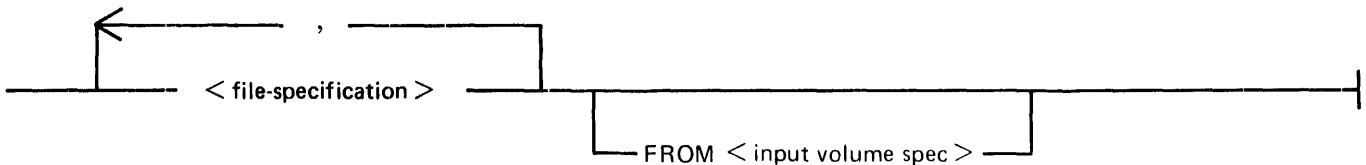


The <copy-from group> is one or a list of files and FROM clauses. Each clause in a list of TO clauses applies to all the <copy-from group>s that precede it in the same <copy request> but not to any <copy from group> separated by another list of TO clauses. When more than one output destination is to be associated with a list of files, the TO clauses must be adjacent. A <copy request> can have a maximum of eight TO clauses.

<copy-from group>

A <copy-from group> is used to identify the files (including the volume on which they reside) that are to be copied to a given volume or volumes.

Syntax:

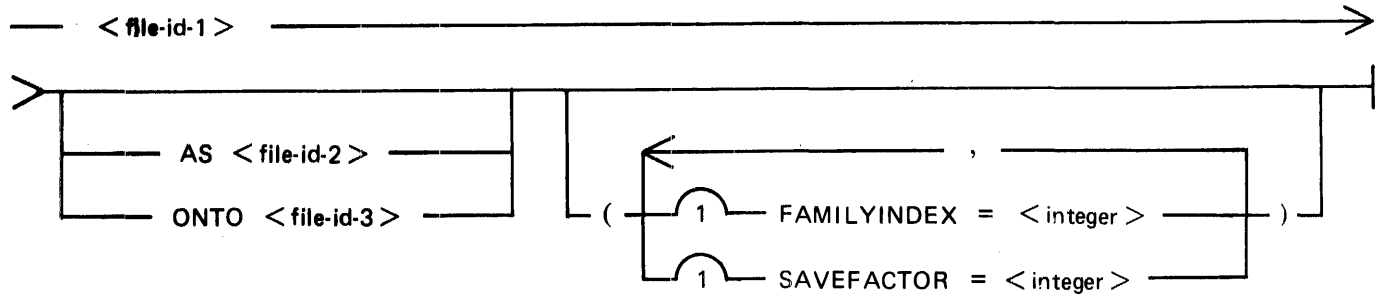


Each FROM clause applies to the list of files preceding it in the same <copy-from group>, but it does not apply to those file or directory names separated by a TO clause or another FROM clause. A copy statement can have a maximum of 32 FROM clauses.

< file-specifications > Clause

File specifications are used to identify the input to the SYSTEM/COPY program; that is, the files that are to be added, copied, and/or compared. The <file-specifications> clause permits the the inclusion of two options and two file attributes.

Syntax:



Semantics:

<file-id-1>

The <file-id-1> must be a valid file-identifier. (See Terminology.) If a family name (disk) or a volume name (tape) is needed, it is always specified in a FROM clause and must not be included as part of the file-identifier.

AS <file-id-2> Option

The AS option is used to give the copied file an identifier distinct from that of the source file. <file-id-2> may not be a <standard-format-identifier>. (See Terminology.)

ONTO <file-id-3> Option

When the COPY command is invoked with the ONTO option specified, the program is directed to copy <file-id-1> to the absolute location assigned by the Installation Allocated Disk (IAD) file specified by <file-id-3>. (If <file-id-3> does not name an IAD file, the SYSTEM/COPY program issues an advisory message and does not perform the copy.) The IAD file and the file specified by <file-id-1> must have exactly the same file specifications or the copy is disallowed. After a successful operation, the IAD file has the same identifier as before, but its file type, EOF pointer, and creation date will have been updated to correspond with the new data in the file.

FAMILYINDEX Attribute

The FAMILYINDEX attribute specifies the device number of the electronic unit of a head-per-track disk on which the file named by <file-id-1> is located. An integer value must be specified; mnemonics may not be used.

SAVEFACTOR Attribute

The SAVEFACTOR attribute specifies the value (number of days) to be entered in the save factor field in the disk file header of the output file. On B 1000 systems, when the number of days from the last access date to the current date exceeds the value in the save factor field, the file is considered as expired and is marked by an asterisk in the directory. Files so marked are not automatically removed, however, and may be copied to other systems.

FROM Clause, TO Clause

The FROM clause specifies the family name (disk) or volume name (tape) that the SYSTEM/COPY program is to use as its input source. The TO clause specifies the family name or volume name to be used for output. Both clauses are optional.

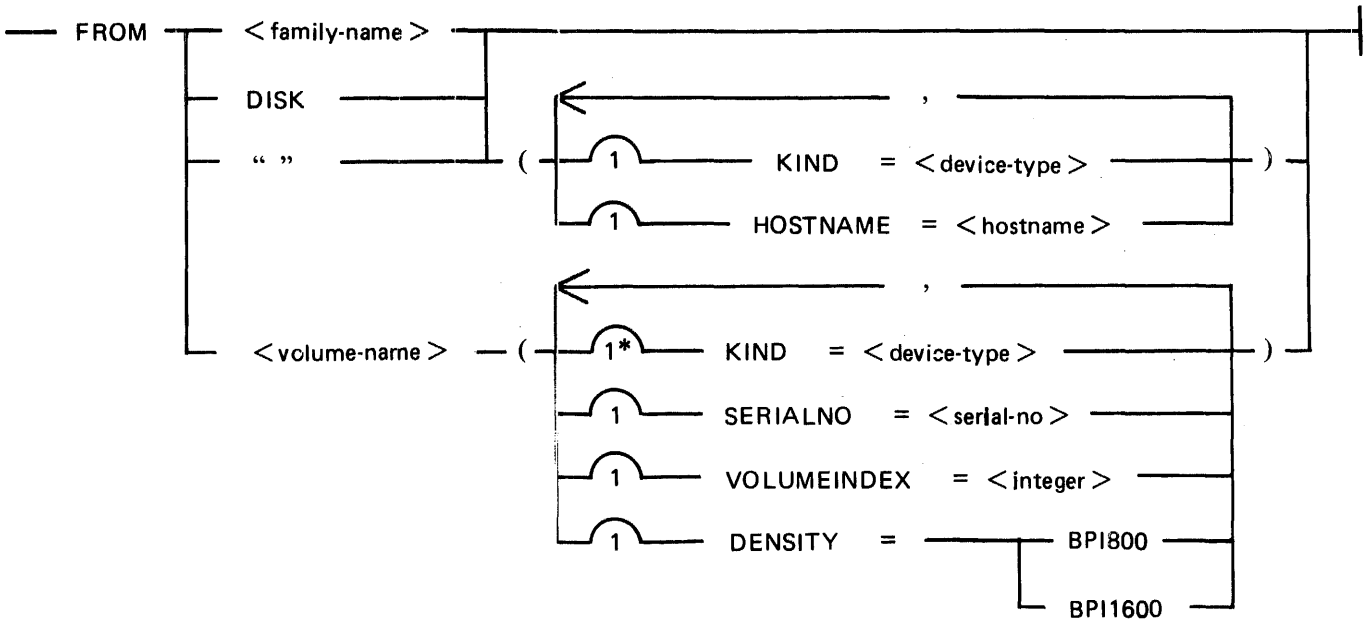
When a disk file identifier has no FROM clause, the SYSTEM/COPY program assigns a default disk as the source, based on the following file-security options:

1. For a non-secured file, the system disk is the default input disk.
2. For a secured file, the user disk assigned to the usercode is the default input disk. However, if a search of the user disk does not locate a specified secured file, the search continues to the system disk.

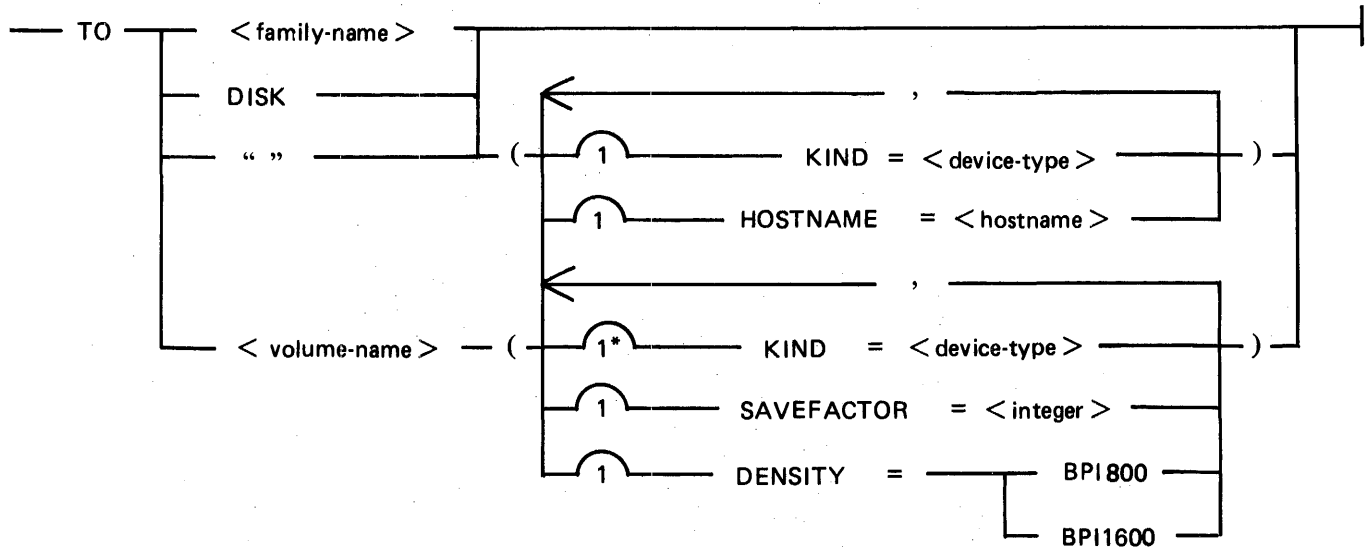
When a disk file identifier has no TO clause, the SYSTEM/COPY program assigns a default disk as the destination, based on the following file-security options:

1. For a non-secured file, the system disk is the default output disk.
2. For a secured file, the user disk assigned to the usercode is the default output disk.

FROM Clause Syntax:



TO Clause Syntax:



Semantics:

<family-name>

The <family-name> is the name of a disk or pack. The name may consist of from one to ten characters. Characters accepted by the SYSTEM/COPY program are A-Z, 0-9, and the hyphen (-). Special characters are accepted only if the <family-name> is enclosed in quotation marks.

A <family-name> may also be the name of a tape volume in some Work Flow Language statements. (See *B 1000 Systems Work Flow Language (WFL) Manual*.)

DISK

Explicitly specifies the system disk.

" "

The construct " " (a space enclosed in quotation marks) may be used to explicitly specify the default disk (system disk for unsecured files; appropriate user disk for secured files).

<volume-name>

The <volume-name> references a library (COPY) tape volume. A tape volume consists of one or more reels and may contain one or more files. Files may cross reel boundaries. Each reel contains a tape directory that identifies every file from the beginning of that reel to the end of the volume.

The <volume-name> may consist of from one to ten characters. The program accepts A-Z, 0-9, and the hyphen (-). Special characters are accepted only if the <volume-name> is enclosed in quotation marks.

KIND = <device = type>

This attribute specifies the hardware device. **KIND = DISK** is optional for disk devices because it is the default condition. One of the following values is required for tape devices:

TAPE	Any tape
TAPEPE	9-track PE tape
TAPE7	7-track tape (NRZ unless otherwise specified)
TAPE9	Any 9-track tape
TAPEGCR	Any group-coded-recognition tape

Example:

```
COPY A TO T1(KIND = TAPEPE)
```

HOSTNAME = <hostname>

The **HOSTNAME** attribute specifies the system on which the specified disk is located in a Burroughs Network Architecture (BNA) network. <hostname> is any 17-character name that is used to specify the local host. (See system command **HN** in section 5 of Volume 1 of the *B 1000 Systems System Software Operation Guide*.)

When the **HOSTNAME** attribute is specified, the **SYSTEM/COPY** program may be executed with the **COPY** command only; **ADD**, **COMPARE**, and **COPY ONTO** are not permitted. Also, file-identifiers with equals signs are valid only for source files that reside on the local host.

Example:

```
COPY A TO D1 (HOSTNAME = LONDON);  
COPY B FROM D1 (HOSTNAME = NEWARK);
```

SERIALNO = <serial = no>

An alphanumeric string of six or fewer characters may be entered as the <serial-no> to match the serial number of a volume (tape). This enables volumes with identical names to be distinguished from each other. The attribute takes effect on input (**FROM**) only; if entered with the **TO** clause, it is ignored. The serial number to be matched is one that has been assigned to the volume by means of the **SN** system command. (See section 5 of Volume 1 of the *B 1000 Systems System Software Operation Guide*.)

Example:

```
COPY A TO T1(KIND = TAPE, SERIALNO = ABC012);
```

VOLUMEINDEX = <integer> (FROM clause only)

The label of each reel of a multi-reel volume includes a volume index (first reel = 1, second = 2, and so on). On input only, the **VOLUMEINDEX** attribute enables the user to specify the desired reel of a multi-reel tape volume. The program uses the attribute value to find the desired reel. The **SYSTEM/COPY** program cannot find a specified reel of a multi-reel volume unless the **VOLUMEINDEX** is included. (See **DIR** Command for more information.)

In cases where the tape directory crosses a reel boundary, the **VOLUMEINDEX** number of the reel on which the directory originates must be specified to access files listed in the directory.

Example:

```
COPY D FROM T1(KIND=TAPE, VOLUMEINDEX=4);
```

SAVEFACTOR = <integer>

Specifies the number of days to be entered in the save factor field of the tape header.

DENSITY Attribute (TO clause only)

For any volume (tape), the **DENSITY** attribute may be used to specify the recording density. Valid B 1000 density values are 800 and 1600 bits per inch (BPI800, BPI1600). **DENSITY** and **KIND** must be compatible.

Example:

```
COPY A TO T1(KIND=TAPE, DENSITY=BPI1600)
```

Examples:

In the examples that follow, file names are specified by the letters A, B, ... , sometimes with a <user-code> prefix. Family-names (disk) are specified by D1, D2, ... , and volume names (tape) by T1, T2, Only the **COPY** action-specifier is used. Commands with other action specifiers (**ADD**, **COMPARE**) are similar.

The first five examples show what happens when the **FROM** or **TO** clauses are omitted. In subsequent examples, usercodes are omitted and the system disk is the default. The user is encouraged to consider the effects of including usercodes.

COPY A AS B

Creates a duplicate of file A and names it file B. Both files are on the system disk.

US JONES/SUE COPY A AS B

Copies file (Jones/A and names it (JONES)/B. Both files are on the user disk assigned to usercode JONES.

US JONES/SUE COPY *A AS B

Copies file A from the system disk and names it (JONES)/B on the user disk assigned to usercode JONES.

COPY A AS B FROM T1(KIND=TAPE);

Copies file A from tape T1 and stores it as file B on the system disk.

The first way is to specify the DIR TAPE command with VOLUMEINDEX = <integer>, as follows:

DIR TAPE "MYTAPE" (VOLUMEINDEX=3);

The <integer> specifies the number of the reel (first reel = 1). The SYSTEM/COPY program goes directly to that reel and prints its directory. (If the tape directory crosses a reel boundary the reel on which the tape directory originates must be specified.)

The second way is used when VOLUMEINDEX has not been specified in the command input string. In this situation, the SYSTEM/COPY program issues the following message:

<tapename>/TAPDIR1977 NOT FOUND

At this point, the program may be dynamically modified with the following commands:

<mix-number> DY FI INFILE REEL <reel number>
<mix-number> OK
<mix-number> IL MTx

If the MANDATORY option is TRUE and the NOT FOUND condition occurs, the program detects the missing tape, displays a message, and waits for the operator to enter <mix#>AX OK or to DS the program. Although the MCP allows a DY command on the file INFILE, the SYSTEM/COPY program resets the tape name before it opens the file. Thus, the DY of the tape name is ignored. Because the program is not waiting for a file, the MCP does not allow an IL.

Figure 31-2 is an example of a tape directory listing.

```

D1000 LIBRARY TAPE DIRECTORY
NAME : SYCOMBKF
KIND : 9-TRACK PE TAPE
SERIAL NO : 13NOV

```

NUMBER	NAME	TYPE	RECORD/BLOCK	EOF	DATE
1	SYCOM/S0	DATA	90/20	142	09/16/81
2	SYCOM/S1	DATA	90/20	149	10/30/81
3	SYCOM/S3	DATA	90/20	1337	10/30/81
4	SYCOM/S4	DATA	90/20	472	09/16/81
5	SYCOM/S2	DATA	90/20	2435	10/30/81
6	SYCOM/S5	DATA	90/20	2824	10/30/81
7	SYCOM/S6	DATA	90/20	525	09/16/81
8	SYCOM/AA	DATA	90/20	625	10/30/81
9	SYCOM/AB	DATA	90/20	43	09/16/81
10	SYCOM/AC	DATA	90/20	1088	10/30/81
11	SYCOM/AD	DATA	90/20	433	09/16/81
12	SYCOM/AE	DATA	90/20	61	09/18/81
13	SYCOM/AF	DATA	90/20	102	10/30/81
14	SYCOM/AG	DATA	90/20	659	09/16/81
15	SYCOM/AH	DATA	90/20	152	09/16/81
16	SYCOM/AI	DATA	90/20	122	10/30/81
17	SYCOM/AJ	DATA	90/20	183	09/16/81
18	SYCOM/AK	DATA	90/20	740	09/16/81
19	SYCOM/DEFINES	DATA	90/20	51	09/16/81
20	SYCOM/MANUAL10	DATA	90/20	12075	11/02/81
21	SYCOM/GAFMAN	DATA	90/20	1387	11/03/81
22	SYCOM/USERS	DATA	90/20	5841	11/03/81
23	SYCOM/INST	DATA	90/20	3241	11/03/81
24	SYCOM/ED	DATA	90/20	6336	11/03/81
25	SYCOMPRT/USLR	BACKUP PRINT	134/5	7768	11/03/81
26	SYCOMPRT/FE	BACKUP PRINT	134/5	8174	11/03/81
27	SYCOMPRT/CAP	BACKUP PRINT	134/5	1655	11/03/81
28	SYCOMPRT/IN	BACKUP PRINT	134/5	4284	11/03/81
29	SYCOMPRT/MANUAL	BACKUP PRINT	134/5	15762	11/12/81

Figure 31-2. Tape Directory Listing from DIR Command

OPERATING INSTRUCTIONS

The following text provides information on preparing and executing the SYSTEM/COPY program. Topics addressed include (1) Program Switches, (2) Operator Input (Examples of Input Strings) (3) Operator Input (AC or AX Messages), (4) SYSTEM/COPY Output, and (5) SYSTEM/COPY Speed Considerations.

The ADD, COPY, and COMPARE commands may be zip-executed from application programs. Use the Work Flow Language system (see *B 1000 Systems Work Flow Language Manual*) or the SYSTEM/LDCONTROL program if the 1500-character limit on command strings executed through the operating system poses a problem.

Program Switches

Program switches 0, 2, 3, and 4 govern the execution of the SYSTEM/COPY program. Table 31-1 describes the valid switch settings. Switch defaults are always 0 (reset).

Switch setting changes must be specified when the command specification string is entered; the program does not respond to changes after it has gone to beginning of job (BOJ). Following is an example of a switch setting change for current execution only:

```
COPY <command specification string>; SW 0 = 1;
```

The object code of the SYSTEM/COPY program may be modified to change program switch settings from the default, as shown in the following example:

```
MODIFY SYSTEM/COPY; SWITCH 0 = 1;
```

Table 31-1. SYSTEM/COPY Program Switches

Switch	Value	Meaning
0	0	Messages are written to the ODT only.
0	1	Messages are written to the printer only.
0	2	Messages are written to the ODT and printer. ("Messages" means syntax and semantic error messages as well as library maintenance messages.)
2	0	Special characters are allowed in tape file identifiers.
2	1	Tape file identifiers may include A-Z, 0-9, hyphen (-) and underscore (_) only.
3	0	First-level (unexpanded) screening of file identifiers.
3	1	First-level and, if necessary, second-level (expanded) screening of file identifiers. (See Duplicate File Identifier Checking.)
4	0	For COMPARE option, no comparison error report is generated.
4	1	For COMPARE option, comparison error report is generated if errors exist.
7	0	Streamer tape mode is not enabled.
7	1	Streamer tape mode is enabled. If a B 9498 tape drive is specified and the file to be copied is 3000 sectors or greater in size, transfer takes place at 100 ips; otherwise, transfer rate is 25 ips.

SYSTEM/COPY Input: Sample Strings

Sample input strings are listed and described in the following subsections.

User Disk to Tape

Assume the following usercode/default-pack-id (user disk) pairs:

ALPHA - D1
KAPPA - D2
SIGMA - D3

Examples:

COPY & SET(COMPARE) (ALPHA)/= FROM D1 TO T1(KIND=TAPE)
Disk D1 is specified explicitly.

COPY & SET(COMPARE) (KAPPA)/= AS (SIGMA)/= TO T2(KIND=TAPE);
The default disks (D2 for user KAPPA and D3 for user SIGMA) need not be specified explicitly.

COPY & SET(COMPARE) A/B, (ALPHA)/C, (SIGMA)/= TO (KIND=TAPEPE);

The default disks (system disk for A/B, D1 for ALPHA, and D3 for SIGMA), need not be specified explicitly.

SYSTEM Disk to Tape

Examples:

**COPY & SET(COMPARE) MCP11, MCP11/MICROMCP, GISMO2, SDL2/INTERP,
SDL2INTRIN/AGGREGATE, SYSTEM/ODT, SYSTEM/INIT, SYSTEM/COPY
SYSTEM/CONTROLLER TO D1;**

All these files are copied and compared from the system disk to disk D1.

COPY & SET(COMPARE) A, B/C FROM DISK TO T1(KIND=TAPE);

Files A and B/C are from the system disk because the keyword DISK is used in the FROM clause. This is also the case when the FROM DISK clause is absent unless the command is executed under a usercode.

COPY & SET(COMPARE) (NEWUSER)/= FROM " " TO D1;

Because there is a usercode, the disk associated with NEWUSER is the default input disk.

Disk to Disk

If a FROM or TO clause is missing and non-secured files are referenced, the system disk is the default disk.

Examples:

COPY & SET(COMPARE) A/= FROM D1 TO D2

Disk=to=disk copy and compare operation, with no defaults.

COPY & SET(COMPARE) A/B FROM D1;

Output goes to system disk.

COPY & SET(COMPARE) A/B FROM D1 TO " ";

As in the preceding example, the output goes to the system disk. Had the command string been executed with a usercode, the " " would have caused the output to default to the user disk associated with the usercode.

Tape to Tape

Example Input:

**COPY & SET(COMPARE) A, B FROM T1(KIND=TAPE), C, D FROM T2(KIND=TAPE)
TO T3(KIND=TAPEPE, TO T4(KIND=TAPE7), TO T5(KIND=TAPE);**

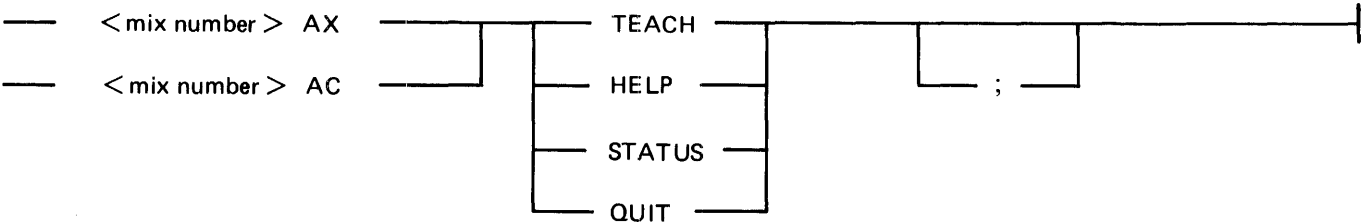
Results:

Input		Output	
Volume Name	File Name	Volume Name	File Name
T1	A, B	T3 (PE tape)	A, B, C, D
T2	C, D	T4 (7-track)	A, B, C, D
		T5	A, B, C, D

Operator Input (AC or AX Messages)

While executing, the SYSTEM/COPY program checks for operator input. Using the MCP ACCEPT mechanism, the operator may make any of the entries shown in the syntax diagram below to query the status of the program. (See AC and AX in section 5 of Volume 1 of the *B 1000 Systems System Software Operation Guide*. AX is used in the text and examples that follow.)

Syntax:



TEACH Command, HELP Command

Entering AX TEACH or AX HELP while the SYSTEM/COPY program is processing causes the list of valid ACCEPT commands to be displayed.

Example:

```
1841AX TEACH
% SYSTEM/COPY =1841 SYSTEM/COPY ACCEPTS THE FOLLOWING COMMANDS:
% SYSTEM/COPY =1841 TEACH / HELP - DISPLAY THESE INSTRUCTIONS
% SYSTEM/COPY =1841 STATUS - DISPLAY CURRENT PROGRAM STATUS
% SYSTEM/COPY =1841 QUIT - DISCONTINUE PROCESSING THE CURRENT
```

STATUS Command

Entering AX STATUS while the SYSTEM/COPY program is processing causes the name of the file currently being copied or compared and the number of records already processed to be displayed.

Example:

```
1843AX STATUS
% SYSTEM/COPY =1843 COMPARING (JMS)/FILE1 FROM M (2940/67 SEGMENTS)
```

QUIT Command

Entering AX QUIT while the SYSTEM/COPY program is processing causes the program to discontinue processing of the current file and go to the next file. If output is to tape, the partially copied file remains. If output is to disk, the partially copied file is removed.

Example:

```
1844AX QUIT
COPY 1844: "(JMS)/FILE1" NOT COPIED FROM "M" AS "(JMS)/FILE2"
COPY 1844 : 1 FILE NOT COPIED
SYSTEM/COPY =1844 EOJ. TIME = 14:53:35.8
```


SYSTEM/COPY Output

The format of a newly-copied file is identical to the format of the source file from which it is created except for pseudo decks. When a pseudo deck is copied, the SYSTEM/COPY program changes the file type from PSEUDO to DATA.

Otherwise, all pertinent file attributes, including CREATION-DATE, are copied from the header of an input file to the header of the output file.

When a file is copied from disk to tape or from tape to disk, only the file attribute specifying the medium is changed. The only exception to this occurs when a file is copied from or to a tape that has been scratched (assigned a serial number with the SN or SNL system command) in a non-EBCDIC mode. In this case, the EBCDIC code, recognized by the B 1000 system, is translated into the format of the code that was specified by the SN or SNL system command when the tape was scratched. (See section 4 of Volume 1 of the *B 1000 Systems System Software Operation Guide* for information on the SN and SNL system commands.)

SYSTEM/COPY Speed Considerations

The SYSTEM/COPY program generally requires more time to copy to disk than to tape because of certain constraints on the number of sectors in a disk output buffer. For output to tape, the number of sectors in a buffer is always 10. (1800 bytes per buffer/180 bytes per sector = 10 sectors per buffer.) For disk, the number of sectors per buffer can be less than 10 because it must be evenly divisible into the number of sectors per area in the disk file, which can vary. As disk buffer size decreases, I/O activity to transfer a given amount of data increases, which is reflected in the time it takes for the COPY to complete.

The system obtains the buffer size that divides evenly by performing modulo operations using the formula $S \text{ MOD } B = R$, where S = sectors per area, B = buffer size in sectors, and R = remainder. The goal, given S , is to find a B that results in $R = 0$. The procedure begins at $B = 10$ and steps down ($B = B - 1$) until $R = 0$ which, in the worst case, would be when $B = 1$.

For example, if a file has 72 sectors per area:

$$\begin{aligned}72 \text{ MOD } 10 &= 2 \\72 \text{ MOD } 9 &= 0\end{aligned}$$

The number of sectors in this buffer is 9.

ERROR HANDLING

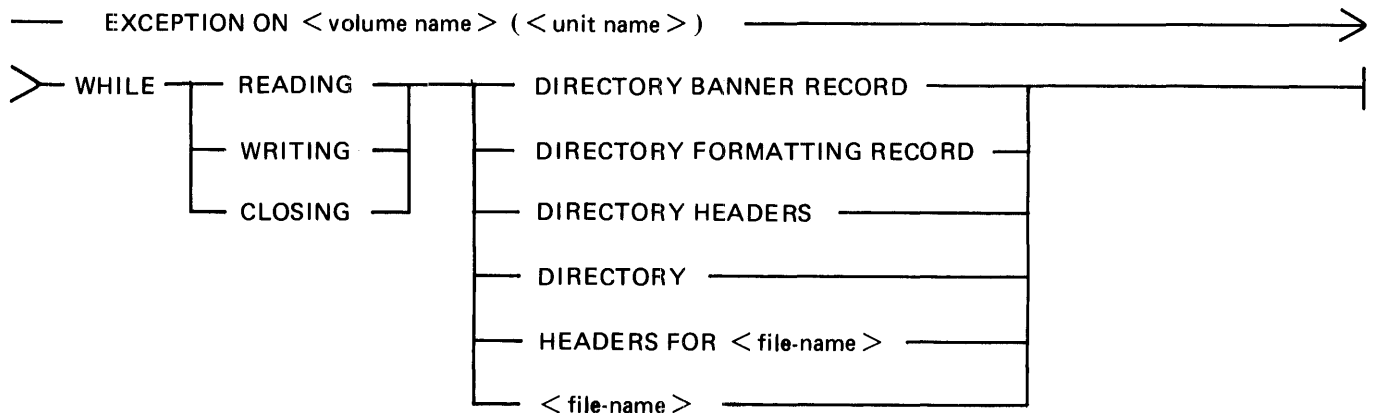
The following paragraphs describe the error handling features of the SYSTEM/COPY program. There are three types of possible errors: I/O errors, comparison errors, and unreliable input errors.

I/O Errors

If the MANDATORY option is TRUE and an uncorrectable I/O error is detected, the program is abnormally terminated with the following message:

UNCORRECTABLE I/O ERROR ON <unit>. MANDATORY = TRUE. DS OR DP.

If the MANDATORY option is FALSE and an I/O error is detected, a message of the following form is displayed:



The SYSTEM/COPY program attempts to recover from I/O errors whenever possible. If it cannot recover, it may abort or it may display the following error-recovery options, and wait for a response:

AX "CONTINUE" TO IGNORE THE ERROR
AX "QUIT" TO ABANDON ONLY THIS FILE
AX "ABORT" TO ABORT ALL COPIES USING THIS UNIT

On parity errors, the SYSTEM/COPY program displays a message that advises the operator of the error and then offers the three error-recovery options. On critical errors, the program advises the operator and then offers only the QUIT and ABORT options.

If the operator selects the ABORT option, the aborted unit is purged and becomes a scratch unit. An abort operation may be performed automatically on an error while writing labels and headers; in such cases, the unit is automatically purged. For example, on an error while writing an output tape directory, the program automatically aborts copying to the faulty tape unit, issues an advisory message, and purges the tape.

If the QUIT option is selected, the program abandons the attempt to copy the file in error. If the file in error is an input file, all output files are abandoned. If the file in error is an output file, only that file is abandoned; copying continues for subsequent files in the input specification list.

All records on COPY library tapes are marked as reliable or unreliable. If the CONTINUE option is selected after an input parity error, the output tape record in question is marked as unreliable. If the QUIT or ABORT option is selected, the last record written to the file is marked as unreliable.

The following message is displayed when a record that was previously marked as unreliable is detected in an input file:

```
[ BLOCK NUMBER <integer> ON ] TAPE <volume name> (<unit name>)  
CONTAINS UNRELIABLE DATA FOR SEGMENTS <integer> - <integer> OF FILE <file-  
name>
```

Comparison Errors

When the COMPARE option (example: COPY AND SET (COMPARE)) is used, the request to compare is ignored if the program encounters any errors while writing a file.

The following comparison error message is displayed on comparison errors:

```
COMPARISON ERROR ON <volume name> (<unit name>) FOR FILE <file-name>
```

When the COMPARE option is used, a printed error report is also produced if program switch 4 has been set to 1. (The printed report is produced unconditionally when the COMPARE command rather than the option is used.)

In-Use Files

When the SYSTEM/COPY program attempts to copy a file to disk and a file with the same name already exists and is in use, the MCP II issues the following message:

```
FILE NOT REMOVED--IN USE.
```

The operator must respond with one of the following system commands before the program can continue:

```
MR (close output file with purge)  
RM (remove duplicate disk file)  
DS (discontinue)  
DP (discontinue and dump memory)
```

If the MANDATORY option is TRUE and the operator enters the MR system command, which would leave the old file present and not complete the copy, the MR system command is processed but the SYSTEM/COPY program abnormally terminates with

```
MANDATORY = TRUE AND MR ENTERED. DS OR DP.
```

When the SYSTEM/COPY program copies a file, it opens the file INPUT LOCK (LOCK ACCESS in COBOL terminology). If the open request is denied, the following action takes place based on the setting of the MANDATORY option: If MANDATORY = FALSE, the SYSTEM/COPY program displays the following message and continues:

<file-name> OPEN OUTPUT OR LOCKOUT, NOT COPIED.

If MANDATORY = TRUE, the following message is displayed:

<file-name> OPEN OUTPUT OR LOCKOUT.
ENTER <mix#>AX OK WHEN FILE IS AVAILABLE OR DS THIS PROGRAM.

An attempt to open a file INPUT LOCK is denied if it is already opened in any output mode or if it is opened LOCKOUT (LOCK in COBOL terminology).

Missing Files

For disk inputs, if the specified disk volume is on-line and the file cannot be found, the advisory message FILE NOT FOUND is issued, no copying takes place and program execution continues unless the MANDATORY option is TRUE; in that case, the program waits for operator intervention. The operator may make the file present and enter <mix#>AX OK or DS the program.

If the specified disk is off-line, the program issues the following message sequence and waits for operator response:

PACK <volume name> IS NOT ON-LINE
PLEASE MOUNT THE PACK AND AX "READY"
OR AX "SKIP" TO SKIP ALL COPIES USING THIS PACK.

If the MANDATORY option is TRUE, the SKIP option is not offered. The operator must either make the pack present and enter the OK system command for the program or DS the program.

For tape inputs, a missing input file condition may arise in either of the following situations:

1. The tape unit is off-line and the SYSTEM/COPY program is waiting for file <tape name>/TAPDIR1977 to be made available. In this case, in order for the program to proceed, the operator must either ready the requested tape or use the IL system command to specify another tape. Either response must make the first reel of a COPY library tape available to the SYSTEM/COPY program.
2. A file identifier is included in the COPY tape directory, but the file has not actually been copied to the tape. This discrepancy between the tape directory and actual tape contents can occur if the program building the COPY library tape completes the directory but is discontinued (DS or DP system command) before it copies all the files. Either of two corrective actions may be taken:

First, First, the operator may enter <mix-number>OF (optional file response) if the ending label of the tape has not been corrupted. This causes the requested COPY of the missing file to be skipped.

Second, the operator may delete the identifiers of the missing files from the command specification and re-execute the command.

Duplicate File Identifier Checking

The SYSTEM/COPY program scans the command specification string for duplicate file identifiers. The screening prevents (1) the same file from being copied redundantly, (2) a destructive copy, which would occur if the same file is specified as both input and output, and (3) a premature copy, which could occur if a file created during a current execution is also specified as an input file in a copy request during the same execution.

The initial scan is done before any equal sign (=) characters in the copy specification are expanded; that is, before /= is investigated to identify all the names represented by the last equal sign. First-level duplicate names (see the following examples) are detected at this time.

```
COPY FILE1, FILE2 AS FILE1 TO T(KIND=TAPE);  
COPY FILE1, FILE2, FILE1 TO T(KIND=TAPE);
```

The second-level duplicate file conditions are detected when the equal-sign expansion is done. (Second-level checking is not done unless program switch 3 has been set to 1.)

When scanning detects duplicate file names, certain rules of precedence are used to determine which copy requests are fulfilled and which are ignored:

1. For disk output files, the last file listed in the command specification string has priority over duplicate files that precede it.
2. For tape output files, the first file listed in the command specification string has priority over duplicate files that follow it.
3. In all other cases, the first file copy request listed in the command specification string has precedence over all subsequent copy requests with duplicate references.

A descriptive advisory message is issued by the SYSTEM/COPY program when duplicate files are encountered during scanning of the command specification string. Requests that would lead to duplicate copies are ignored, but all non-duplicate copy requests are performed without interruption. Because the duplicates are discovered before copying begins, no duplicates are copied and no duplicate file names are placed into the directory.

The advisory messages describing duplicate file errors are listed and described in the subsection entitled Syntax and Semantics Advisory Messages.

ERROR AND ADVISORY MESSAGES

Messages issued by the SYSTEM/COPY program are classified as (1) syntax and semantics error messages, (2) syntax and semantics advisory messages, and (3) library maintenance messages. Only (1) describes fatal errors; (2) and (3) are advisory messages. (The advisory messages and prompts returned in the course of error handling are included and described in context in the Error Handling subsection, above.)

Message Output Control (Switch 0, MCPPII COPY Option)

Program switch 0 controls the output device for all three message categories:

```
Switch 0 = 0:   All messages to ODT.  
Switch 0 = 1:   All messages to line printer.  
Switch 0 = 2:   All messages to ODT and line printer.
```

File LINE may be modified to cause output intended for the line printer to be routed to a backup file, in order to avoid tying up a line printer during the execution of the SYSTEM/COPY program.

For library maintenance only, the MCPPII option COPY overrides SYSTEM/COPY program switch 0. When the COPY option is reset, library maintenance message output is suppressed.

Syntax and Semantics Error Messages (Fatal Errors)

"ADD" AND "ONTO" BOTH SPECIFIED

The ADD command and ONTO option are incompatible requests. The keyword COPY must be used when the ONTO attribute is specified.

"ADD" NOT ALLOWED FOR TAPE

Since scratch tapes (which contain no files) must be used as tape output devices for the SYSTEM/COPY program, the ADD command is an invalid request.

ATTRIBUTES SPECIFIED FOR SYSTEM DISK

The operator must not specify volume attributes for the system disk.

BAD ATTRIBUTE VALUE

The SYSTEM/COPY program encountered a left parenthesis "(" character that was not followed by a recognizable attribute keyword. A common cause of this error is the failure to enclose a usercode in quotation mark (") characters.

BLANK ID IN NAME

A blank is not valid as a file name (example: A/" ").

DISK FILE NAME HAS MORE THAN 2 IDS

The SYSTEM/COPY program recognizes only usercodes and file names as file-identifier parts. The disk family name must be specified in a FROM or TO clause.

DISK FILE NAME HAS MORE THAN 10 CHARS IN ID

To be recognized by the SYSTEM/COPY program, a disk file name must conform to the format accepted by the MCPPII: one to ten characters; special characters only if the file name is enclosed in quotation marks.

EXPANDED NAME TOO LONG

While expanding the equal sign (=) character in a file name, the SYSTEM/COPY program encountered a disk file name with more than two fields, or a tape file name with more than 13 fields. A common cause of this error is the improper use of trailing equal sign (=) characters.

INVALID USE OF ASTERISK

The asterisk (*) character, used on B 1000 systems to assign an exact identifier to a file, has been used in an invalid situation. An asterisk (*) character must immediately precede a file name or a usercode.

MISSING ATTRIBUTE VALUE

The SYSTEM/COPY program could not find a value following an attribute keyword.

MISSING FINAL PARENTHESIS

The SYSTEM/COPY program could not find an ending right parenthesis ")" character to match a left parenthesis "(" character when processing a COPY command specification string.

MISSING KEYWORD

The SYSTEM/COPY program began execution, but could not find a valid command keyword (ADD, COPY, COMPARE, DIR TAPE) at the beginning of the command specification string.

MISSING NAME

The SYSTEM/COPY program expected a file name as input but could not find one. File specifications must precede FROM and TO clauses in the command string.

MISSING VOLUME NAME

The keyword TO or FROM was encountered but was not followed by a volume identifier. When a blank is intentionally used to denote the default disk, it must be enclosed in quotation marks (" "). TO and FROM, if used as family names or volume-names, must be enclosed in quotation marks. The keyword DISK is used to identify the system disk only.

NAME HAS TRAILING SLASH

A trailing (ending) virgule (/) character is invalid in a file name because it another field is expected. Up to thirteen fields are possible with tape file names. Only two fields are allowed with disk file names. In either case, the trailing virgule is wrong.

NUMERIC VALUE EXPECTED

The SYSTEM/COPY program expected a numeric value as a parameter but found a non-numeric value. Mnemonics may not be substituted for device numbers.

NULL ID IN NAME

Neither blanks nor null fields are valid file identifiers. Two adjacent virgules (//) delimit an invalid null identifier.

NON STANDARD NAME FOR TAPE FILE

Tape file names may have a maximum of 13 fields of 17 characters each. Tape file names containing special characters, enclosed by quotation marks or not, are non-standard and are allowed only if program switch 2 is reset (0).

NOT COPIED – SECURITY VIOLATION OR FILE REMOVED

The file was found but could not be opened, either because of a security violation or because the file was removed by another operation.

NOT COPIED – COPY OVER SELF NOT ALLOWED

The copy operation was not done because it would have destroyed the input file

NOT COPIED – FILE IN USE

The file was in use by another program and could not be opened read-only by the SYSTEM/COPY program.

NOT COPIED – ERROR IN GET/CHANGE ATTRIBUTE REQUEST

The copy operation was aborted because a GET or CHANGE attribute request failed. This occurs only when copying to a foreign host.

"ONTO" NOT ALLOWED FOR TAPE

Installation Allocated Disk (IAD) files cannot be created on tapes; thus, the ONTO clause is meaningless, and hence is disallowed.

QUOTED STRING INCOMPLETE

The SYSTEM/COPY program began processing a string beginning with a quotation mark but failed to find an ending quotation mark.

SAME TAPE SPECIFIED AS INPUT AND OUTPUT

The SYSTEM/COPY program never uses a file for both input and output during a single copy request. If different tapes with the same names are to be used for input and output, the SERIALNO attribute must be used to distinguish the tapes. (See FROM Clause, TO Clause.)

TAPE NAME MUST BE IN QUOTES

The SYSTEM/COPY program expects the <volume-name> specified in a DIR TAPE command to be enclosed within quotation marks.

TOKEN TOO LONG

The SYSTEM/COPY program cannot process an input command specification if any words (tokens) in the specification string contain more than 17 characters. Blank characters must be included between the command keywords. For example, the blank characters in the command COPY & COMPARE are required.

UNEXPECTED DELIMITER

The SYSTEM/COPY program unexpectedly encountered a special character that it regards as a delimiting character when processing the command specification string. Any file name, family name, or volume name that includes special characters must be enclosed in quotation (") characters, or the special characters may be interpreted as delimiters. Special characters are defined as all other than A through Z and 0 through 9.

UNEXPECTED TEXT FOLLOWING DIR REQUEST

The DIR TAPE command specification does not allow comments to be included in the specification string. Any comments must be separated from the DIR TAPE command with a percent sign (%) character.

UNKNOWN ATTRIBUTE

The SYSTEM/COPY program was expecting an attribute description but encountered an unknown keyword.

UNMATCHED EQUALS

When the AS clause is specified in a COPY request, the source and destination file names must contain compatible equal sign characters. A trailing equal sign (=) character encompasses all equal sign (=) characters and file identifiers that follow the last name specified. For example, the tape file names A/B/C and A/B/E/F would both be included when A/= is specified as the file identifier. The equal sign (=) characters are compatible; however, in the string COPY A/= AS B/= /C/= the equal sign (=) characters are not compatible. The number of identifiers in the file name can vary. The string COPY A/= AS B/C/= is a valid copy request.

<unit> HEADER LEVEL VERSION MISMATCH

Major software changes made to the SYSTEM/COPY program can result in an incompatibility between COPY library tapes created by different versions of the SYSTEM/COPY program. Information in file headers can prevent access of an incompatible tape when a level mismatch is detected.

<unit> NOT CREATED BY B 1000 COPY

The COPY library tape referenced was not created by the SYSTEM/COPY program on a B 1000 system.

<unit> NOT A LIBRARY TAPE

The tape unit named as an input volume in the command specification string was not created by the SYSTEM/COPY program and is not a library tape.

VOLUME NAME CONTAINS SPECIAL CHARS

When a disk family name contains special characters (any character other than A through Z or 0 through 9), that name must be enclosed in quotation mark (") characters. Tape volume names may contain the characters A through Z, 0 through 9, the hyphen (-), and the underscore (_).

VOLUME NAME HAS MORE THAN 10 CHARS

No family name or volume name may contain more than ten characters.

Syntax and Semantics Advisory Messages

If the MANDATORY option is TRUE, all the following messages are of type one and cause fatal errors.

ATTRIBUTE NOT YET IMPLEMENTED

The SYSTEM/COPY program recognized the specified file attribute as valid; however, the program cannot accept the attribute as input.

"COPY AS" NAME SAME AS ORIGINAL

The copy command specification string requests that a file be copied and given its original name. The AS clause is redundant and is not allowed.

COPY TO SELF IGNORED

A redundant copy was requested.

DESTRUCTIVE COPY TO <unit> IGNORED

The command specification string requested a copy operation that would either overwrite a file used as input elsewhere in the specification string, or attempted to use a file created during this execution of the SYSTEM/COPY program as an input file elsewhere in the command specification string.

DUPLICATE COPY TO <unit> IGNORED

A redundant copy was requested.

DUPLICATE OUTPUT UNIT IGNORED

An output file was listed in more than one TO clause for a single file.

EMPTY ATTRIBUTE LIST

The SYSTEM/COPY program encountered left and right parentheses with nothing between while processing the copy command specification string.

TOO MANY INPUT UNITS – UNIT IGNORED

Only one FROM clause can be associated with an input file.

TOO MANY OUTPUT UNITS – UNIT IGNORED

The maximum number of TO clauses that can be associated with an input file is eight. Multiple phrases must not be used to attempt to circumvent this restriction. For example, the string

COPY A TO T1, TO T2, TO T3, TO T4, TO T5, TO T6, TO T7,
TO T8, TO T9

results in this output message.

TOO MANY SLASHES

A disk file name may include a single virgule (/), and a tape file name may include twelve.

TOO MANY UNITS FOR ONE STMT – UNIT IGNORED

In a single SYSTEM/COPY command string, the combination of family names and volume names in the input plus the output must not exceed 255.

"COMPARE" NOT ALLOWED WITH "HOSTNAME"

"ADD" NOT ALLOWED WITH "HOSTNAME"

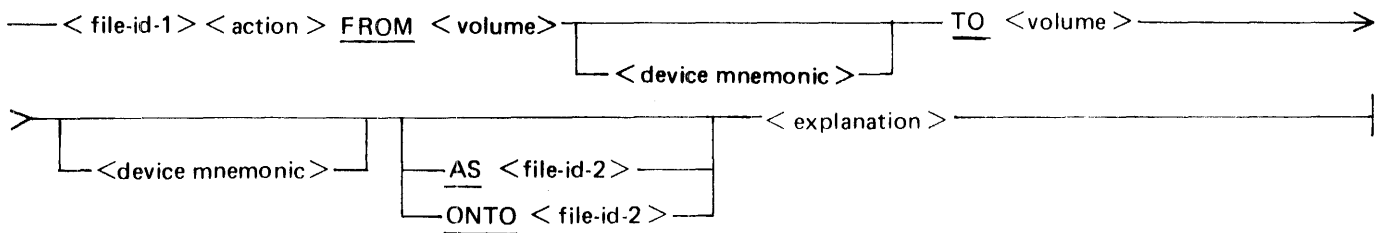
"COPY ONTO" NOT ALLOWED WITH "HOSTNAME"

TAPE VOLUMES NOT ALLOWED WITH "HOSTNAME"

"=" NOT ALLOWED FOR SOURCE FILES NOT ON LOCAL HOST

Library Maintenance Messages

Library maintenance messages are displayed in the following format.



Semantics:

The various words that fit the category <action> are identified next, under Messages Denoting Actions. The words that fit the category <explanation> and further explain the <action> are identified following that, under Explanatory Messages.

Messages Denoting Actions

COPIED

The file named was copied. If no explanation accompanies this message, the file was copied successfully, although there is no inherent implication that the copied file has been compared with the original file. Accompanying explanations can note errors that occurred while copying or comparing the new file.

COPIED AND COMPARED

The file named was copied successfully, and no errors were found during the requested comparison.

NOT COPIED

The file named was not copied. An explanation included in the message describes the reason that the file was not copied.

COPIED BUT NOT COMPARED

The file named was copied but could not be compared although the COMPARE option was specified. Tape files that cross reel boundaries cannot be compared.

Explanatory Messages

NOT FOUND

The SYSTEM/COPY program could not find the file named in the command specification string.

FILE REMOVED BEFORE COPY WAS COMPLETE

The file named was found when the copy operation began. If the copy destination is a library tape, the file name is included in the tape directory, and a dummy file is created on the tape. If the copy destination is a disk, the copy request is ignored.

ONTO FILE NOT FOUND

The IAD file named in the ONTO option of the File Specification clause could not be found, and the copy request is ignored.

INCOMPATIBLE ONTO FILE

The IAD file named in the ONTO option of the File Specification clause is incompatible with the input file. When the ONTO option is used, the input and the output (IAD) file must have identical attributes.

MULTI-PACK FILE

Files that exist as multi-pack files cannot be copied. The copy request is ignored.

TOO MANY UNITS IN USE

Only eight output files can be in use by the SYSTEM/COPY program at a given time, although a maximum of 255 total output units can be named in the command specification. Because the SYSTEM/COPY program accesses each input file only once and makes all requested copies of that file at that time, a situation might arise which requires more than eight output units to be in use by the SYSTEM/COPY program at one time. This is not allowed.

This error is independent of the restriction that a single file can only be copied to eight output units.

REQUIRES INPUT TAPE REWIND

The SYSTEM/COPY program reads an input tape only once. If a copy function requires a tape to be rewound, the SYSTEM/COPY program cannot perform the copy. This error arises when the operator attempts to copy a file from a tape and assigns new identifiers to the copied files. For example, the copy request COPY A FROM TAPE1, A AS A2 FROM TAPE1 generates this error and produces the error message.

BAD FILE HEADER

The input file (<file-id-1>) named in the <action> clause of the error message has a bad file header and cannot be copied.

COMPARISON ERROR

The file named in the <action> clause of the error message was copied, but an error was detected while comparing the input and copied files. Accuracy of the data is questionable.

IO ERRORS

I/O errors (any exception condition) were encountered during the copy. When I/O errors are encountered, the SYSTEM/COPY program offers the operator error recovery options. When this explanation accompanies the COPIED message, it indicates that the operator chose the error recovery option CONTINUE. The records in which the errors occurred are marked as unreliable. When this explanation accompanies the NOT COPIED message, it indicates that the copy of this file was not performed, because either the operator selected the QUIT or ABORT error-handling option, or the I/O errors were irrecoverable.

An MCP/II-generated error message which describes the I/O error in detail precedes this SYSTEM/COPY error message.

NOT COPIED – UNRELIABLE INPUT

The user chose to terminate the copy because the input file contained unreliable data.

Errors When Using HOSTNAME

NOT COPIED – NO USERCODE FILE IN NAME TABLE

NOT COPIED – INVALID USERCODE/PASSWORD PAIR

NOT COPIED – MISSING HOST – <host name>

NOT COPIED – USERCODE NAMING CONVENTION VIOLATED

NOT COPIED – REQUESTED FILE IS AN EXECUTE-ONLY CODE FILE

NOT COPIED – OUTPUT PACK NOT PRESENT

NOT COPIED – ERROR IN GET/CHANGE ATTRIBUTE REQUEST

The copy was aborted because a GET or CHANGE attribute request failed.

COMPARE Errors (Line Printer Listings)

When errors occur during a COMPARE, the SYSTEM/COPY program gives an explanation of the comparison error in a line printer listing. Following are the possible comparison error messages.

HEADER COMPARISON ERROR DATA COMPARISON NOT DONE

Certain size-related fields in the header did not match; therefore, the data was not compared.

HEADER COMPARISON ERROR IGNORED [DATA SUCCESSFULLY COMPARED]

A comparison error in the header was not significant enough to prevent the comparison of the data.

In the following messages, <file> may be a source or destination file.

<file> NOT FOUND

The specified file was not located.

<file> IN USE

The file is being updated by another program.

<file> INVALID HEADER

<file> INVALID DIRECTORY ENTRY

DATA COMPARISON ERROR

<file> IO ERROR

<file> UNRELIABLE TAPE DATA

<file> REQUIRES TAPE REWIND

COMPARISON TERMINATED BY "QUIT"

<file> UNEXPECTED END-OF-FILE

<file> ended before the end-of-file (EOF) record was reached on the other file.

CANNOT COMPARE MULTI-PACK FILES

CANNOT COMPARE VARIABLE FILES

CANNOT COMPARE RELATIVE FILES

All three of the above messages refer to the COMPARE command. To copy variable and relative files, use the COMPARE option. Neither the command nor the option can be used to copy multi-pack files.

<file> IN DIRECTORY BUT NOT ON TAPE

The file was available at the time the tape directory was created but was removed before the data transfer to tape took place.

SECTION 32

SYSTEM/DISK-DUMP

The SYSTEM/DISK-DUMP program is a normal-state utility program that copies data from disk cartridges or disk packs to disk cartridges or disk packs of the same or larger capacity. Neither the input nor the output disk can be in use (have open files) by another program when the dump is processing. The system disk, which has MCP files in use, therefore cannot be copied to another disk with this utility program. Another normal-state utility program, SYSTEM/COPY, will copy file-by-file to or from a system disk that is in use.

The program can perform two types of complete copies (dumps) of disks:

1. A sector-by-sector copy that produces an identical copy of the source disk. The stand-alone utility program DISK/DUMP also performs a sector-by-sector copy. Irrecoverable I/O errors cause the SYSTEM/DISK-DUMP program to terminate.
2. A file-by-file copy that produces a squashed output disk. The output disk is always a user disk (type U). The stand-alone utility program STANDALONE/DISK-DUMP also performs a file-by-file copy. The file-by-file copy allows a disk with a bad label to be copied. When the SYSTEM/DISK-DUMP program is unable to recover from an I/O error, the data in error is copied to the new disk, and the disk sector address of the error is displayed on the operator display terminal (ODT). The user should verify the data at that address and correct or reload as necessary. If the irrecoverable I/O error occurs in the disk directory or in a file header, the associated file or files are not copied.

OPERATING INSTRUCTIONS

The SYSTEM/DISK-DUMP program is executed from the ODT and accepts dump specifications through AC or AX commands. The program issues prompts for all required information.

DUMP SPECIFICATIONS

The SYSTEM/DISK-DUMP program prompts the operator for dump specifications. Operator response must be in the following format:

<mix number>AX<response>

Possible responses are described in the following paragraphs.

INPUT DRIVE

A unit mnemonic that names the drive on which the source disk is mounted. The dump cannot begin while the input disk is in use by any other program. After the completion of each dump, the SYSTEM/DISK-DUMP program repeats the prompt sequence, beginning with the following message:

Prompt: ENTER INPUT DRIVE <DC? OR DP?>

A blank response terminates the program.

COPY TYPE

The COPY TYPE message asks for the file-by-file or sector-by-sector copy type. When the input disk is a User disk, or the input disk has a bad label, the SYSTEM/DISK-DUMP program offers the operator a choice of copy types. A YES response to the prompt requests a file-by-file copy; a NO response requests a sector-by-sector copy.

Prompt: IS FILE BY FILE COPY DESIRED? <YES OR NO>

OUTPUT DRIVE

The OUTPUT DRIVE message asks for a unit mnemonic that names the drive on which the output disk is mounted. The dump cannot begin while the output disk is in use by any other program.

Prompt: ENTER OUTPUT DRIVE <DC? OR DP?>

VERIFICATION

The VERIFICATION message asks whether the optional verification of data will be performed.

Prompt: IS VERIFICATION REQUIRED? <YES OR NO>

PROGRAM OUTPUT

The following paragraphs describe the program output.

Output Disk

After a successful dump, the output disk is the same pack type as the input disk, and the data on the disks is identical, with the exception of the pack serial number and other label information. If a sector-by-sector copy was performed the output data will be in the same sector layout as the input data. If a file-by-file copy was performed, the output disk is a squashed copy of the input disk, and the disk directory has been reconstructed.

Informative Messages

When a file-by-file copy type is selected, the following message indicates that the copy (or copy and verify) is beginning:

BEGIN FILE BY FILE DUMP <AND VERIFICATION>

When a sector-by-sector copy type is selected, the following message indicates that the copy is beginning:

BEGIN SEGMENT BY SEGMENT DUMP

If verification is requested with a sector-by-sector copy type, the following message indicates that the verify pass is beginning:

VERIFICATION BEGINS

After completion of the copy (or copy and verify) and assuming no errors, the following message is displayed on the ODT:

```
DUMP COMPLETE <input-drive> <serial-number>  
TO <output-drive> <serial-number>
```

After the SYSTEM/DISK-DUMP program has completely processed a disk, whether or not the dump was successful, it repeats the prompt sequence. A blank input message at this point will cause the program to go to end of job (EOJ).

NOTE

Neither the serial number nor label information is copied from the input disk to the output disk.

Error Messages

The following error messages can be produced by the program:

DISK ERROR = RESULT = <disk I/O result descriptor>
An I/O error was not corrected after 10 retries.

IS RETRY DESIRED? <YES OR NO>

This message follows the DISK ERROR message. A YES response causes the operation to be retried; NO causes termination of the program.

ENTER DESIRED NUMBER OF RETRIES

Follows an affirmative response to the IS RETRY DESIRED? message. The entry of a blank or zero terminates the program.

INVALID RESPONSE - TRY AGAIN

Either the response to a prompt was other than YES, Y, NO, N, DCx, or DPx, or the requested drive is not available.

TEMP TABLE FILLED - <input-drive>

Temporary areas cannot currently be assigned on the input drive. This problem can be alleviated by performing a CLEAR/START operation.

DISK NOT READY <drive>

Ensure that both the input and output disks are ready and enter "<mix-number> OK".

WRITE LOCKOUT <output-drive>

Remove the write lockout on the output disk drive and enter "<mix-number> OK".

REMOVED SECTORS ON <drive>

A disk with removed sectors must be copied in the file-by-file mode. This message indicates that a sector-by-sector copy was requested.

PACK LABEL BAD - <drive>

If this message names the input drive, a file-by-file copy must be requested. If the message is for an output drive, the output disk must be initialized.

I/O ERROR - <drive> <disk-address>

A parity, time-out, address parity, extended result descriptor, or address error occurred and all retries failed.

RESULT = <disk I/O result descriptor>

Follows the I/O ERROR message and indicates the type of error.

<file-id> IS A MULTI PACK FILE AND CANNOT BE COPIED

If the serial numbers of the input and output packs do not agree, a multi-pack file cannot be copied.

BAD MAIN DIRECTORY <disk-address> <input-drive>

Invalid input directory at indicated address. No file whose directory entry is within that segment will be copied.

BAD SUB DIRECTORY <disk-address> <input-drive>

Invalid input subdirectory at indicated address. No file whose directory entry is within that subdirectory will be copied.

BAD HEADER <file-id> <disk-address> <input-drive>

Invalid input disk file header found; the associated file is not copied.

BAD SECTOR - <drive> <disk-address> <file-id>

The data contained in this sector of disk could not be read or written without the occurrence of an exception condition.

WARNING - <file-name> IS NOT AN IAD FILE

The input file was an Installation Allocated Disk (IAD) file, but the output file will not be an IAD file.

SECTION 33

SYSTEM/DISK-INIT

The SYSTEM/DISK-INIT program is a normal state utility program that labels, initializes, verifies, and reconfigures disk packs and disk cartridges for use on B 1000 systems. The SYSTEM/DISK-INIT program also relocates bad sectors on B 9494-5, B 9494-10, and B 9484-12 type disks. without the loss of any data. This program is very similar to the PACK/INIT stand-alone cassette utility program, except that it executes under MCP control and thus allows other programs to execute in a multi-programming environment. The resulting pack types are System (S), User (U), or Interchange (I).

All disks must be initialized before they can be used with the B 1000 system software. Fixed disks such as the B 9499-6, B 9494-5, and B 9494-10 are initialized during the manufacturing process. A B 9499-6 disk should not need more than a cylinder 0 initialization on site. The B 9494-5 and B 9494-10 need only a Factory Convert (FC). Whenever a factory-initialized disk must be initialized on site, the Cylinder Initialization (CI) or Reinitialization (RI) actions should be used, if possible, to preserve the factory-determined relocation of bad sectors.

During initialization, the SYSTEM/DISK-INIT program formats each sector by writing addresses, sync bits, data patterns, and error protection codes, and then reads the addresses and data to check for errors.

If five or fewer sectors in any cylinder are bad, they are relocated to one of the five spare sectors in that cylinder. If more than five sectors in a cylinder are bad and the REMOVE option is set, the first five are relocated to the spare sectors, and the remainder are removed (by the equivalent of XD) from the Master Available Table on the disk. The REMOVE option is set by default for B 9499-6, B 9494-5, and B 9494-10 disks and for the RC (Reconfiguration), RI, and CI actions.

If more than five of the first 64 sectors in cylinder 0 are defective, that disk cannot be used with the B 1000 system software.

For a disk cartridge, if a sector is found to be bad, the entire track in which it resides is removed.

The program individually processes each cylinder of a disk to verify its integrity. The actual number of read and write operations performed on each cylinder is based on the characteristics of the disk device being processed. Cylinder 0, the outermost cylinder of the disk, is initialized first. The Master Available Table and the Relocate Table are constructed in cylinder 0 to store information on relocated and removed sectors. A printed analysis of these tables may be obtained by executing the normal-state utility program INITIALIZE/ANALYZER. Label information and the remainder of the MCP structures are built into cylinder 0 at the end of the initialization process.

Disk device characteristics and initialization characteristics are documented in appendix A of this volume.

Note

Beginning with the Mark 11.0 System Software release, a disk cartridge may not be initialized as a system disk.

PROGRAM OPERATION INSTRUCTIONS

The program may be executed from the card reader or from the ODT. It must be explicitly executed with the command EXECUTE SYSTEM/DISK-INIT. Once the program has begun processing a disk, that disk cannot be accessed by the system until the SYSTEM/DISK-INIT program has finished processing it. An OL inquiry reports the disk as IN USE BY SYSTEM/DISK-INIT while the program is running. After the disk is processed (initialized, verified, reconfigured, or sectors relocated), the system attempts to recognize it and bring it on-line; however, if the pack type is System, the disk cannot be recognized and remains off-line. Note that if the program is DSed, the disk will not be automatically readied; the operator must enter RY DPx before the MCP will attempt to bring the disk on-line.

Program Switch Settings

The program switches are set by the SWITCH program control instruction as a part of the EXECUTE string or, after BOJ, by the SW system command. (Refer to the *B 1000 Systems System Software Operation Guide, volume 1.*) The following table lists the switch settings for this program. (The default value is always 0.)

Switch	Name	Value	Meaning
1	Listing Switch	0, 2-15	Provide a printer listing of program operation.
		1	Do not provide a listing.
2	Auto EOJ Switch	0, 2-15	Program goes to EOJ after selected action is performed.
		1	Program asks for next unit mnemonic when the action is completed.

ODT Operation

When the program is executed from the ODT, it prompts the operator for all initialization specifications. Operator response must be in the following format:

<mix-number>AX <response>

The ODT prompt sequence begins with the following:

ENTER UNIT MNEMONIC <DC?, DP? OR BLANK FOR EOJ>

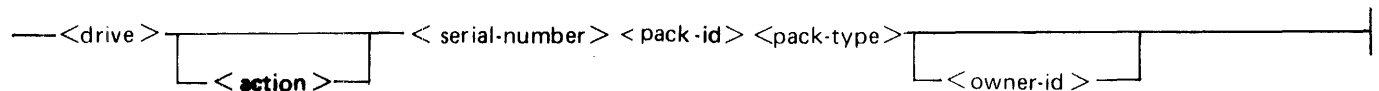
A blank or null input response to this prompt terminates the program. Any valid disk unit mnemonic is followed by the prompt sequence described in Initialization Specifications, later in this section.

Card Reader Operation

The control card format used to execute the program from the card reader follows:

```
? EXECUTE SYSTEM/DISK-INIT FILE SPEC NAME <spec-file-id>
? DATA <spec-file-id>
  <specification cards>
? END
```

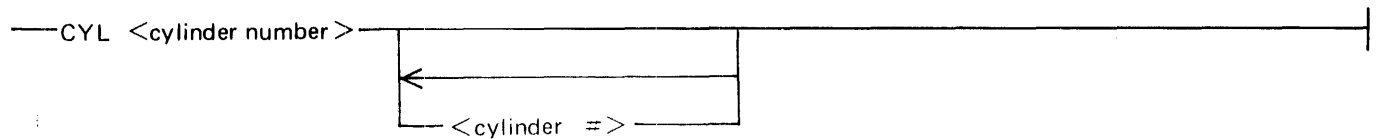
All specifications are entered in free-form format (up to 96 columns). The first six specifications must be on a single card, and must be separated by one or more spaces; comma (,) characters are not valid separators. The initialization options and the cylinder specifications must be entered on separate cards. The first specification card maintains the same order as the ODT prompt sequence and must be formatted as follows:



The semantics of the command are described in Initialization Specifications.

When the action CV (cylinder verify) or CI (cylinder initialize) is specified, the following card format must specify the cylinders to be processed.

Syntax:



If verification is requested (action V), then only the <drive> and <action> specifications are read; the remainder of the specification card is ignored.

The initialization specifications used when the program is executed from cards are identical to those used when it is executed from the ODT. (See Initialization Specifications.)

STATUS Command

While the program is executing, the operator may interrogate the program status with a <mix-number> AX ST command. In response, the program displays a message that indicates the cylinder that is the next to be processed and the number of sectors that have been relocated or removed thus far. An example of this message follows:

```
CYL 302 of 814 (37%): 8 BAD SECTORS (7 RELOCATED, 1 REMOVED)
```

INITIALIZATION SPECIFICATIONS

The following paragraphs are the initialization specifications.

User Interface Information

Drive

A disk unit mnemonic that names the drive on which the disk to be processed is located, for example, DPC.

Prompt: ENTER UNIT MNEMONIC <DC?, DP? OR BLANK FOR EOJ>

Action

Specifies the function to be performed on the drive. When an I, RI, RC, or V action is requested, the program processes the entire disk in sequential cylinder order, beginning with cylinder 0. When the requested action is CI or CV, the program issues prompts during processing that ask the operator to specify the cylinders to be processed. When an RR action is requested, the program prompts for each single sector to be relocated or reallocated.

Prompt: ENTER ACTION: <RR, I, V, RC, RI, CI, OR CV>

Descriptions of the possible actions follow.

I = Initialization

Initialize and verify the entire disk using the default initialization characteristics documented in appendix A of this volume.

V = Verification Only

Check that all sectors can be found and the data read with good parity. Only one pass is made on the disk and the reads are not stressed (no head offset or threshold changes).

RC = Reconfiguration

Purge the disk, verify it, and change the pack type to that specified. No cylinder initialization is performed unless a new bad sector is found on a cylinder. The affected cylinder will then be initialized, any sectors that had previously been relocated will again be relocated, and the new bad sector will then be relocated, or removed if there are no more spares available on the cylinder.

RI = Reinitialization

This is the same as an I action, with the addition that all sectors found to be bad on the previous initialization will automatically be considered bad on this initialization. This action is highly recommended for fixed disks that have been previously initialized at the factory.

CI = Cylinder Initialization

Initialize single cylinders or a range of cylinders as specified in answer to the following CYLINDER NUMBER prompts. Any sectors that had previously been relocated or removed will automatically again be relocated or removed. This action may also be used to restart a previously aborted initialization within 50 cylinders of where it was aborted.

If the disk has a bad label, Master Available Table or Working Available Table, or the operator specifies that the disk is to be purged, then cylinder 0 will automatically be initialized, thus purging the disk.

If all of the necessary structures on the disk are found to be valid and the previous initialization did not abort before it initialized the entire disk, the following prompt is displayed on the ODT:

IS A PURGE DESIRED? <YES OR NO>

A YES response causes the SYSTEM/DISK-INIT program to prompt for SERIAL NUMBER, PACK ID, PACK TYPE, OWNER'S NAME and OPTIONS. It will then automatically initialize cylinder 0, thus rendering inaccessible all data previously on the disk. Any further initialization will be performed in response to CYLINDER NUMBER prompts.

On a NO response, only those cylinders that were specified in response to CYLINDER NUMBER prompts are initialized. The disk is not purged. Cylinders or ranges of cylinders that are in use (not entirely available) are not initialized.

CV = Cylinder Verification

Verify only those cylinders specified in answer to the CYLINDER NUMBER prompts, except that cylinder 0 is always verified.

RR = Relocate and Reallocate

This action is only available for B 9494-5, B 9494-10, and B 9484-12 disks. It is used to relocate or reallocate (the reverse or undoing of a relocate) a single sector at a time. The sector may be in use and its data will be retained intact, whenever possible. When this action is requested, the following prompt sequence ensues:

A SECTOR CAN BE RELOCATED TO A SPARE OR
REALLOCATED FROM A SPARE

ENTER <SECTOR ADDRESS> TO RELOCATE, "-"
<SECTOR ADDRESS> TO REALLOCATE

WHICH SECTOR (BLANK TO QUIT)?

Response examples:

```
1000
@0003E8@
- 1000
- @0003E8@
```

The first two examples cause sector 1000 to be relocated; the next two cause it to be reallocated if it is found to be relocated. Entering a blank or null input will terminate the RR action and the program, unless program switch 2 = 1, which indicates that the program should not automatically terminate.

FC = Factory Convert

This action is available only for B 9494-5 and B 9494-10 fixed disks. The FC function converts the factory Interchange format to the B 1000 native-mode format and is used only once when the disk is first installed. The prompt sequence will ask for the PACK TYPE and OWNER'S NAME.

NOTE

The actions RR, RC, RI, CI, and CV are valid only when processing a disk that has previously been initialized with the Mark 8.0 (or later) version of the SYSTEM/DISK-INIT or PACK/INIT programs and the disk label, available tables, and Relocate Table are intact and valid.

Initialization Messages

SERIAL NUMBER

The SERIAL NUMBER message requests a six-digit decimal number to be used on a new or purged pack to identify the disk. This number must be non-zero and must be unique to avoid confusion. Note that the MCP does not allow two User disks with the same serial number to be on-line at the same time.

Prompt: ENTER 6 DIGIT SERIAL NUMBER

PACK ID

The PACK ID message requests a name that is assigned to the disk using 10 or fewer characters and containing no embedded blank characters. The system uses this name to reference the disk. When input specifications are entered from cards, omitting this entry causes an error for pack-type. If the disk is a User disk, the reserved name DISK will not be allowed.

Prompt: ENTER PACK ID

PACK TYPE

The PACK TYPE message requests the type of use that this disk will be dedicated to until the next initialization. The type is specified by a one-letter abbreviation of the allowable pack types, which are User (U), System (S), or Interchange (I). Only removable disk packs are allowed to be initialized as Interchange packs; therefore, the prompt message varies according to the type of disk being initialized.

Prompt: ENTER PACK TYPE - <U OR S>
(for fixed, non-removable disks)

Prompt: ENTER PACK TYPE - <U, I OR S>
(for removable disk packs)

OWNER'S NAME

The OWNER'S NAME message requests an optional 14-character field available to identify the person responsible for maintaining the disk. The system does not reference this field.

Prompt: ENTER OWNER'S NAME

OPTIONS

The OPTIONS message requests any of several options to be entered. These options are available during initialization of a disk to provide a more flexible initialization and verification procedure. The options allow the operator to extend or enhance the default initialization characteristics of the disk.

The OPTION prompt is repeated until a blank AC or AX command is transmitted, at which point the previously requested action begins.

The options are described in Initialization Options.

Prompt: ENTER OPTIONS

CYLINDER NUMBER

The CYLINDER NUMBER message asks for the number or the range of cylinders that are to be processed. This information is required only when either the CI or CV action is specified. The SYSTEM/DISK-INIT program sequentially processes the specified cylinders and reissues this prompt until it receives a blank or null input response. A single cylinder (indicated by one number) or a range of cylinders (indicated by number-space-number or by number-space-dash-space-number) may be specified. The following example illustrates this prompt and valid responses.

Prompt:	ENTER CYLINDER NUMBER
Responses:	25 40
	25 - 40
	300

When card input is used, the cylinder number specification must be on a separate card from the input specifications. The keyword, CYL, must appear in columns 1-3, followed by the cylinder numbers to be processed. Several CYL cards may be used.

Initialization Integrity Protection

When the action CI is requested, the SYSTEM/DISK-INIT program prevents a disk from being only partially initialized by checking the Relocate Table maintained on the disk for the initialization status. If the table indicates that the previous initialization did not finish, the program automatically completes the initialization before processing any operator-specified cylinders. The ENTER CYLINDER NUMBER prompt is not issued until the program ensures that the disk is completely initialized.

This feature of the CI action allows a restart of either disk initializer program (SYSTEM/DISK-INIT and PACK/INIT) when a new disk is being initialized. The restart of the program processes only those sectors that are not initialized.

Initialization Options

The operator can choose to extend or to enhance the default initialization characteristics for a disk by using the initialization options. The REMOVE and LIMIT options control the total number of bad sectors that will be handled for a single cylinder or for an entire disk before the disk becomes a candidate for rejection. The Marginal Sectors option specifies sectors that are known to be bad and requests that they be automatically relocated or removed. The Dollar Sign options are used to control the physical I/O operations that occur when processing a disk.

REMOVE Option

The REMOVE option allows the default errors-per-cylinder initialization error limit for disk packs (see appendix A) to be overridden. It specifies that after the first five bad sectors in the same cylinder have been relocated to a spare sector, the remaining bad sectors are to be removed (XD'ed) from the Master Available Table.

The default errors-per-pack limit might also cause a pack to be rejected unless the LIMIT option has also been specified.

The REMOVE option is set by default when processing B 9499-6, B 9494-5, and B 9494-10 disk drives and all disk cartridges, and when a RC, RI, or CI action is requested. When a sector is found to be bad on a disk cartridge the entire track on which the sector is located is removed from the Master Available Table.

The REMOVE option is set with the following syntax:

```
REMOVE
```

LIMIT Option

This option allows the default error limit totals to be overridden for disk packs and disk cartridges. If the LIMIT option is requested, a disk is not rejected until the total error count for the disk exceeds the value specified in the option. For B 9484-5 and B 9499-8 disk packs, the default error limit of five errors for each cylinder might cause a pack to be rejected, unless the REMOVE option has also been requested.

The LIMIT option is requested with the following syntax:

```
LIMIT <integer>
```

Marginal Sectors Option

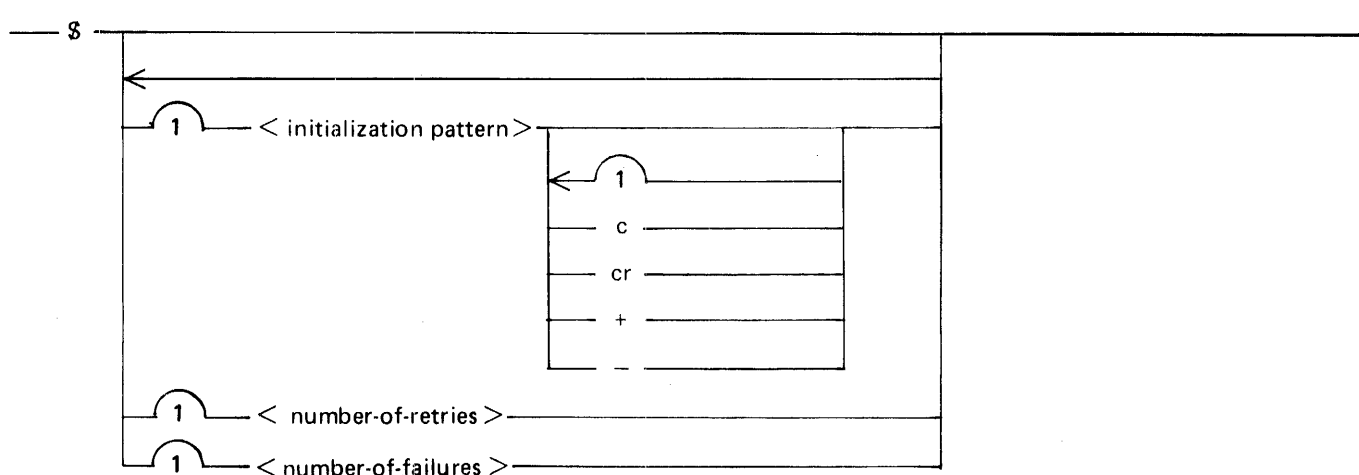
The Marginal Sectors option allows the operator to specify removal or relocation of sectors that might pass the disk integrity verification check but still be unreliable in normal use. Marginal sectors that are removed or relocated are included in the error count totals. Sector addresses specified in this option must be valid decimal addresses. More than one sector address can appear in each response to the ENTER OPTIONS prompt, or on each input specification card, with the restriction that the addresses are separated by spaces, not by commas.

If CI or CV is requested as the action, the operator must ensure that the specified marginal sector resides within a cylinder that is to be processed (has been included in a CYLINDER NUMBER input specification). Note that the first cylinder is cylinder 0, not cylinder 1.

Dollar Sign Option

The Dollar Sign option causes verification with or without offset, comparison during verification, modification on the number of retries on bad sectors, or modification on the number of failures that can occur before a sector is considered bad.

Dollar-sign option Syntax:



Semantics:

Initialization Pattern

The initialization pattern entry is a four-digit representation of a hexadecimal pattern within the range 0000 through FFFF. The first dollar-sign statement in a series of dollar-sign statements that omits this entry causes the default pattern to be used for that pass (refer to the disk device characteristics for the default patterns).

Offset: + or -

The plus-sign (+) or minus-sign (-) entry indicates that in (+) or out (-) head offset or, for the B 9499-6, high or low threshold is to be used for stressed reads during verification. Default is no offset or normal threshold.

The term offset refers to a disk drive capability which is a means of causing the disk unit to create a critical head alignment useful in testing marginal disk conditions. This capability is not available in Design Level I B 9499-7/B or B 9499-8 disk pack drives or the Design Level I DPEC. Specifying offset on disk packs of this level has no effect on initialization.

Compare: C or CR

The C or CR entry requests the options compare or compare-and-remove during verification of a disk cartridge. This option compares the data pattern that is written to the disk with the data pattern read during verification.

Number-of-retries

The number-of-retries entry specifies the number of times during verification that the same I/O operation is attempted after an exception condition has occurred. The default is 10 retries. This specification is not valid for B 9494-5, B 9494-10, and B 9499-6 type disks.

Number-of-failures

The number-of-failures entry specifies how many of the retries specified by the number-of-retries entry are allowed to fail before the sector is considered bad and is relocated or removed. Default is one failure. This specification is not valid for B 9494-5, B 9494-10, and B 9499-6 type disks.

NOTE

If the default values for both number-of-retries and number-of-failures are used while verifying a user disk, all 10 retries must be successful to allow the sector to remain.

PROGRAM OUTPUT

The following paragraphs describe the program output.

Informative Messages

The SYSTEM/DISK-INIT program issues ODT messages that describe the action that is being performed upon the selected disk and, where necessary, issues prompts that allow the operator to control the action. These messages are generally identical to those included in the summary listing and vary with the type of processing (action) requested.

Summary Listing

The SYSTEM/DISK-INIT program generates a backup printer listing that summarizes all activity that takes place during processing. The name of the printer backup file is displayed by the MCP in an ODT message issued when the SYSTEM/DISK-INIT program opens its LIST file. The messages in this listing are generally identical to those messages displayed on the ODT by the SYSTEM/DISK-INIT program during its execution.

B 1000 Systems System Software Operation Guide, Vol. 2
SYSTEM/DISK-INIT

The following example illustrates the information contained in this listing when cylinder processing is requested.

```
SYSTEM/DISK-INIT 12.0.000 (07/02/84 03:08 PM)
- RUN ON 9/7/84 AT 2:43 PM

ENTER UNIT MNEMONIC <DC?, DP? OR BLANK FOR EOJ>
---> ODT INPUT: "DPC"
DPC(206): ID = "DHH", SER. # 010371, TYPE = U
ENTER ACTION: <I, V, RC, RI, CI OR CV>
---> ODT INPUT: "I"
DPC(206): ID = "DHH", SER. # 010371, TYPE = S
INITIALIZE WILL CAUSE DPC TO BE PURGED
IS A PURGE DESIRED? <YES OR NO>
---> ODT INPUT: "YES"
ENTER 6 DIGIT SERIAL NUMBER
---> ODT INPUT: "010371"
ENTER PACK ID
---> ODT INPUT: "LADN"
ENTER PACK TYPE - <U, I OR S>
---> ODT INPUT: "S"
ENTER OWNER'S NAME
---> ODT INPUT: "SBPC - GOLETA"
ENTER OPTIONS
---> ODT INPUT: " "

INITIALIZE BEGINS      - CYL 0 - DPC
THE FOLLOWING SECTORS ARE IN ERROR AND WILL BE RELOCATED - DPC
SECTOR @005A24@ = 23076
SECTOR @045152@ = 282962
SECTOR @049112@ = 299282

DPC(206): ID = "LADN", SER. # 010371, TYPE = S
3 BAD SECTORS RELOCATED
INITIALIZE COMPLETED - DPC
```

B 1000 Systems System Software Operation Guide, Vol. 2
SYSTEM/DISK-INIT

For processing that is not cylinder oriented, the following example illustrates the summary listing produced by the SYSTEM/DISK-INIT program.

```
SYSTEM/DISK-INIT 12.0.000 (07/02/84 03:08 PM)
- RUN ON 9/10/84 AT 2:48 PM

ENTER UNIT MNEMONIC <DC?, DP? OR BLANK FOR EOJ>
----> ODT INPUT: "DPE"
DPE (677): ID = "LAH", SER. # 031268, TYPE = U
ENTER ACTION: <RR, I, V, RC, RI, CI OR CV>
----> ODT INPUT: "CI"
DPE (677): ID = "LAH", SER. # 031268, TYPE = U
IS A PURGE DESIRED? <YES OR NO>
----> ODT INPUT: "YES"
ENTER 6 DIGIT SERIAL NUMBER
----> ODT INPUT: "031268"
ENTER PACK ID
----> ODT INPUT: "LADNH"
ENTER PACK TYPE - <U, I OR S>
----> ODT INPUT: "U"
ENTER OWNER'S NAME
----> ODT INPUT: "SBPC"
ENTER OPTIONS
----> ODT INPUT: " "

INITIALIZE BEGINS      - CYL 0 - DPE
INITIALIZE COMPLETED - CYL 0 - DPE

ENTER CYLINDER NUMBER OR RANGE
----> ODT INPUT: "15 18"

INITIALIZE BEGINS      - CYL 15 - DPE
INITIALIZE COMPLETED - CYL 18 - DPE

ENTER CYLINDER NUMBER OR RANGE
----> ODT INPUT: " "

DPE (677): ID = "LADNH", SER. # 031268, TYPE = U
0 BAD SECTORS
INITIALIZE COMPLETED - DPE
```

Error Messages

The following are some of the error messages that will be displayed when the associated error conditions are encountered.

INVALID RESPONSE: BLANK OR NULL INPUT

A blank or null ODT prompt response is not valid at this time.

INVALID RESPONSE <response>

The ODT prompt response was invalid; try again with the proper response.

INVALID SERIAL NUMBER

Re-enter the serial number. It must be a six-digit, non-zero decimal number.

INVALID SECTOR NUMBER

The addresses of marginal sectors must be specified in decimal.

THE FOLLOWING SECTORS ARE IN ERROR AND WILL BE RELOCATED

This informative message heading is issued when the first bad sector on a disk is encountered during an initialization operation. The addresses of the bad sectors follow the heading.

THE FOLLOWING SECTORS ARE IN ERROR AND THEIR TRACKS WILL BE REMOVED

This is an informative message heading issued when the first bad sector is encountered while initializing a disk cartridge. The addresses of the bad sectors follow.

DPB(206): ID = "NEH", SER. # 060276, TYPE = U

INITIALIZE WILL CAUSE DPB TO BE PURGED

IS A PURGE DESIRED? <YES OR NO>

This is a safety measure to ensure that the operator has specified the correct disk. A YES response allows the action to proceed; a NO response aborts the action and restarts the prompt sequence from the beginning.

WOULD YOU LIKE TO RETRY? <YES OR NO>

This message follows various other error messages. It offers the operator the option of retrying an I/O operation that has failed. Depending on the reason for the failure, it may be necessary for the operator to take some action to correct the failure, such as removing a WRITE LOCKOUT condition from the disk drive, before allowing the program to continue. A YES response causes the I/O operation to be retried; a NO response results in termination of the program.

DISK ERROR(10 TRIES) <disk-address> -

RESULT = < disk-io-result-descriptor >

An irrecoverable I/O error still exists after 10 retries. The operator is offered the option of retrying the I/O operation another 10 times.

THIS PROGRAM MUST BE RESTARTED TO CONTINUE

The program cannot continue. Re-execute the program and try again.

WRITE LOCK OUT <unit-mnemonic>

Reset the WRITE LOCKOUT status (turn on WRITE ENABLE) on the specified disk unit and respond to any following ODT prompts.

DISK NOT READY <unit-mnemonic>

Ready the specified disk unit and respond to any following ODT prompts.

THE PACK ON <unit-mnemonic> CANNOT BE USED WITH THE MCP

The initialization of the current disk has been aborted before it finished initializing the entire disk. The disk is not usable by the MCP because it does not have a label or directory.

SECTOR <disk-address> REMOVAL REQUIRED – CANNOT REMOVE

ANY OF THE 1ST 64 SECTORS>

All of the first 64 sectors of a disk must be available or the disk cannot be used on a B 1000 system. The disk has been rejected, and the program will terminate following this message.

REMOVED <disk-address>

This is an informative message that specifies that the sector has been removed (XDed) from the Master Available Table because no more spares were available on its associated cylinder.

COMPARISON ERROR <disk-address>

The C or CR Dollar Sign option is enabled, and a comparison error has been detected on a disk cartridge.

ERROR LIMITS HAVE BEEN EXCEEDED – <unit-mnemonic>

The maximum number of bad sectors allowed for the currently attached disk (either the default value or the value specified by the LIMIT option) has been exceeded. The disk has therefore been rejected and the program will terminate.

INVALID CYLINDER NUMBER <cylinder-requested>

The cylinder number requested is beyond the capacity of the attached disk. Try again with a valid cylinder number.

ONLY "STATUS" OR "ST" CAN BE ACCEPTED AT THIS TIME

Once the program has begun to perform an action on a disk, it will not accept any ODT input, other than a request for the status of the program, prior to the completion of the requested action.

MORE THAN <remove-limit> SECTORS HAVE BEEN REMOVED ON <unit-mnemonic>

MORE THAN <cylinder-transient-limit> TRANSIENT ERRORS HAVE BEEN ENCOUNTERED ON CYLINDER <cylinder-number>

MORE THAN <disk-transient-limit> TRANSIENT ERRORS HAVE BEEN ENCOUNTERED ON <unit-mnemonic>

Any of the preceding messages may be accompanied by the following message:

**WARNING: <unit-mnemonic> DOES NOT MEET BURROUGHS
STANDARDS FOR AN ACCEPTABLE PACK
DO YOU WANT TO IGNORE THIS WARNING AND CONTINUE?
<YES OR NO>**

One of the preceding error limits has been exceeded. This would normally result in disk rejection; however, the operator is usually given a chance to override these limits. A YES response causes the limit to be ignored; a NO response result in disk rejection and program termination.

Use caution when ignoring limits, particularly in the case of transient errors, because of possibilities like the following: (1) A hardware problem may exist; that is, the failure may be associated with the disk control or the drive rather than the disk medium. (2) The disk may exhibit a high error and retry rate when used with the MCP. (3) A failing sector may not have failed enough times to be considered as a candidate for relocation, or a sector failing when read as one of a group of sectors may defy isolation to a single failing sector.

WARNING: UNABLE TO ISOLATE INTERMITTENTLY FAILING SECTOR ON CYL <cylinder-number>

- SECTORS <disk-address-range>, RESULT = <disk-io-result>

An I/O exception was encountered while verifying a group of sectors, but no single failing sector could be found that might be causing the problem. This is a warning. The sector does not fail when read as a single sector, but it might cause the MCP to do single-sector retry on it. Examine the ELOG file (ET system command). If a sector in this range shows a failure, relocate it. Use RR for B 9494-5, B 9494-10, and B 9484-12 disks, and CI and the Marginal Sectors option for all other disks.

SECTION 34

SYSTEM/ELOGOUT

The SYSTEM/ELOGOUT program is a system utility program designed to analyze the Field Engineering Maintenance Log (SYSTEM/ELOG file) and produce highly organized, meaningful reports of the results. Errors are reported in chronological order by unit-mnemonic, and appropriate totals are summarized at the end of the output listing.

OPERATING INSTRUCTIONS

The ET system command causes the SYSTEM/ELOG file to be transferred, creating a file labeled ELOG/#<integer> and a new (empty) SYSTEM/ELOG file. The SYSTEM/ELOGOUT program is executed automatically by the MCPPII following the transfer and is file-equated to the transferred ELOG file.

If an analysis of a previously-transferred ELOG file is desired, the SYSTEM/ELOGOUT program must be executed as follows:

```
EXECUTE SYSTEM/ELOGOUT FILE ELOGFILE NAME = ELOG/# <integer>;
```

Program Switches

Program switches 0, 1, and 2 control certain functions of the SYSTEM/ELOGOUT program:

Switch 0

If this switch is set to any non-zero value, the input ELOG file is closed with PURGE after the input phase has been completed. If the switch is at zero (the default value), the input ELOG file is not removed from disk.

Switch 1

When this switch is set to any non-zero value, the entire output listing is printed single-spaced and all page skipping and multiple spacing operations are suppressed. When the switch is at zero (the default value) the default printer spacing is used.

Switch 2

When this switch is set to any non-zero value, the system serial number is printed at the beginning of each individual report. No action is taken if no system serial number has been specified.

The program switch feature is also used to specify the system serial number, as follows:

```
MODIFY SYSTEM/ELOGOUT SWITCH = <system number>
```

If the system serial number as well as other program-switch-controlled functions are to be specified, the serial number must be specified first in the control string, as follows:

```
MODIFY SYSTEM/ELOGOUT SWITCH = 513; SWITCH 0 = 1; SWITCH 1 = 1
```

If the system serial number has been specified, it is printed on the first page of the output listing.

Error Message

If no records are found in the ELOG file when the SYSTEM/ELOGOUT program is run, the program goes to end-of-job after displaying the following message:

<ELOG-file-id> EMPTY

OUTPUT REPORTS

A number of reports are produced by the SYSTEM/ELOGOUT program. These reports are described in order of appearance in the paragraphs that follow.

System Information Report

The first portion of the line printer output listing consists of information describing the system and the ELOG file being analyzed.

The first item is a block containing the system serial number (if one was specified in the program switches) plus the name of the ELOG file and the time period of interest.

The second item, under the heading CLEAR/STARTS gives the dates and times of all CLEAR/START operations in the order they occurred (if any occurred) during the time period of interest.

The third item, SYSTEM CONFIGURATION, is a description of the system I/O configuration. This is included at least once if a CLEAR/START operation took place during the time period. Items included are processor identification and memory size as well as each hardware device name with its device-id (control-id in hexadecimal), port, channel, number of units, and, if appropriate, the exchange port and channel. For B 1990 systems, a DRIVE TRANSFORMATION TABLE describing the order of the disk drives is also produced.

A description of the system software in use follows. This is headed SOFTWARE CONFIGURATION and, like the system configuration, appears at least once if a CLEAR/START operation took place. This description includes the current names for the MCP, the MCP Interpreter, GISMO, the Initializer, and the MICRO.MCP.

Additional copies of the system configuration and software configuration tables are produced for each CLEAR/START operation if either configuration changed from that in effect at the time of the preceding CLEAR/START operation.

NOTE

The MCP places the I/O and software configuration information into the ELOG file only during CLEAR/START operations. Therefore, these reports are not produced if no CLEAR/START operation has taken place during the time period of the ELOG file being analyzed.

Operator Message Report

Messages that have been entered in the ELOG file by the system operator are listed chronologically, together with the date and time of entry. (Refer to the EM system command in section 5 of Volume 1 of the *B 1000 Systems System Software Operation Guide*.) If no messages were entered, there is no report.

Memory Error Report

Memory parity errors and related information are reported. In addition, since all B 1860, B 1870, and B 1900 series systems include error-correcting memory, certain types of memory errors can be dynamically corrected. Thus, system or program failures are avoided and replacement of any possibly defective memory chips can be scheduled for a convenient time.

The memory error report for these systems includes the date and time of the error, a hexadecimal representation of the processor's ELOG register (which reports the memory error to the software), and a decoded representation of this register. This decoded information enables the Burroughs Field Engineer to quickly locate and replace questionable chips.

For uncorrectable memory parity errors, the job number of the program that was terminated and its base and limit addresses are also included in the report.

Cassette Error Report

This report is produced if any data errors are detected during programmatic loading of a cassette through the cassette reader. The SYSTEM/LOAD-CAS program is an example of a programmatic cassette loading sequence.

The report includes the date and time of the error and the job number and base and limit addresses the program reading the cassette.

Disk Cache Error Report

This report is produced when any one of several types of errors occurs in the Disk Cache subsystem.

The report includes the date and time of the error and event counts as follows:

Cache timeout. An expected cache response or cache state did not occur in an allotted time period, so the attempted cache operation was aborted.

Cache ECC error. A non-zero syndrome was produced during the transfer of a word read from the cache memory to the DSC RAM, indicating a cache memory error. The read operation was completed from disk.

Cache echo error. The command echo from cache does not match what is expected, indicating a failure in hardware or possibly an interaction problem.

Cache copy-read error. The cache did not go idle after copying information during a read from disk.

Cache copy-write error. The cache did not go idle after copying information during a write to disk.

Cache post-Terminate copy-read error. An error was detected during the post-Terminate phase of a Read operation.

Result Code Definitions

Result codes, used in the Unit Error Reports, are defined in a block immediately preceding the first Unit Error Report.

Unit Error Reports

Peripheral unit errors are reported and summarized by unit-mnemonic and, where multiple controls are connected to the same unit (for example, a PE/NRZ tape exchange), by hardware address.

The Unit Error Reports, one for each peripheral with one or more errors, present a number of items of information. These items are described below.

The heading of each Unit Error Report includes the unit mnemonic, the device type and, as appropriate, port, channel, and unit numbers. The date is also provided.

For each Unit Error Report, there also may be a Result Descriptor Analysis report and, for disk units, a Disk Error Address Analysis report. These are described in the paragraphs that follow. Following all the Unit Error Reports is a Unit Error Summary; this is described in the last paragraph of this section.

TIME

The time of each error (chronological order).

RESULT

The result descriptor in hexadecimal.

OP

The I/O operation code and, to its left, R for read, W for write, or S for space.

DATA RQST

The data length (in bytes) requested in the I/O descriptor.

DATA XFER

The actual length (in bytes) of the data transferred.

RETRY COUNT

The number of retries attempted. This is often shown as zero for MCPII operations, even though retries were actually attempted.

RESULT CODE

A code indicating the success or failure of the recovery operation. (These codes are defined in a block preceding the first Unit Error Report.)

JOB

The job number of the program encountering the error. If the job number is prefixed with an asterisk, the ELOG entry was made by MCPII at the request of the specified program.

SERIAL NUMBER

For magnetic tape, disk packs, and disk cartridges.

LABEL

The label of the file. (Always the pack-id for disk devices.)

ADDRESS

The disk address. (For disk devices only.)

EXTENDED RESULT

The extended result descriptor (for disk packs only). If an extended result descriptor is present, it is shown in hexadecimal on the next line of the Unit Error Report. On the immediately following lines, the extended result descriptor is decoded to provide salient information (CYL, HEAD, SEC, OP, VXYZ, UNIT, error types).

Cylinder addresses in the extended result descriptor can appear to be incorrect when TEST operations are being performed to a disk drive. This is because a TEST operation has no associated cylinder address and the old address does not change. If an exception condition occurs during a TEST operation, the exception condition is logged into the SYSTEM/ELOG file when the next READ or WRITE operation is initiated. To correctly identify which operation was contained in the extended result descriptor, examine the OP column of the line that gives the extended result descriptor.

REEL

For tape units only, this column gives the reel number on which the error occurred.

BLOCK

For tape units only, this column gives the block in which the error occurred.

Result Descriptor Analysis

Following the Unit Error Report is the Result Descriptor Analysis. This shows the descriptors in both hexadecimal and binary format and provides a brief description of the error. Each result descriptor is listed once in this report, regardless of the number of times it appears in the Unit Error Report.

Disk Error Address Analysis

For disk devices, this follows the Result Descriptor Analysis. Here, the errors are grouped by disk address, sorted in ascending order, and summarized by address and actual data transfer length (in sectors). Each line represents the errors for a particular disk address and actual length pair, giving the total number of retries (retry counts shown as zero in the chronological listing are counted as one retry), and decoding the beginning and ending disk addresses involved in the operation into their actual hardware locations. This decoding can enable the Burroughs field engineer to locate not only the source of the failure, but also to make reasonable inferences as to errors that can be caused by a single read/write head, even though the addresses can be widely scattered across an entire disk surface.

Power On/Off Error Rate Report

This report gives snapshots of the error count information logged when a disk is powered on or off. Historic and recent information by unit and pack is given. The total number of input/output operations as counted by GISMO and the number of IO errors as recorded by the MCP are reported. The IO error count is incremented by one for each user IO that has a problem, no matter how many retries are done.

See the ER command in the *B 1000 Systems System Software Operation Guide, Volume 1*, for information on error rates and on resetting the unit and pack error rate information.

Following are the meanings of the various fields for the Power On/Off Error Rate Report.

PWR

Indicates whether a power ON or OFF caused the line to be generated.

SERIAL NUMBER

Serial number of pack at time of power ON or OFF.

UNIT HISTORIC DATE

Date the unit was last reset or coldstarted.

UNIT HISTORIC IO COUNT

Number of unit I/Os as counted by GISMO since the last unit reset or coldstart.

UNIT HISTORIC ERROR COUNT

Number of unit I/O errors as recorded by the MCP since the last unit reset or coldstart.

UNIT RECENT DATE

Date the unit was last reset or clear/started.

UNIT RECENT IO COUNT

Number of unit I/Os as counted by GISMO since the last unit reset or clear/start.

UNIT RECENT ERROR COUNT

Number of unit I/O errors as recorded by the MCP since the last unit reset or clear/start.

PACK HISTORIC DATE

Date the pack was last reset or initialized.

PACK HISTORIC IO COUNT

Number of pack I/Os as counted by GISMO since the last pack reset or pack initialize.

PACK HISTORIC ERROR COUNT

Number of pack I/O errors as recorded by the MCP since the last pack reset or pack initialize.

PACK RECENT DATE

Date the pack was last reset or powered ON.

PACK RECENT IO COUNT

Number of pack I/Os as counted by GISMO since the last pack reset or pack power ON.

PACK RECENT ERROR COUNT

Number of pack I/O errors as recorded by MCP since the last pack reset or pack power ON.

Configuration Disk Unit Error Rate Report

This report gives snapshots of the historic error information by unit based upon information obtained at clear/start or when the log is transferred. The I/O operations are counted by GISMO and the I/O error count is recorded by the MCP. The I/O error count is incremented by one for each user I/O that has a problem, no matter how many retries are done.

See the ER command in the *B 1000 Systems System Software Operation Guide, Volume 1*, for information on error rates and on resetting the unit and pack error rate information.

The meaning of the various fields for the CONFIGURATION DISK UNIT ERROR RATE REPORT follows:

PORT

Hardware port number.

CHANNEL

Hardware channel number.

UNIT

Hardware unit number.

DATE

Date of clear/start or log transfer.

TIME

Approximate time of clear/start or log transfer.

DATE

Date of last reset or coldstart.

IO COUNT

Number of unit IOs as counted by GISMO since the last unit reset or coldstart.

ERROR COUNT

Number of unit IO errors as recorded by the MCP since the last reset or coldstart.

Unit Error Summary

This report is a summary of the peripheral unit errors listed in the Unit Error Reports. Here, totals are given for each unit-mnemonic and, if applicable, for each hardware address. For each device listed, counts of input errors (read and space operations), output errors (write operations), and memory access errors are presented in tabular form.

The Unit Error Summary also includes cassette error and main memory error counts.

SECTION 35

SYSTEM/FILE-INIT

The SYSTEM/FILE-INIT program is a utility program that can allocate and initialize relative files for COBOL74 programs. A description of the format of the B 1000 implementation of relative files as defined for the COBOL74 language, definitions of the initialization process, and details on the use of this program are included in this section.

RELATIVE FILES

A relative file may be visualized as a serial string of spaces of fixed length, each of which is capable of holding one logical record. Each of these spaces is addressed by a relative record number. Thus, relative record number 10 addresses the record in the tenth space even if there are no records in the first nine spaces. The length of this space and, therefore, the maximum record size is specified by the file attribute MAXRECSIZE.

The space is considered empty if it contains no valid record. It can be made empty by making its contents inaccessible through the DELETE mechanism in the operating system.

A block comprises one or more record spaces. The number (BLOCKSIZE) is set when the file is created and cannot be changed. Appended to the beginning of each block is a Block Control Information (BCI) field. The BCI field includes one bit for each record space in the block plus enough filler bits to reach the next higher modulo 8 value. The BCI field keeps track of valid records; each assigned bit is set (1) if the record is valid and reset (0) if not. As a consequence of this design, the BLOCKSIZE (in bits) in a relative file is equal to the number of record spaces times the maximum size of a record (MAXRECSIZE) plus the BCI. The SYSTEM/FILE-INIT program allocates the disk area and resets all the presence bits.

INITIALIZATION

Initialization of files is, ultimately, a responsibility of the MCP. The mode of access to a file determines the method of initialization. Execution of the SYSTEM/FILE-INIT program, either by the MCP or by the operator, is one method.

If the file access mode is sequential, the MCP II allocates the disk area and the logical I/O routines initialize each block before it is accessed.

If a sequential-access file is undergoing a CLOSE operation and the end-of-file (EOF) pointer is not at the end of a disk area, the MCP II calls the SYSTEM/FILE-INIT program to initialize the remainder of that disk area before the CLOSE operation is completed.

For random or dynamic access modes, if a new disk area is to be allocated to the file, the MCP II calls the SYSTEM/FILE-INIT program to ensure that the BCI fields are initialized.

In all cases where the MCP calls the SYSTEM/FILE-INIT program, execution of the program requesting the disk area is suspended and the SYSTEM/FILE-INIT program is called into execution at the same priority level as the requesting program. The requesting program is reinstated when initialization is complete, even if an irrecoverable error is detected during the initialization process. The procedure is similar to that used for a program sort. Any USE procedures declared for the file are executed at this time.

Only when a file is to be initialized prior to an actual production run is it necessary to provide information to the SYSTEM/FILE-INIT program. Initializing a file before it is actually required ensures that the required disk space is available and minimizes the amount of processing required to add a record to a new disk area. The latter consideration can be important in an on-line environment, where response time can be critical.

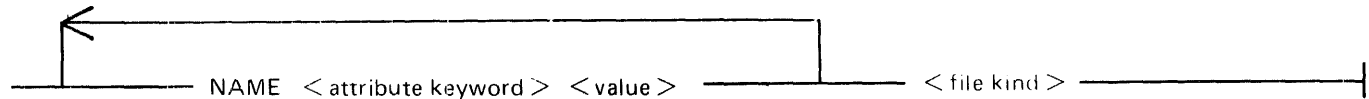
PROGRAM EXECUTION

The system operator is responsible for providing file attributes for each file to be initialized. Attributes are entered from one of three sources: a disk file named RELFILE equated to another file name, a file of card images named CARD, or the ODT.

SW0 = 0 The source is a file equated to RELFILE.
 SW0 = 1 The source is a card file named CARD.
 SW0 = 2=15 The source is ODT input.

The attributes are shown in table 35-1. The keywords listed in the table are used when input is from cards or the ODT. Entry format is shown in the following syntax diagram:

Syntax:



The keyword-value pairs must not be split across record boundaries. The <file kind> is always RELATIVE. (The SYSTEM/FILE-INIT program only handles relative files.)

Table 35-1. File Attributes

Attribute	Keyword	Default
File name	NAME	none
Record size	RECORD.SIZE RECSIZE RSZ	175
Records per block	RECORDS.BLOCK R.B	1
Blocks per area	BLOCKS.AREA B.A	100
Number of areas	AREAS ARE	25
File kind	FILEKIND FILE.TYPE	none

EXAMPLES

The following is an example of a card file of instructions that can be used to invoke the SYSTEM/FILE-INIT program:

```
?EXECUTE SYSTEM/FILE-INIT;SWITCH 0=1;
?DATA CARD
NAME=MY/FILE2 RECSIZE=80 RECORDS.BLOCK=2
NAME=MY/FILE3 NAME=MY/FILE4 AREAS=105
NAME=PACKID/MY/FILE5
RECSIZE=80
RECORDS.BLOCK=2
BLOCKS.AREA=4
AREAS=105
FILEKIND=RELATIVE
?END
```

If only one file is to be initialized, a file-equate is probably the easiest way to execute the program. An example of a card file input follows. (The defaults shown in table 35-1 apply to any attributes not specified.)

Example:

```
? EXECUTE SYSTEM/FILE=INIT; SWITCH 0=0;
? FILE RELFILE NAME TEST/F1 RECORD.SIZE 80 RECORDS.BLOCK 19
? BLOCKS.AREA 10 FILE.TYPE=RELATIVE; % AREAS DEFAULT TO 25
```

ERROR MESSAGES

When the program is executed automatically by the MCPII, error messages are not displayed; however, three conditions can be returned to the calling program.

1. The file is not present on disk.
2. The file is locked (by some other program).
3. An irrecoverable I/O error occurred on initialization.

These conditions are stored in the file status variable for the appropriate file, and the calling program must take action based on them. The first two conditions are unlikely to occur because the calling program is either creating the file (in the case of a new file), or accessing an existing file. The third condition is more likely to occur; therefore, it is advisable to ensure that the calling program either handles the condition or is discontinued.

When the program is executed from the ODT or from the card reader, the following error messages may be displayed:

INVALID <token name> -- <value>

The <token name> entered is a valid attribute name but the value associated with it exceeds the MCPII limits for that attribute. Possible values for the particular attributes are given in section 4 of Volume 1 of the *B 1000 Systems System Software Operation Guide*.

INVALID TOKEN -- <name>

The attribute specified by <name> is not recognizable by the program.

EQUAL SIGN EXPECTED

The equal sign (=) character between the attribute and its value is mandatory.

VALUE EXPECTED AFTER =

No value was found in the input string after the equal sign (=) character in an attribute specification. When the input is from cards, the attribute value must be contained on the same card as the attribute name.

UNEXPECTED TOKEN -- <name> -- NAME EXPECTED

The program was looking for a NAME attribute and encountered a different attribute name.

<file-name> NOT INITIALIZED FILE MISSING

<file-name> NOT INITIALIZED FILE LOCKED

<file-name> NOT INITIALIZED EXCEPTION ON WRITE

The file was not initialized because of the reason specified. The first two messages cannot occur if the program is executed from the ODT or the card reader. In those cases, the program always creates a new file. Therefore, the program cannot be used to initialize one area of an existing file.

SECTION 36

SYSTEM/ICMD-INIT

The SYSTEM/ICMD-INIT program is a normal-state utility program that is designed to initialize, verify, and label the industry-compatible mini-disk (diskette or floppy disk). The program assigns addresses to the appropriate sectors, writes a fixed data pattern in the sector data area, and reads this pattern back to ensure that the sectors are not defective.

If a sector cannot be read after 10 retries or cannot be written after two retries (whether the error is a sector address error or a data error), the sector is considered bad and the entire track is relocated to a spare track. A diskette is rejected as unusable if there are more than two defective tracks.

OPERATING INSTRUCTIONS

The SYSTEM/ICMD-INIT program operates under control of the MCPPII and receives input through AC or AX system commands entered through the ODT in response to ACCEPT messages. A listing of the test data pattern used, relocated tracks, and any errors that occurred during program execution is produced.

Executing the program with switch 9 = 1 allows an operator-specified test pattern to be entered. If switch 9 = 0, the default pattern of @6363@ is used.

The following messages, displayed during execution, require operator response:

ENTER UNIT NAME, FD<?>

Respond with the unit-mnemonic of the diskette to be initialized.

DISKETTE ON UNIT <unit-mnemonic> IS ALREADY INITIALIZED, ENTER Y FOR RE-INIT (BLANK FOR EOJ)

This message is displayed only if the specified diskette is already initialized. A Y response invokes the requested initialization; otherwise, the program terminates.

WARNING

Entering Y erases all data on the diskette.

FOR INTERLACED ADDRESSES ENTER Y (BLANK FOR SEQUENTIAL)

A Y response interlaces the sector addresses in the following order: 1, 14, 2, 15, 3, 16,...,12, 25, 13, 26.

ENTER PATTERN – 4 HEX NUMBERS (BLANK FOR DEFAULT OF @6363@)

This message is displayed only if the SYSTEM/ICMD-INIT program is executed with switch 9=1. Four hexadecimal digits are required for the response.

DISKETTE GOOD

This message is displayed twice if the initialization is successful and the diskette is good. The standard pattern of @E5E5@ is written on the diskette when the second message is displayed.

FOR RETRY PATTERN ENTER Y (BLANK FOR EOJ)

This message is displayed only if the SYSTEM/ICMD-INIT program is executed with switch 9=1. A Y response causes the program to request a new test pattern. A blank causes the ending pattern of @E5E5@ to be written.

ENTER 1 TO 6 ALPHANUMERIC CHARACTERS FOR VOLUME ID OR LEAVE BLANK

Respond with the new diskette volume-id (a blank results in a blank volume-id).

ERROR MESSAGES

INVALID UNIT NAME, RE-ENTER

The <unit-mnemonic> was not in the form FDx.

****ERROR**PORT AND CHANNEL NOT ASSIGNED**

Unit cannot be accessed. The program is terminated. Possible problems: missing unit, defective unit.

ILLEGAL PATTERN – RE-ENTER 4 HEX NUMBERS (BLANK FOR DEFAULT OF @6363@)

Each character must be a valid hexadecimal digit.

PATTERN TOO LONG, RE-ENTER

Pattern must be four hexadecimal digits in length.

****I/O OPERATION NOT COMPLETE**

The diskette control has control errors.

<unit-mnemonic> NOT READY, ENTER Y FOR RETRY (BLANK FOR EOJ)

The diskette is not ready.

****ERROR**MORE THAN 2 BAD TRACKS**

Diskette is bad and unusable. The program is terminated.

SECTION 37

SYSTEM/IS-MAINT

The SYSTEM/IS-MAINT program is a normal-state utility program that performs library maintenance functions for index sequential files. An index sequential consists of one cluster file, one data file, and any number of index files.

The cluster file has the same name as the index sequential file and contains a dictionary of the names of all associated subfiles (data files and index files).

OPERATING INSTRUCTIONS

The SYSTEM/IS-MAINT program may be executed from the ODT, a card reader, or a pseudo-deck. Program commands may be entered from the ODT or from a data file.

From the ODT, execution is invoked by entering:

```
EXECUTE SYSTEM/IS-MAINT
```

At BOJ, the program displays MCP ACCEPT requests for program input. Input is from a data file must be file-equated at this time:

Examples:

```
EXECUTE SYSTEM/IS-MAINT;FILE SOURCE NAME CARD-FILE CARDS
```

```
EXECUTE SYSTEM/IS-MAINT;FILE SOURCE NAME IS/INPUT DISK
```

Input From a Data File

Input from a data file requires the internal file SOURCE to be file-equated to the desired input file at execution time. The file is opened using input blocking. Each record in the data file corresponds to a single AC or AX command from the ODT. Command prompts that normally appear are not displayed.

Input from the ODT

The name of the cluster file upon which maintenance is to be performed is entered in response to the ACCEPT message ENTER FILE NAME.

If the cluster file is missing or invalid, the program notifies system operator and repeats the ACCEPT message. When a valid cluster file is found, the program displays the ACCEPT message ENTER COMMAND and waits for the AC or AX response. Commands entered from this point on are performed on the cluster file.

END, EOJ, or <mix-number>AX<null>

Entry of the commands END or EOJ or an AC or AX null command terminates the program.

Syntax:

—END—

—EOJ—

—< job-number> AX <null >—

HELP

In response to the HELP command the program displays its operating instructions on the ODT. This command functions in the same manner as the TEACH command.

LIST

The LIST command instructs the program to produce a listing of the name and version dates of the cluster file, the data file, and all the index files on the ODT or the line printer, depending on which is specified. The default destination is the line printer.

NEXT

The NEXT command provides the capability to reference another cluster file. In response, the program again displays the ENTER FILE NAME prompt and the ACCEPT mechanism waits for an AC or AX response. with the new cluster file name. Command entry then proceeds.

REMOVE

The REMOVE command removes the cluster file and all of its subfiles, the names of which are contained in the cluster file. For a file to be removed, it must have the correct version date and no users. If a subfile is missing, the program displays the following message.

NOT ALL STRUCTURES ARE PRESENT
DO YOU WANT THE REMOVE TO CONTINUE? – YES OR NO

If the operator enters a NO response, the program asks for another command. If the response is YES, the program removes the files that are present. After completing the remove operation, the program repeats the process of displaying an ENTER FILE NAME prompt and waiting for the AC or AX command input.

TEACH

In response to the TEACH command, the program displays its operating instructions on the ODT. This command functions in the same manner as the HELP command.

for RPG/ISAM

(c) GLOBAL

UPDATE

The UPDATE command modifies data in the cluster file dictionary to reflect the new names of data or index files. The names can take one of these forms:

<pack-id>/<multifile-id>/<file-id>
<pack-id>/<multifile-id>/=
<pack-id>/=/=

If the first name used cannot be found in the cluster file, or if the second name is already present in the cluster file, an error message is displayed. The program makes no effort to determine if these files are actually on disk.

Syntax:

UPDATE <file-name> TO <file-name>

VERIFY

The VERIFY command determines if all the subfiles are present in the disk directory and if their version dates are correct. A hardcopy summary of the verification results is listed on the line printer and the results are also displayed on the ODT.

USER " " " "
BARAZ2 / ARM / ØØ CLOSED
Ø1 ...

SAMPLE EXECUTION

The following example shows program execution from the ODT.

Example:

```
EXECUTE SYSTEM/IS-MAINT;
SYSTEM/IS-MAINT =38 BOJ. PP=4, MP=4 TIME = 10:30:37.7
% SYSTEM/IS-MAINT =38 ENTER FILE NAME
SYSTEM/IS-MAINT =38 ACCEPT
% SYSTEM/IS-MAINT =38 ENTER FILE NAME
38 AX CL/FILE
%SYSTEM/IS-MAINT =38 ENTER COMMAND
SYSTEM/IS-MAINT =38 ACCEPT
38 AX LIST ODT
% SYSTEM/IS-MAINT =38 FILE NAME : CL/FILE
% SYSTEM/IS-MAINT =38 FILE TYPE : INDEX SEQUENTIAL
% SYSTEM/IS-MAINT =38 # SUBFILES : 2
% SYSTEM/IS-MAINT =38
% SYSTEM/IS-MAINT =38 KEY/1
% SYSTEM/IS-MAINT =38 CL/DATA
% SYSTEM/IS-MAINT =38 ENTER COMMAND
SYSTEM/IS-MAINT =38 ACCEPT
38 AX CHANGE KEY/1 CL/KEY1
"KEY/1" CHANGED TO "CL/KEY1"
% SYSTEM/IS-MAINT =38 1 STRUCTURE CHANGED
% SYSTEM/IS-MAINT =38 ENTER COMMAND
SYSTEM/IS-MAINT =38 ACCEPT
38 AX NEXT
% SYSTEM/IS-MAINT =38 ENTER FILE NAME
SYSTEM/IS-MAINT =38 ACCEPT
38 AX EMPLOYEE/INFO
% SYSTEM/IS-MAINT =38 ENTER COMMAND
SYSTEM/IS-MAINT =38 ACCEPT
38 AX VERIFY
SYSTEM/IS-MAINT =38 "LINE" = "BACKUP.PRT/19"
SYSTEM/IS-MAINT =38 "BACKUP.PRT.19" RELEASED
% SYSTEM/IS-MAINT =38 VERIFICATION COMPLETE - 3 ERRORS
% SYSTEM/IS-MAINT =38 ENTER COMMAND
SYSTEM/IS-MAINT =38 ACCEPT
38 AX
SYSTEM/IS-MAINT =38 EOJ. TIME = 10:32:53.9

SYSTEM/IS-MAINT =38 ACCEPT
```

ERROR MESSAGES (ERRORS AND WARNINGS)

Error messages for the SYSTEM/IS-MAINT program are displayed on the ODT. There are two kinds: messages preceded by the word **ERROR** and messages preceded by the word **WARNING**.

An **ERROR** message is displayed when a command is entered incorrectly and cannot be executed. Explanations of all the possible **ERROR** messages are provided in the alphabetized list at the end of this subsection.

A **WARNING** message is displayed when a command being processed encounters problems in some part of the task. The commands **REMOVE**, **CHANGE**, and **UPDATE** are subject to **WARNING** messages because action may be required on more than one file.

WARNING Messages

Following are the possible messages that may be displayed:

WARNING – <file name-1> NOT REMOVED

WARNING – <file name-1> NOT CHANGED TO <file name-2>

WARNING – <file name-1> NOT UPDATED TO <file name-2>

The specific message displayed includes one of the following explanations: associate

MCPII DENIED REQUEST
NOT ON DISK
PACK NOT ON-LINE
ALREADY IN DICTIONARY
ALREADY ON DISK
VERSION MISMATCH
HAS USERS

ERROR Messages

Following are the **ERROR** messages and their descriptions.

ERROR – BOTH NAMES ARE THE SAME

The file names used in the **UPDATE** or **CHANGE** commands cannot be the same.

ERROR – CLUSTER FILE IN USE BY ANOTHER PROGRAM

No other users may access the cluster file while the SYSTEM/IS-MAINT program is accessing it.

ERROR – CLUSTER FILE IS CORRUPTED

The cluster file dictionary of subfiles could not be read.

ERROR – CLUSTER FILE NOT FOUND

In response to the **ENTER FILE NAME** prompt, a cluster file was specified that could not be found. The operator must ensure that the file is on the system.

ERROR – COMMAND NOT ALLOWED IN THIS CONTEXT

The program expected the entry of a file name. Commands acting on a file are not allowed.

ERROR – COMMAND REQUIRES PARAMETERS

The command just entered required parameters that were omitted. Refer to the appropriate command syntax.

ERROR – DIFFERENT PACK-IDS NOT ALLOWED IN CHANGE

The pack-ids used in the CHANGE command must be the same. Refer to the CHANGE command syntax.

ERROR – ID EXCEEDS 10 CHARACTERS

The pack-id, multifile-id, or file-id name cannot contain more than 10 characters each. Refer to <file id> of the File Specification Clause in the SYSTEM/COPY section for B 1000 file naming conventions.

ERROR – INVALID COMMAND, TRY "TEACH" FOR LIST OF VALID COMMANDS

Either the command was erroneously entered or the command is not recognized by this program. Enter the TEACH or HELP command for a listing of valid commands.

ERROR – MCP II DENIED COPY REQUEST

The MCP II failed to process this command, possibly because of invalid file or tape names. Refer to the SYSTEM/COPY program for further explanation.

ERROR – MISSING ENDING QUOTE

The file name requiring quoted text lacks the final quotation character.

ERROR – MORE PARAMETERS EXPECTED

The program expected more parameters than were entered. Refer to the command syntax or operating instructions.

ERROR – NAME CONTAINS ILLEGAL DELIMITER

Special characters within a file name such as the semicolon (;) require enclosing quotation marks. Ensure that file names conform to B 1000 file-naming conventions.

ERROR – NAMES HAVE UNMATCHED EQUALS

The UPDATE or CHANGE commands have been used improperly and the file names have mismatched equal sign (=) characters. Refer to the command semantics for additional information.

ERROR – NOT AN INDEX SEQUENTIAL FILE

The file is not index sequential.

ERROR – ONLY "ODT" OR "PRINTER" ALLOWED AFTER "LIST"

The ODT or line printer are the only permissible output destinations. Refer to the command syntax.

ERROR – ONLY THREE IDS ALLOWED IN NAME

The file name can contain only the pack-id, multifile-id, and file-id. Refer to the B 1000 file-naming conventions.

ERROR – STRUCTURE IS NOT PART OF THIS FILE

The structure was erroneously specified.

ERROR – UNEXPECTED TEXT FOLLOWING <COMMAND or NAME>

More than one command was encountered. Only one input is allowed with each AC or AX command.

ERROR – "/" IS INCORRECTLY USED

File names such as /A/B or A//B, for example, are not allowed. Ensure that the use of the slash (/) character conforms with file-naming conventions.

ERROR = "=" NOT ALLOWED AS <PACK-ID, MULTIPLE-ID, or FILE-ID>

In some cases, the equal sign (=) special character is not allowed with a replacement for the pack-id, multifile-id, and/or file-id. Refer to the command syntax.

INTERNAL FILES

The following is a list of the names and functions of each file used by the SYSTEM/IS-MAINT program.

Internal File Name	Function
IS-FILE	The cluster file that is currently being accessed.
COPY	Used to pass copy requests to the SYSTEM/COPY program.
SOURCE	If file-equated to a name other than IS__MAINT/IS__MAINT/IS__MAINT, this file is used for input commands instead of the ODT.

SECTION 38

SYSTEM/ISVERIFY

The SYSTEM/ISVERIFY utility program tests the integrity of an indexed sequential (ISAM) file as supported by the B1000 MCP. The SYSTEM/ISVERIFY program lists information from the global (cluster) file, optionally lists information from the index file(s), verifies the mapping of index files onto data files, verifies cluster file information, and validates the overall structure of the index files.

If an ISAM file passes all of the verification procedures correctly, the file is unlocked by the SYSTEM/ISVERIFY program and can be opened for input or update by other programs.

When errors are detected during the verification process, a warning is issued stating that unpredictable results can occur if the file is opened for update without first rebuilding the index files.

It is recommended that all ISAM files be verified with the SYSTEM/ISVERIFY program on a regular basis.

OPERATING INSTRUCTIONS

The name of the ISAM cluster file is required in order to execute the SYSTEM/ISVERIFY utility program. To verify a file enter the following program control instruction.

```
EXECUTE SYSTEM/ISVERIFY; FILE CLUSTER NAME <file-name>;
```

PROGRAM SWITCH SETTING

Switch

- | | |
|---|------------------------------|
| 0 | Index file(s) is not listed. |
| 1 | Index file(s) is listed. |

PROGRAM OUTPUT

The SYSTEM/ISVERIFY program produces informative messages, error messages and a print file.

1. Informative messages can consist of notification of situations that must be corrected prior to the actual verification process. For example, if the cluster file is in use, a message is displayed and the program terminates.

If no errors are found during the verification procedure, the following informative message is displayed:

```
VERIFICATION COMPLETED CORRECTLY
```

2. Fatal error messages are issued by the program as errors are encountered during the verification process. The presence of any of these fatal error messages requires that the ISAM file be re-created prior to further updating. Failure to do this can result in corrupt data files or system halts.
3. A print file is generated that can be used in error analysis whether or not the ISAM file verifies correctly. This file includes information from the global file, disk file header extensions, file tables and structures for the data and index files, as well as the messages previously described.

SAMPLE FILE OUTPUT

The following is a sample output print file for an ISAM file called "X". This file has one prime key and one alternate key for which duplicates are allowed. The verification for this file was not completed correctly and the file must be re-created before being opened for update. The data file is missing. Explanatory comments for some entries are in brackets.

LISTING OF CLUSTER FILE: X

GLOBAL INFORMATION

VERSION NUMBER	@07@
NUMBER OF USERS	@00@
NUMBER OF UPDATERS	@00@
DISK COPY	@F24037040@
SIZE	@0130@ (00304)
MEMORY ADDRESS	@000000@
LOCK BITS	@0@
IO DESC CHAIN	@000000@
MAX STRUCTURE NUMBER	@01@ (001)
DMS FLAG	@0@
AUDITED	@0@
OVERRIDDEN INTEGRITY	@0@
WRITE IN PROCESS	@0@
AUDIT WRITE REQUIRED	@0@
WRITE ERROR	@0@
CONCURRENT INFO	@00@
SYSTEM DESCRIPTOR(00)	@000000000000040000FA@
SYSTEM DESCRIPTOR(01)	@000000000000050000FA@

DFH EXTENSION INFORMATION

EXTENSION VERSION	@A4F461AD7@
NEXT FREE RECORD	@FFFFFFFF@
NEXT FREE BLOCK	@00000000@
ROOT TABLE	@FFFFFF@ (255, 65535)
UPDATE FLAG	@0@
EXTENSION VERSION	@A4F461AD7@
NEXT FREE RECORD	@FFFFFFFF@
NEXT FREE BLOCK	@00000000@
ROOT TABLE	@FFFFFF@ (255, 65535)
UPDATE FLAG	@0@

FILE TABLE INFORMATION

EXTENSION NUMBER	@00@	(000)		
EXTENSION OFFSET	@0001@	(00001)		
EXTENSION SIZE	@00A0@	(00160)		
FILE NAME	"U	(SMITH)	00X	"
EXTENSION NUMBER	@01@	(001)		
EXTENSION OFFSET	@0002@	(00002)		
EXTENSION SIZE	@00A0@	(00160)		
FILE NAME	"U	(SMITH)	01X	"

STRUCTURE INFORMATION

NUMBER	@00@			
TYPE	@5@	(DATA FILE)		
USER COUNT	@00@			
BUFFER LIST PTR	@FFFFFF@			
RECORDS PER BLOCK	@001@	(0001)		
SEGMENTS PER BLOCK	@01@	(001)		
RECORD SIZE	@0050@	(00080)		[in bits]
BUFFER SIZE	@0058@	(00088)		[in bits]
BLOCKS PER AREA	@01C6@	(00454)		
SEGMENTS PER AREA	@01C6@	(00454)		
DFH PTR	@000000@			
OFFSET TO EXTENSION	@0000@			
CURRENT POINTER	@000000@			
PRIME KEY	@0@	(NO)		
DUPLICATES ALLOWED	@0@	(NO)		
SIMPLE KEY	@1@	(YES)		
AUDIT FLAG	@0@			
AUDIT WRITE RQD	@0@			
DFH EXT CHANGED	@0@			
HEADER CHANGED	@0@			
SPLIT FACTOR	@000@	(0000)		
NUMBER OF KEYS	@00@			
KEY OFFSET	@0000@	(00000)		
SIZE	@000@	(0000)		

B 1000 Systems System Software Operation Guide, Vol. 2
SYSTEM/ISVERIFY

NUMBER	@01@		
TYPE	@6@	(INDEX FILE)	
USER COUNT	@00@		
BUFFER LIST PTR	@FFFFFF@		
RECORDS PER BLOCK	@0A7@	(0167)	
SEGMENTS PER BLOCK	@0E@	(014)	
RECORD SIZE	@0070@	(00112)	
BUFFER SIZE	@4936@	(18742)	[includes BCI]
BLOCKS PER AREA	@000A@	(00010)	
SEGMENTS PER AREA	@008C@	(00140)	
DFH PTR	@000000@		
OFFSET TO EXTENSION	@0000@		
CURRENT POINTER	@000000@		
PRIME KEY	@1@	(YES)	
DUPLICATES ALLOWED	@0@	(NO)	
SIMPLE KEY	@1@	(YES)	
SPLIT FACTOR	@06F@	(0111)	[in number of records]
NUMBER OF KEYS	@01@		
KEY OFFSET	@0000@	(00000)	[in bits]
SIZE	@050@	(0080)	[in bits]

WARNING: FILE "U/(SMITH)/00X" NOT PRESENT, OR LOCKED
VERIFICATION TERMINATED - IRRECOVERABLE ERROR

SECTION 39

SYSTEM/LDCONTRL

The SYSTEM/LDCONTRL (LOAD CONTROL) program is a normal-state utility program that creates pseudo-reader files. Physical card decks are the primary source for pseudo file input. Pseudo-reader files are often referred to as pseudo decks.

A pseudo deck (file) is a special disk file (FILE-processed as though it were a card deck. All system control messages in a pseudo deck are processed by the control card driver in the MCPPII; thus, only the first 72 columns of each record are processed. Each record of the input file is represented by a 96-column card image in the pseudo deck.

Invalid characters in control records are replaced by an EBCDIC question mark character (@6F@) in the pseudo deck.

Pseudo decks have a file identifier of DECK/<nnnnn>, where <nnnnn> is a unique five-digit number assigned by the MCPPII. Numbers are assigned sequentially, beginning with 00001 and incremented as each new file is assigned. The counter is reset when there are no pseudo decks in the disk directory, or when the count reaches 99999.

The CD, ED, IL, LD, OL, RD, and RN system commands operate on pseudo decks. Refer to section 5 of Volume 1 of the *B 1000 Systems System Software Operation Guide* for information on these commands.

OPERATING INSTRUCTIONS

Primary operation of the SYSTEM/LDCONTRL utility program is through the LD system command. An explicit EXECUTE SYSTEM/LDCNTRL command is also valid.

The SYSTEM/LDCONTRL program expects a card data file identified as ? DATA CTLDCK. This data file contains card files to be loaded into pseudo decks; one or more card files can be included in the input data file CTLDCK. Each card file must be identified with a ? DATA <file-name> card, and ended with a ? END card. The input data file CTLDCK must be terminated with a ? ENDCTL card.

The SYSTEM/LDCONTRL program recognizes all end-of-deck (? END) control cards. Each time an end-of-deck control card is encountered, the SYSTEM/LDCONTRL program closes the file that it is currently creating; thus, each card file is loaded as a separate pseudo deck. All control statements, including the end of deck card, that are included in each card deck are written to the pseudo deck along with all normal data cards. Control statements are cards that have an invalid character or a question mark (?) character in column 1.

Because the SYSTEM/LDCONTRL program accepts the control message ? END as program input, the special message ? ENDCTL must be used to terminate the input data file CTLDCK. An example of an input data file CTLDCK follows.

```
? DATA CTLDCK
? EXECUTE A/B;
? PRIORITY = 5;
? DATA CARDS
  data for file CARDS
? END %end of deck 1
? DATA INPUT
  data for file INPUT
? END
.
.
.
? ENDCTL
```

GENERAL RULES

The user should be aware of the following rules:

1. A card file labeled ?CTLDCCK cannot be read by a pseudo reader.
2. A maximum of 99,999 cards can be loaded into a single pseudo deck.
3. When reading from a pseudo reader, the MCPPII recognizes a question mark (?) character in column 1 of a card image as the control card indicator. Any invalid character in the input card file is translated to a question mark (?) character when the pseudo file is created.
4. The STREAM and TERMINATE cards replace DATA and END cards in the input data file CTLDCCK. In this case, the following four restrictions apply.
 - 1) The STREAM or TERMINATE card must contain an invalid character in column 1, and the keyword STREAM or TERMINATE must be the first token on the card after the invalid character in column 1.
 - 2) The file-id on the STREAM and TERMINATE cards must match, and the identifier on each TERMINATE card must match the identifier on the immediately preceding STREAM card, or the TERMINATE card is included in the pseudo deck.
 - 3) No embedded (nested) STREAM or TERMINATE cards are allowed.
 - 4) Any invalid character found in column 1 of the card file delimited by the STREAM and TERMINATE cards is translated to the null character (@00@) rather than the question mark (?) character; thus, these card images are not recognized as control cards when the pseudo deck is read.
5. Pseudo decks can only be removed by the ED and RD system commands. (Refer to section 5 of Volume 1 of the *B 1000 Systems System Software Operation Guide*.) The only way to prevent a pseudo deck from eventually being allocated to a pseudo reader and being read is to remove the deck.
6. When the SYSTEM/LDCONTRL program is ZIP-executed, either explicitly with the EXECUTE program control instruction or implicitly with the LD system command, job spawning information is included on the first record of each pseudo deck. If the spawning program has gone to EOJ when the pseudo deck is read, the MCPPII removes the pseudo deck without processing it.
7. Pseudo deck characteristics are:

Record size (bytes):	288	(three 96-character card images)
Records per block:	1	
Records per area:	100	
Multifile-id:	DECK	
File type:	3	(pseudo deck)
8. The internal identifier of the input file is CARD.
9. Copying a pseudo-deck file from disk to tape using the SYSTEM/COPY program is allowed; however, copying from tape to disk causes the pseudo-deck file to be loaded as a data file.

SECTION 40

SYSTEM/LOAD-CAS

System programs such as compilers, interpreters, object code, and system software can be loaded from a cassette to the system disk by using the SYSTEM/LOAD-CAS program. Cassettes used for input to this program must be created using the CASSETTE/MAKER utility program.

This program is executed from the Operator Display Terminal (ODT) using a standard EXECUTE statement. After the program goes to beginning of job (BOJ), an ACCEPT message is issued requesting the name of the file to be loaded. The file name can be entered as <multifile-id>/<file-id>, <multifile-id>/=, or =/=. The file name must conform to the file-naming conventions prescribed for B 1000 systems. After the program loads the specified file, it requests additional file entries. A null entry terminates the program.

Example:

```
EXECUTE SYSTEM/LOAD-CAS
SYSTEM/LOAD-CAS =73 BOJ.
% SYSTEM/LOAD-CAS =73 ENTER NAME OF FILE TO LOAD
SYSTEM/LOAD-CAS =73 ACCEPT.
73 AX COBOL/=
```

After a file or files have been loaded, the cassette must be rewound. If rewinding is neglected, an error message is issued when a subsequent program load is attempted. If more than one cassette is needed, the following message is issued.

```
MOUNT NEXT CASSETTE AND <mix-number>AX
```


ERROR MESSAGES

The following error messages are generated by the SYSTEM/LOAD-CAS program. The error messages appear alphabetically with their descriptions.

BAD HEADER

The reel number could not be read due to a bad header.

CASSETTE READ ERROR - <file-name> NOT LOADED

The hash-total is incorrect, possibly caused by a cassette not rewound or a bad cassette.

<file-name> NOT LOADED DUE TO OPERATOR INTERVENTION

File not loaded.

INCORRECT CASSETTE

PLEASE LOAD REEL NO. <reel #> AND <mix #> AX

Incorrect cassette.

INVALID FILE NAME

The file name did not conform with B 1000 file-naming conventions for one or more of the following reasons: (1) unmatched quotation mark (") characters, (2) more than 10 characters in the file-id, (3) the equals sign (=) character is used as a multifile-id only.

INVALID FIRST CHARACTER IN FILE NAME

None of the following characters may be used as the first character in a file name: percent sign (%), blank, semicolon (;), comma (,), virgule (/).

<file-name> NOT FOUND

The file was not on cassette.

SECTION 41 SYSTEM/LOAD-DUMP

PROGRAM PRODUCT NO LONGER SUPPORTED

As of the Mark 12.0 software release, the SYSTEM/LOAD-DUMP program is no longer supported by the B 1000 system software.

SECTION 42

SYSTEM/LOGOUT

The SYSTEM/LOGOUT program is a system utility program that produces a formatted analysis of the operating log. The MCP II maintains the system log when the LOG option is set. The SYSTEM/ELOGOUT program accesses a transferred copy of the system log. Refer to the LG and LN system commands in section 5 of volume 1 of the *B 1000 Systems System Software Operational Guide*, form number 1169000 for information on transferring the system log file.

OPERATING INSTRUCTIONS

Primary operation of the SYSTEM/LOGOUT program is through the LG and LN system commands. An EXECUTE SYSTEM/LOGOUT command can also initiate the program. The internal file-id of the logfile is LOGFILE.

PROGRAM OUTPUT

The SYSTEM/LOGOUT program writes directly to a line printer. If no line printer is available, a backup print file is generated.

Information that is produced includes CLEAR/START records, job information records, DMS information, file information, and comment records.

The twelve categories that are written by the MCP to the system log are listed next, followed by paragraphs detailing each category.

1. Clear/Start
2. Job schedule
3. Job BOJ
4. File open
5. File close
6. DMS database open, close
7. DMS structure invocation open, close
8. Peripheral assignment for initializer I/O
9. Peripheral release from initializer I/O
10. Job EOJ
11. Log transfer
12. Comment

CLEAR/START Records

The SYSTEM/LOGOUT program generates three lines of output for each CLEAR/START record in the log file. This output describes (1) the date and time of the CLEAR/START operation, (2) the name and version of the MCP II being used and the amount of S-memory in bits, and (3) the names of the interpreter and GISMO being used by the MCP.

Job Information Records

The SYSTEM/LOGOUT program combines several log records to generate information on each job processed by the system. The following items of job information are produced:

1. The job number of the program that is executing.
2. The charge number used to execute the job.
3. The full name of the program.
4. The BOJ date and time.
5. The elapsed time for the program, in seconds.
6. Any abnormal job termination messages.
7. The type of termination (for example, NORMAL, ABORTED).
8. The termination (EOJ) date and time.
9. The compilation date and time of the executed program.
10. The processor and memory priorities assigned at BOJ.
11. The date and time that the program was scheduled.
12. The interpreter used.
13. The type of execution (for example, EXECUTE, COMPILE).
14. The number of files declared in the program.
15. The amount of static memory required by the program.
16. The amount of dynamic memory (data overlay space) used by the program
17. The total memory used by the program for its run structure (Base/Limit area).

Some information is given by environments. The MCP maintains this information for the primary, DMS, and IBASIC environments. The information given for each environment includes interpreter time (accumulated processor time for each environment), MMCP time (accumulated MMCP processor time for each environment), SMCP time (accumulated SMCP processor time for each environment), code overlays (the number of code overlays for each environment), and data overlays (the number of data overlays for each environment).

DMS Information

Database open and close records are combined to yield the following information:

1. The logical database name.
2. The physical database name.
3. The open time.
4. The unit and hardware type.
5. The close time.
6. The pack name.
7. The number of non-update operations.
8. The number of update operations.
9. The elapsed open time.
10. The processor time.
11. The number of exceptions.
12. The number of transactions.
13. The transaction state time.
14. The number of contentions.
15. The contention wait time.
16. The number of I/O waits.
17. The I/O wait time.

DMS structure invocation and close records are combined to yield the following information:

1. The structure name.
2. The invoked time.
3. The close time.
4. The database name.
5. The job number.
6. The file name.
7. The unit and hardware type.
8. The number of random and sequential FINDs.
9. The number of INSERTs, UPDATEs, DELETEs, SYSTEM CHANGEs, and EXCEPTIONs.
10. The number of logical and physical READs and WRITEs.

File Information records

The log file contains information about files used during execution of a program. Files that are not accessed are not logged. The SYSTEM/LOGOUT program combines file open and close records to generate the following information about each file.

1. The internal file name.
2. The external file identifier.
3. The record size, in bytes.
4. The hardware device type assigned to the file.
5. The number of buffers assigned to the file.
6. Either (a) the number of records per block (fixed length files), or (b) the maximum size of a physical block (variable length records).
7. The file type (for example, CODE, RANDOM).
8. For disk files only, the number of blocks per disk area and the number of disk areas. For non-disk files, this field is blank.
9. The memory space required to open this file.
10. For tape files only, the serial number of the last reel of tape accessed by the file. For non-tape files, this field is blank.
11. The number of records in the file.
12. The number of blocks in the file.
13. The time and type of each OPEN performed.
14. The time and type of each CLOSE performed.
15. The total elapsed time that the file was open.
16. The hardware unit name where the file now resides. For temporary files, this field is blank.
17. The number of times the program had to wait for protection = protected (if the number is greater than zero).

Comment Records

Operator-supplied comments, inserted into the log by means of the LC (log comment) system command are printed along with the time and date of the insertion.

SECTION 43

SYSTEM/MAKEUSER

The SYSTEM/MAKEUSER program is a normal-state utility program used to create, access, or modify the (SYSTEM)/USERCODE file, a file describing all usercode and password combinations that are valid on the system. The (SYSTEM)/USERCODE file maintains all information needed by the MCP II to support the file security mechanism, including usercodes and their associated passwords, default disk-identifiers, charge numbers, maximum allowable run-time priorities, and default security codes. A maximum of 1023 usercode/password combinations is allowed in the (SYSTEM)/USERCODE file.

OPERATING INSTRUCTIONS

All communication and control of the SYSTEM/MAKEUSER program is performed through the ODT except for certain automatic features described in the following paragraphs. AC or AX commands are used to input commands and data to the program. Success/failure messages are displayed on the ODT after each command has been processed, as follows:

EXECUTE SYSTEM/MAKEUSER

SYSTEM/MAKEUSER=1 BOJ....

SYSTEM/MAKEUSER=1 ACCEPT.

1AX <command> <optional comment>

%SYSTEM/MAKEUSER=1 <success/failure message>

The <optional comment>, if present, must begin with a percent sign (%) character.

Several automatic features are included in the SYSTEM/MAKEUSER program and are described next.

1. When a card file labeled NEW/USER-CODES (such as is produced by the PUNCH command) is present at beginning of job (BOJ) or after execution of any command, the SYSTEM/MAKEUSER program automatically performs a CREATE command followed by a list of the newly created (SYSTEM)/USERCODE file. The SYSTEM/MAKEUSER program then goes to end of job (EOJ). No operator action or intervention is required. Refer to the CREATE command for information describing the required input file.
2. When the SYSTEM/MAKEUSER program is executed with SWITCH = L, a listing of the current (SYSTEM)/USERCODE file is produced without operator intervention, and the SYSTEM/MAKEUSER program then goes to end of job (EOJ).

Example:

```
EXECUTE SYSTEM/MAKEUSER SWITCH=L;
```

3. When the SYSTEM/MAKEUSER program is executed with SWITCH = L, the current (SYSTEM)/USERCODE file is punched to a card deck labeled NEW/USER-CODES. No operator intervention is allowed, and the SYSTEM/MAKEUSER program goes to end of job (EOJ) after card deck is punched.

Example:

```
EXECUTE SYSTEM/MAKEUSER SWITCH=P;
```


FILE SECURITY ATTRIBUTES

The valid file security attributes are described in table 43-1. along with their default values. The usercode (US) and password (PW) attributes have no default and must be specified explicitly for every usercode included in the (SYSTEM)/USERCODE file. A null ("") password is valid, but must be explicitly specified.

Table 43-1. SYSTEM/MAKEUSER Security Attributes

Option	Description	Default Value
US = <usercode>	Usercode	No default
PW = <password>	Password	No default
PACK = <pack-id>	Default pack-id	System disk (" ")
CHG = <integer>	Default charge number	Zero (0)
PRI = <integer>	Maximum priority	Seven (7)
*PRIV	PRIVILEGED indicator	Not PRIVILEGED
*NONPRIV	NON-PRIVILEGED indicat.	Not PRIVILEGED
PUBLIC	File security	PRIVATE
PRIVATE	File security	PRIVATE
SL = <integer>	Security level	Zero (0)
MAXTIME = <integer>	Default maximum execution time	Zero (0) (infinite)
HOSTNAME = <identifier>	BNA hostname	*NONE

The <usercode> can be up to eight characters in length. A <usercode> of SYSTEM is not allowed. Restrictions in the <usercode> format are:

1. An asterisk must not be the beginning character.
2. A left parenthesis must not begin or a right parenthesis end the <usercode>.
3. Special characters (other than A through Z and 0 through 9) are permitted in a <usercode>; however, unpredictable results can occur if special characters are misused.

The <password> can be up to ten characters in length; it can also be null (""), indicating that no password is required for the associated <usercode>.

Multiple entries with the same <usercode> are allowed; however, each must have a unique <password>. In addition, multiple entries with the same <usercode> must all specify the same default <pack-id> and security (PUBLIC or PRIVATE).

The security level attribute defines the multifile identifiers that a usercode can access. The values can be 0, 1 or 2. A value of 0 allows access to any multifile identifier. A value of 1 allows access to a multifile identifier that is a usercode, for example (PAYROLL). A value of 2 allows access to a multifile identifier that is the same as the usercode supplied with the input command or the same as the usercode under which the job is executing.

The MAXTIME attribute is used by the MCPII in determining the maximum amount of processor minutes for a job. The maximum value of <integer> is 27962.

The HOSTNAME attribute relates only to BNA host services. <identifier> is a string of up to 17 characters for use as a BNA hostname. Lower-case characters are translated to upper-case characters and the underscore (_) character is translated to a minus sign (-) character. The values *ANY or *NONE can be used instead of an identifier.

*ANY means that the associated usercode/password is valid from any BNA host as well as from the local host.

*NONE means that this usercode/password is invalid from all BNA remote hosts. The usercode/passord is still valid from the local host.

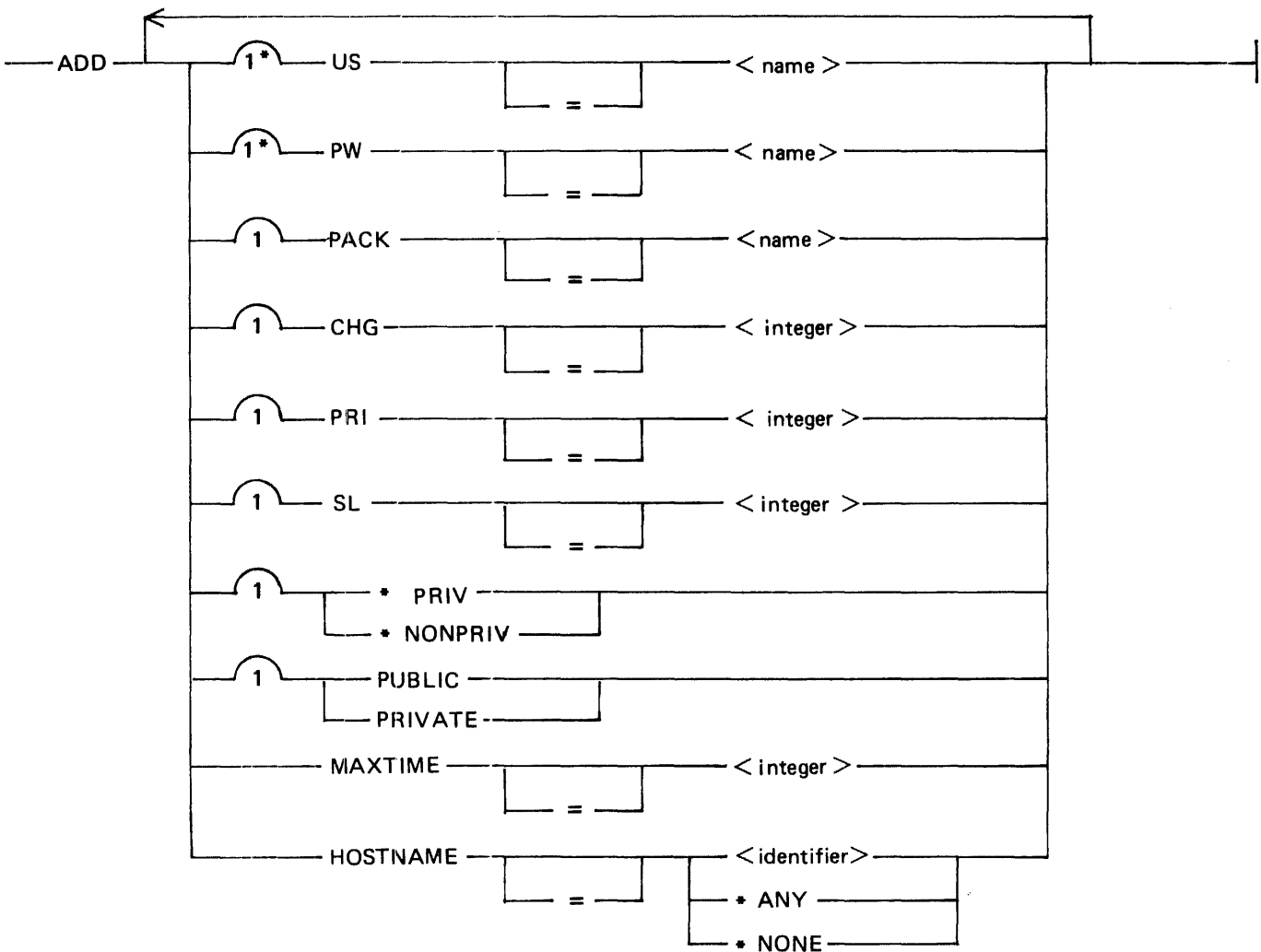
COMMANDS

The syntax, semantics, and examples of actual input commands are documented in this section, along with a description of the input specification format. Valid commands are ADD, CHANGE, COPY, CREATE, DEBUG, DELETE, DISPLAY, END, EOJ, HELP, LIST, PUNCH, and PURGE. The commands are presented in alphabetical order.

ADD Command

The ADD command allows entries to be added to the (SYSTEM)/USERCODE file.

Syntax:



The usercode (US) and password (PW) options are both required; any other options described File Security Attributes are allowed.

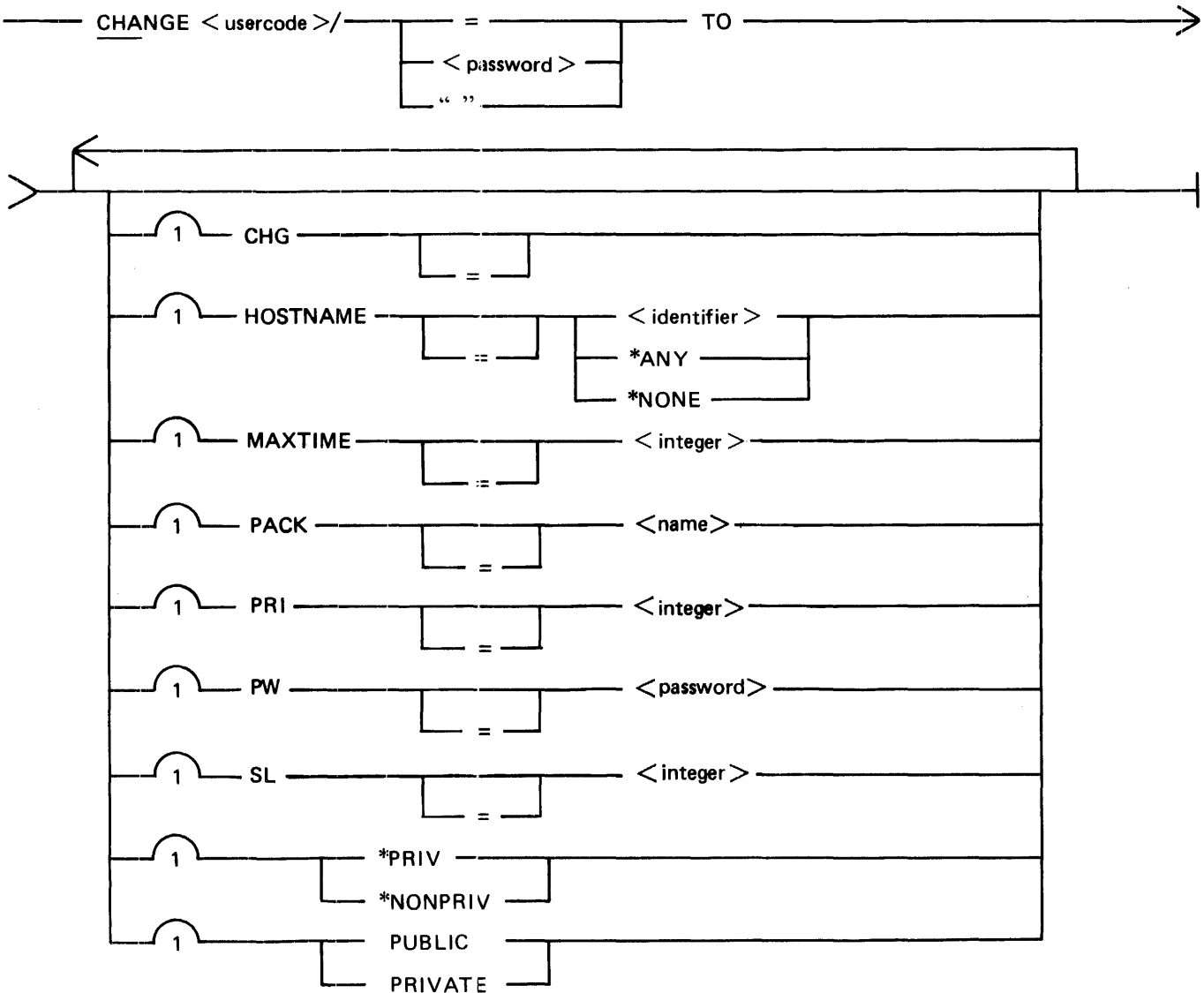
Examples:

```
1AX ADD US=NEWUSER PW=NEWPW CHG=1234 PRI=5 PACK=USER
1AXADD CHG=50 US=SITE PW=""
1AX ADD US=SITE PW=PRIV PRI=15           %System disk is default
1AXADD US=FINANCE PW=VP                  %All other options default
1AX ADD US REMOTE PW BNA HOSTNAME *ANY  %*ANY means all remote BNA
                                           %hosts
1AX ADD US STUDENT PW 3 MAXTIME = 3     %Maximum execution time
                                           %is 3 minutes
1AX ADD US PAYROLL PW PAY SL = 1        %Allows access to a
                                           %multifile that is a usercode
```

CHANGE Command

The CHANGE command allows the attributes of existing usercodes to be modified.

Syntax:



The usercode-specifier must have one of the following formats:

<usercode>/<password>
 <usercode>/""
 <usercode>/=

The first two formats are used to change a specific entry in the (SYSTEM)/USERCODE file (the "" option is for usercodes having a null password). The "=" option is used to change all entries having the specified <usercode>.

Only the attributes specified in the CHANGE command are modified. All file security attributes except the usercode (US) can be modified with the CHANGE command, with the following two restrictions.

1. The password (PW) option is not allowed if the /= format of the usercode-specifier is used.
2. The PUBLIC, PRIVATE, *PRIV and *NONPRIV options are allowed only when the /= form of the usercode-specifier is used.

WARNING

The CHANGE command must not be used when other programs are executing.

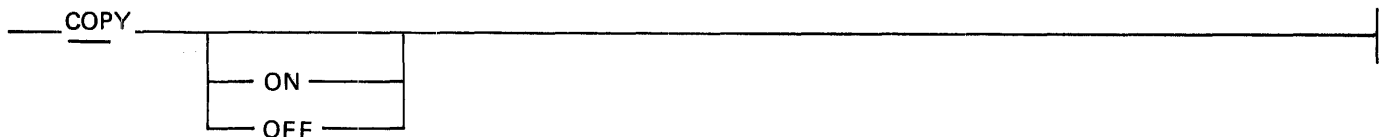
Examples:

```
1AX CHANGE FINANCE/VICEPRES TO PW=VP CHG=1234 PRI=10
1AXCHA SITE/= TO CHG=55555
1AXCHANGE NEWUSER/'''' TO PW=NEWPW PACK=''''
1AX CHANGE STUDENT/1 TO MAXTIME = 3
1AX CHANGE CLASS/= TO HOSTNAME *NONE % All usercodes are now denied
                                     % execution from all remote BNA
                                     % hosts.
1AXCHA SITE/= TO CHG=55555
```

COPY Command

The COPY command controls listing of all ODT input and output messages on the line printer.

Syntax:



By default, the COPY option is ON, and all SYSTEM/MAKEUSER ODT messages are listed on the line printer. When the COPY option is turned off, the printer listing is suppressed.

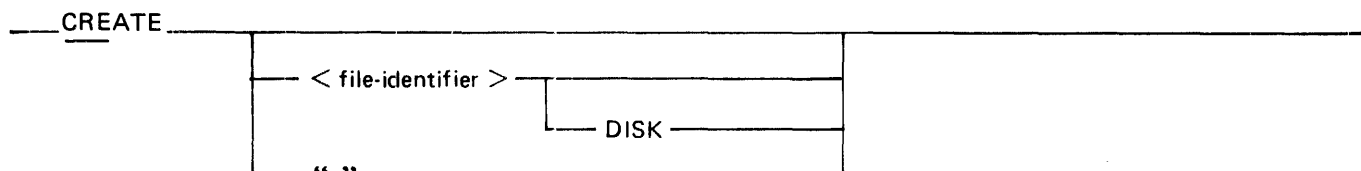
Example:

```
1AXCOPY OFF
```

CREATE Command

The CREATE command causes a new (SYSTEM)/USERCODE file to be built which can be entered into the US slot of the Name Table with the CM system command. (Refer to section 5 of Volume 1 of the *B 1000 Systems System Software Operation Guide*.)

Syntax:



If no options follow the keyword CREATE, then the input file-identifier is assumed to be CARD. The input file is assumed to be a card file unless the keyword DISK is specified.

The null ("") CREATE option allows users to create a usercode file through the ODT alone. The input specification CREATE "" produces a usercode file that contains one entry: the privileged usercode/password pair DEFAULT/PRIVILEGED. This privileged usercode is valid for all usercode operations.

The SYSTEM/MAKEUSER program enters the new (SYSTEM)/USERCODE file into the US slot of the Name Table. If a (SYSTEM)/USERCODE file already exists, it is removed from disk and replaced by the new (SYSTEM)/USERCODE file. The SYSTEM/MAKEUSER program and the SYSTEM/ODT program must be the only programs executing when the CREATE command is used to replace an existing (SYSTEM)/USERCODE file, or the replacement is not performed.

The input file for the CREATE option contains input specifications in a free-form format, 80-character record. Each file security attribute that is specified must be preceded by the keyword that identifies the attribute. Options can be entered in any order, but all attributes for any usercode must be contained on one record. The usercode (US) and password (PW) attributes must be included on every record. Other attributes can be included as desired; if omitted, the default values apply. Refer to File Security Attributes in this section for a description of the attributes.

NOTE

The CREATE command can be executed only when SYSTEM/MAKEUSER is the only program executing. If any other programs are executing, the newly created (SYSTEM)/USERCODE file is NOT entered into the US slot of the Name Table.

Examples of input records in the card file CARD follow:

```
US=USERA PW=X CHG=123456 PRI=4 PACK=USERPACK *PRIV
CHG=1000 PRI=8 US=USERB PW=ME
US=USERA PW="" PACK=USERPACK
US=SITE PW="" CHG=999999 PRI=15 *PRIV PUBLIC
US=USERB PW=PW3 PACK=""
```

Examples of ODT input commands:

```
1AXCREATE
1AX CREATE CARDFILE
1AXCRE USERDISK/USER/CODEFILE DISK
1AX CREATE "" %CREATES DEFAULT USERCODE FILE
```

DEBUG Command

The DEBUG command controls listing of debugging output on the line printer. The command is intended for system software development and debugging and should not be used in normal operation.

Syntax:

```
___ DEBUG _____|
      |
      | ON _____|
      | OFF _____|
```

DELETE Command

The DELETE command allows removal of existing entries from the (SYSTEM)/USERCODE file.

Syntax:

```
___ DELETE < usercode-specifier > _____|
   DEL
```

The usercode-specifier is specified in the same manner as is shown for the CHANGE command. The "=" option deletes all entries with the specified <usercode>.

WARNING

In rare cases, deleting an active usercode can cause the system to halt. Be sure that no programs are scheduled or running when deleting a usercode/password.

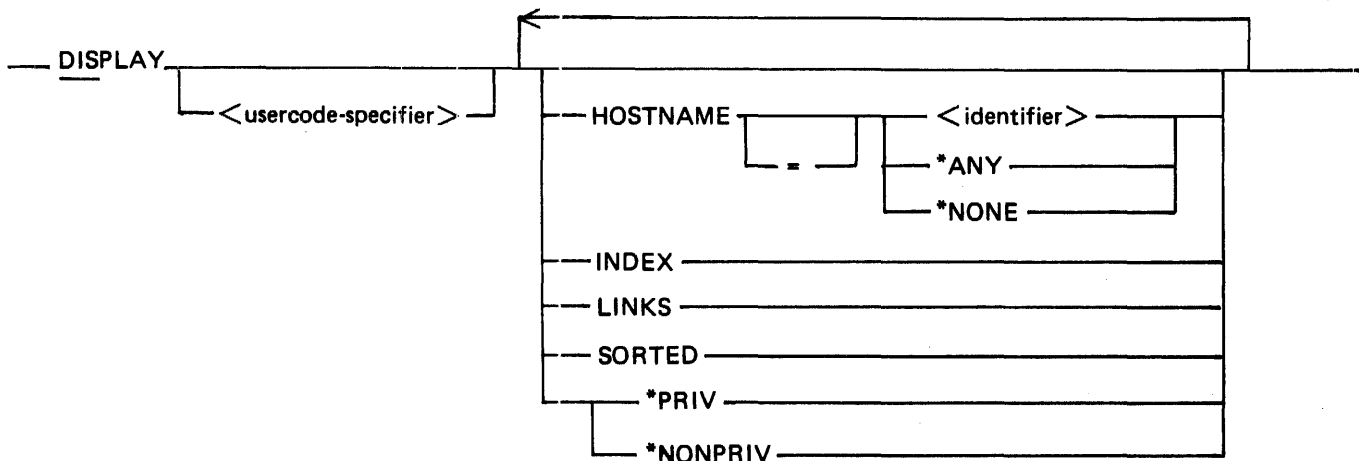
Examples:

```
1AXDELETE USER1/PASS1
1AX DEL USER2/=
1AXDEL USER3/""
```


DISPLAY Command

The DISPLAY command displays the existing (SYSTEM)/USERCODE file on the ODT.

Syntax:



If no options are specified, the DISPLAY command causes the entire file to be displayed. If the <usercode-specifier> is entered, only the specified usercodes are displayed. The use of the *PRIV option allows the display of only privileged usercodes. The HOSTNAME option allows usercodes on the specified BNA host to be displayed. If the LINKS option is specified, the usercode file is displayed on the ODT by usercode index. The SORTED option causes the usercode file to be displayed in alphabetical order with the link fields. If the INDEX option is specified, the usercode index is displayed on the ODT. The display may be interrupted by supplying the SYSTEM/MAKEUSER program with another AC or AX system command.

Example:

```

    DISPLAY HARRY/= *NONPRIV      %Display those usercode/password
                                %pairs of HARRY which are
                                %nonprivileged.

    DISPLAY *PRIV HOSTNAME *ANY   %Display only those entries
                                %which are privileged and valid
                                %from the remote host.

    DIS LINKS SORTED              %Display the entire usercode file
                                %in alphabetical order with the
                                %link fields.

                                %nonprivileged.
  
```

END/EOJ Command

The END/EOJ command causes the SYSTEM/MAKEUSER program to go to end of job.

Syntax:



Example:

2AX END % THIS IS THE SAME AS "EOJ"

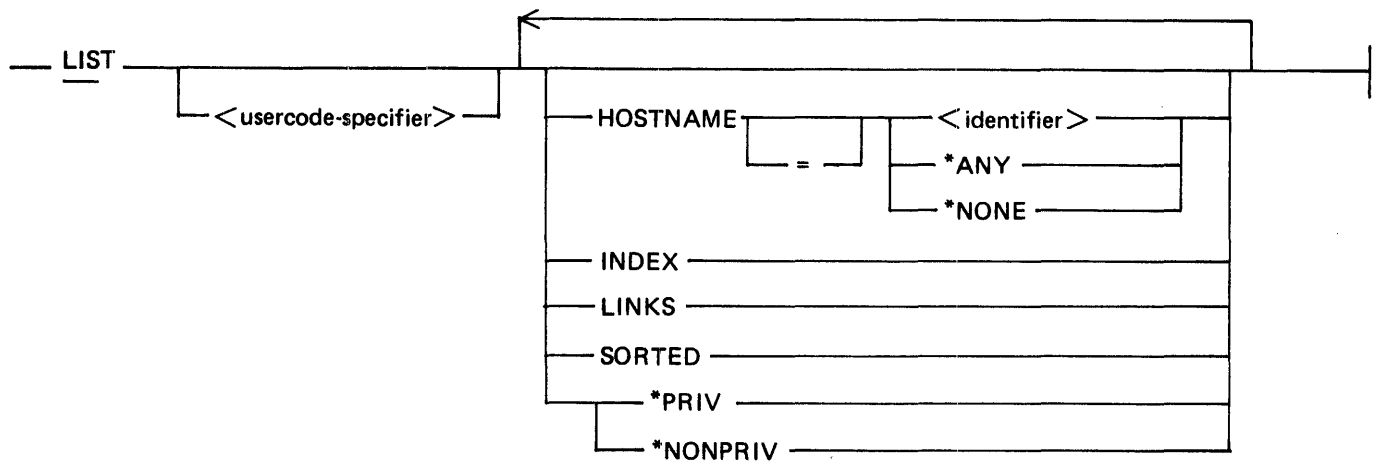
HELP Command

Displays the list of commands, followed by a list of keywords.

LIST Command

The LIST command allows listing of all or part of the existing (SYSTEM)/USERCODE file on the line printer.

Syntax:



If no option is specified with the LIST command, the (SYSTEM) /USERCODE file is sorted by usercode and the entire file is listed on the line printer. The <usercode-specifier> option allows the listing of only the designated entries. If the LINKS option is specified, the usercode file is listed on the line printer by usercode index. The SORTED option causes the usercode file to be listed on the line printer in alphabetical order with the link fields. If *PRIV is specified, only the entries for usercodes marked as PRIVILEGED are listed. Specifying *NONPRIV causes only those usercodes marked NONPRIVILEGED to be listed on the line printer. If a usercode family is specified in the LIST command, the usercode family can be listed only by usercode index. If INDEX is specified, the actual usercode index is listed.

The pack-id listed for all usercodes which have the default pack assignment of the system disk is *SYS DISK*.

Examples:

1AX LIST

1AX LIS USER1/=

1AXLIST SITE/PRIV

1AX LIS USERA/""

1AXLIST *PRIV

1AX LIST HOSTNAME HUB

% Lists only the entries which are
% valid from the remote host called
% HUB.

1AX LIST REX/= HOST SA1

% Lists only those entries of REX
% which are valid from the remote
% host SA1.

1AX LIST *NONPRIV HOSTNAME USER3

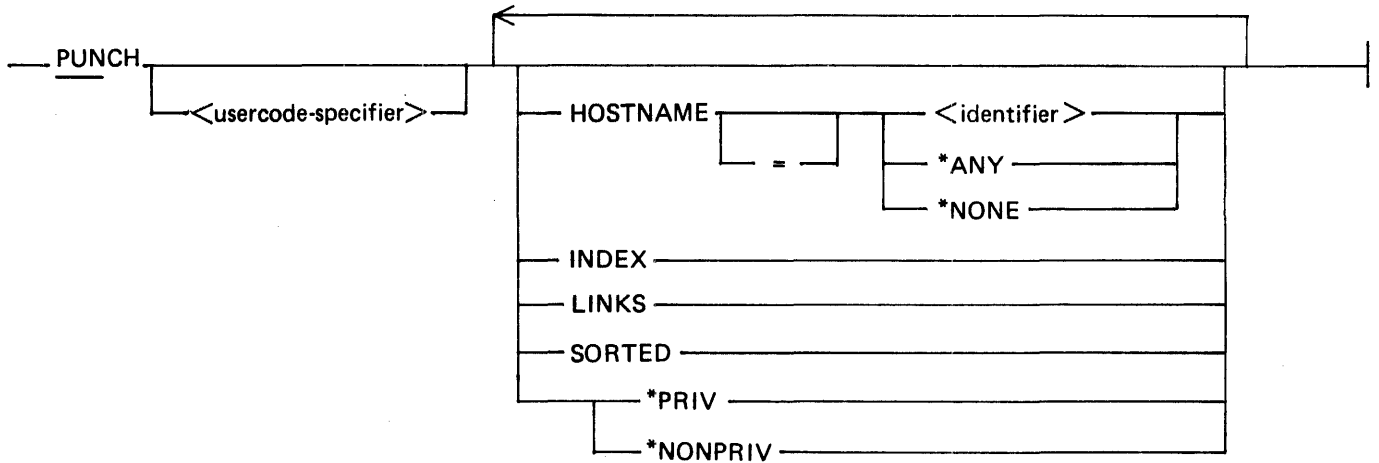
% Lists only those entries which are
% nonprivileged and which are valid
% from remote BNA host USER3.

% HUB.

PUNCH Command

The PUNCH command punches a card deck containing the current (SYSTEM)/USERCODE file. The PUNCH option can be requested as an automatic function by executing the SYSTEM/MAKEUSER program with SWITCH = P.

Syntax:



The output card file is labeled NEW/USER-CODES and is acceptable as input to the CREATE command (either explicit or automatic). The usercode-specifier and *PRIV options have the same meaning as they do when used with the LIST command.

Example:

```

1AX PUNCH
275AX PUNCH USER1/=
17AX PUNCH SITE/PRIV
2468AX PUNCH *PRIV
  
```

The PURGE Command

The PURGE command removes the file named (SYSTEM)/USERCODE and deletes the usercode file from the US slot in the name table. The PURGE command is used when the SYSTEM/MAKEUSER program is the only job running. The PURGE command has no options.

Syntax:



ERROR MESSAGES

The SYSTEM/MAKEUSER program provides the following error messages:

CANNOT DELETE "<usercode>" SINCE <n> JOB(S) RUNNING OR SCHEDULED WITH THIS USERCODE

UNKNOWN COMMAND "<command>", TRY ONE OF "CHANGE, ADD, DELETE CREATE, LIST,DEBUG,PUNCH,COPY,END, EOJ,PURGE,DISPLAY,HELP."

COMMANDS MUST BE FROM 3 TO 6 CHARACTERS IN LENGTH.

Enter a valid command or command abbreviation.

UNRECOGNIZED COMMAND/KEYWORD <command> OR TOO MANY PARAMETERS FOR THIS COMMAND.

Enter a valid command with all required options specified.

PARAMETER REQUIRED AND NOT FOUND FOLLOWING <input parameter>.
REQUIRED PARAMETERS WERE OMITTED FOR THIS COMMAND.

NUMBER FIELD FOR "CHARGE NUMBER" [or "PRIORITY"] TOO LARGE.

The maximum allowable value for CHARGE is 9999999, and the maximum allowable value for PRIORITY is 15.

"(SYSTEM)/USERCODE" FILE NOT ON DISK, COMMAND IGNORED.
ACREATE IS REQUIRED.

If the NEW/USER-CODES file is not available as a disk file or as a card file, use the null option of the CREATE command to produce a single-entry usercode file that describes the usercode/password DEFAULT/PRIVILEGED.

"(SYSTEM)/USERCODE" FILE LOCKED, COMMAND IGNORED.

MAXIMUM FILE SIZE OF 1024 ENTRIES EXCEEDED, COMMAND TERMINATED.

INVALID USERCODE "<usercode>" ENTRY DISCARDED.

DIFFERENT PACK NAME FOR SAME USERCODE.
<usercode> ENTRY DISCARDED.

FILE NAME MUST BE SPECIFIED WITH "DISK" OPTION, COMMAND IGNORED.

If DISK is specified with the CREATE command, a <file-identifier> must also be supplied.

PACK NAME IS INVALID FOR CARD FILES, COMMAND IGNORED.

If a <pack-id> is included in the <file-identifier> specified in the CREATE command, DISK must also be specified.

INPUT FILE SPECIFIED IS NOT ON DISK, COMMAND IGNORED.

The disk file specified in the CREATE command does not exist.

NO USERCODE FILE PRESENT, COMMAND IGNORED.

A CREATE must be performed to create a usercode file.

SPECIFIED ENTRY DOES NOT EXIST, COMMAND IGNORED.

The requested <usercode-specifier> is not in the (SYSTEM)/USERCODE file.

NUMERIC FIELD CONTAINS NON-NUMERIC CHARACTERS.

INVALID DELIMITER "<delimiter>", COMMAND IGNORED.

CHANGE TO PACKNAME REQUIRES "<usercode>/=".

"<usercode>/<password>" ALREADY EXISTS.

Cannot ADD a <usercode>/password> entry that already exists in the (SYSTEM)/USERCODE file.

CANNOT CREATE NEW USERCODE FILE WHEN OTHER JOBS ARE RUNNING.

The SYSTEM/MAKEUSER program must be the only program in the job mix if the existing usercode file is to be replaced with a new usercode file during execution of a CREATE command. The OVERRIDE option is not valid.

CANNOT CREATE USERCODE FILE WITH NO ENTRIES. USE CREATE "" FOR DEFAULT.

SECURITY MISMATCH – MIXED "PRIVATE" AND "PUBLIC" NOT ALLOWED.

CHANGE TO SECURITY REQUIRES "<usercode>/=".

CANNOT CHANGE ALL PASSWORDS.

CANNOT SPECIFY BOTH "PUBLIC" AND "PRIVATE".

CANNOT SPECIFY BOTH "**PRIV" AND "**NONPRIV".

REMOTE EXECUTION DENIED.

ILLEGAL USERCODE.

SECTION 44

SYSTEM/MARK-SEGS

The SYSTEM/MARK-SEGS program is a normal-state utility program that marks program code segments as important (1) or unimportant (0) for use with the Extended Segment Decay facility of the MCP11 Priority Memory Management System. By default, all code segments are marked as "unimportant" by the compilers. Segment and segment-page numbers are noted on the compilation listing of the program.

Appendix A of Volume 1 of the *B 1000 Systems System Software Operation Guide* provides further details on Extended Segment Decay and Priority Memory Management.

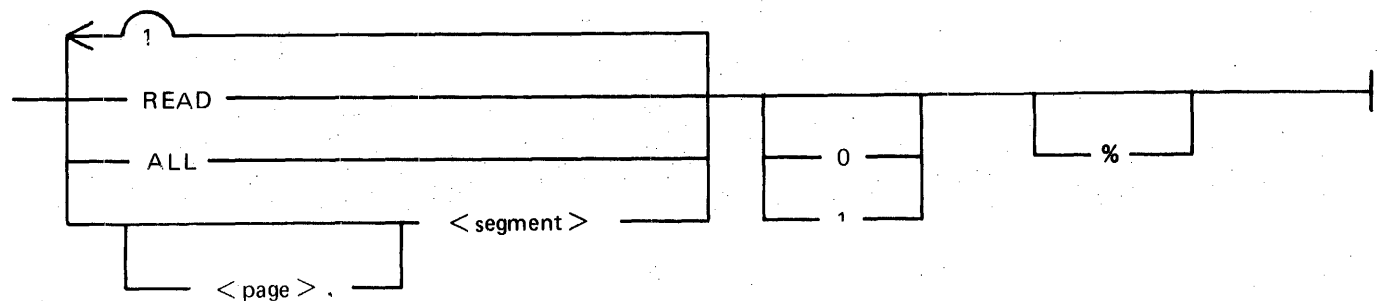
OPERATING INSTRUCTIONS

The SYSTEM/MARK-SEGS program marks only code (object) files. The file identifier of the file to be marked must be file-equated to the name CODE when the program is executed. The program expects to receive specification strings from a card file labeled CARD; however, if that file is not present when the SYSTEM/MARK-SEGS program begins execution, the program prompts for input specifications and expects to receive the specification strings by means of AC or AC commands from the ODT. Card input can be forced by entering the keyword READ (R) with first AX command.

INPUT SPECIFICATIONS

The SYSTEM/MARK-SEGS program expects free-form format input in the first 72 columns of each input specification card. Multiple specifications can be entered in each input card or AX command. A percent sign (%) character causes scanning on that input string to be terminated, allowing comments to be included. A description of the specification string syntax follows.

Syntax:



The keyword READ forces input specifications to be read from cards. This keyword must be used only if the SYSTEM/MARK-SEGS program is executed from the operator display terminal (ODT) and input specifications are to be entered via a card reader.

ALL causes every code segment in the program to be marked as important (1), or unimportant (0). When used, this option must be the first input specification (after READ).

The <page> and <segment> specifications allow individual segments to be marked. These numbers must be specified as integer values in either decimal or hexadecimal notation. The numbers must be enclosed in parentheses and must be followed by a 1 or a 0.

The SYSTEM/MARK-SEGS program ignores invalid (page, segment) specifications; furthermore, it terminates scanning the current specification string at the point where it encounters invalid specifications.

Page numbers are required if the source language of the program includes page numbers in its segment notation. Page numbers are optional only when they do not apply to the source language.

A null or blank specification string generates a printer listing of the status of all segments of the specified code file and terminates the SYSTEM/MARK-SEGS program.

An example of the sequence follows.

Example:

```
? EX SYSTEM/MARK-SEGS
? FILE CODE NAME TEST/PROGRAM
? DATA CARD
(0,1) 1 % Specification
(2, @A@) 1 (1,7) 1 % Strings
(@F@,5) 0
? END

EX SYSTEM/MARK-SEGS FI CODE NAME PAYROLL/APPLICATION
SYSTEM/MARK-SEGS=7385.BOJ
SYSTEM/MARK-SEGS=7385 ACCEPT
7385AXREAD
.
.
SYSTEM/MARK-SEGS=7385.E0J
```

PROGRAM OUTPUT

The SYSTEM/MARK-SEGS program generates a listing that notes errors in the specification strings; additionally, the report on each segment's status, if requested, appears in this listing. This listing goes directly to a line printer if one is available; otherwise, the report is saved in a backup file.

SECTION 45

SYSTEM/ODT

The SYSTEM/ODT program handles all required I/O operations on the Operator Display Terminal (ODT). The program is loaded automatically by the COLDSTART process, entered in the appropriate slot in the Name Table, and executed automatically by the MCP during the CLEAR/START process.

It is recommended that the SYSTEM/ODT program be run at a high priority. Operators can adjust this priority in accordance with individual needs. Thus, for example, if response from remote terminals is the most important activity on the system, the priority of the network controller can be set to a value equal to or higher than the priority of the SYSTEM/ODT program.

NOTE

Because it is a normal-state program, SYSTEM/ODT response to ODT commands may be slightly slower if system activity is heavy. To prevent degradation of response, the program is placed at one end of memory to avoid checker-boarding, and it is frozen in memory. Also, the highest priority in its list of tasks is to accept operator input. Even if the system is so busy that ODT output response is significantly degraded, the SYSTEM/ODT program generally accepts operator input and conveys it to the MCP for action.

OPERATING INSTRUCTIONS

No operating instructions are required for this program. It is executed automatically by the MCP during the CLEAR/START operation. Should the program terminate for any reason, this occurrence is recognized by the MCP and the program is automatically executed again.

NDL Handler (B 1990 Systems)

On B 1990 systems, the SYSTEM/ODT program opens a remote file to communicate with the remote ODT. This necessitates the inclusion of an operational NDL handler in Name Table slot C. The COLDSTART operation loads a default NDL handler, named SYSTEM/CONTROLLER. Section 6 of Volume 1 of the *B 1000 Systems System Software Operation Guide* provides information on declaring an NDL handler.

KB System Command

The KB system command is used to specify certain characteristics for displaying the ODT information. This command is completely described in section 5 of Volume 1 of the *B 1000 Systems System Software Operation Guide*.

FILES (REFERENCE INFORMATION)

The SYSTEM/ODT program contains six file declarations. Modification of the attributes of the files is strictly prohibited in most cases.

The SYSTEM/ODT program communicates with the MCP through two queue files. One of the queue files is used to transfer system output messages that are to be displayed on the screen; messages that are entered by the operator are transferred in the other queue file. The SYSTEM/ODT program performs all I/O operations on the ODT through direct communication with GISMO or the NDL handler.

In addition to the two queue files, the SYSTEM/ODT program also contains a declaration for a line printer file. This file is activated by a KB LP ON instruction. Also, since the line printer file is handled in the same manner as any other line printer file, the ODT messages that are written to the line printer can be directed to a backup medium. Most of the file attributes associated with the line printer file, the internal name of which is LINE, can be modified by the operator.

The SYSTEM/ODT program contains one disk file (internal name: ODTLOG; label SYSTEM/ODTLOG) that is used only when the MCP option ODTL is set. (Refer to section 3 of Volume 1 of the *B 1000 Systems System Software Operation Guide*.) This disk file is written by the SYSTEM/ODT program.

The SYSTEM/ODT program also contains a remote file declaration, for use with the remote ODT on B 1990 systems.

Modification of most of the attributes of any of the files, exclusive of the line printer file, is prohibited. If any of the attributes are set to an improper value (one that was not compiled into the program) the entire system can be rendered inoperable and a COLDSTART operation is necessary to effect recovery.

MEMORY REQUIREMENTS

The SYSTEM/ODT program is frozen in memory. The following memory requirements are approximate and can vary to a small extent.

Run Structure	10,200 bytes
Q.IN File	100 bytes
Q.OUT File	100 bytes
Segment Dictionary	35 bytes
Total	10,435 bytes

Under normal circumstances, only the files listed above are open. Operating with KB LP ON adds roughly 350 bytes to the memory requirements. Operating with the ODTL system option set adds approximately 350 bytes. Space requirements for input and output messages which are stored in the queue files mentioned also add to the memory requirements by small, varying amounts.

SECTION 46

SYSTEM/ODTLOGOUT

The SYSTEM/ODTLOGOUT program transfers, formats, and prints a log of ODT (operator display terminal) messages. The program works with the contents of the logfile and produces a formatted hard-copy version of ODTLOG.

The program provides the capability of printing ODT messages pertinent to a specific time frame without printing the entire ODT log. This option is invoked by setting program switch 1 equal to 1.

OPERATING INSTRUCTIONS

Execution of the SYSTEM/ODTLOGOUT program is performed using the ODT and takes one of several forms, depending on the task.

To transfer the log to disk, enter:

TL ODT

The MCPPII commands LG and LN cause the ODTLOG to be transferred, analyzed, and printed in its entirety. These commands are entered as follows:

LG ODT
LN ODT

To transfer, analyze, and print messages within a specified time frame, program switch 1 is set to 1:

LG ODT; SW1 1
LN ODT; SW1 1

To print a transferred log of ODT messages (for example, after a TL ODT command), enter the following command:

EX SYSTEM/ODTLOGOUT FILE LOGFILE NAME ODTLOG/<file number> DISK;

To print messages pertinent to a specific time frame (after the log has been transferred), enter the following command:

EX SYSTEM/ODTLOGOUT FILE LOGFILE NAME ODTLOG/<file number> DISK; SW1 1

SWITCH SETTINGS

There are two program switches that affect the operation of this program: Switch 0 and switch 1. If program switch 0 is set to an integer other than 0, the logfile is printed and then purged. If program switch 1 is set to 1, the program accepts user specifications for the date and time periods to be printed.

After the program is executed with program switch 1 set to 1, pertinent messages are displayed on the ODT. The response to the message

ENTER STARTING AND ENDING DATES (MM/DD/YY-MM/DD/YY)

would be in either of the following formats:

<mix-number> AX <month>/<day>/<year>

<mix-number> AX <month>/<day>/<year>-<month>/<day>/<year>

Examples:

<mix-number> AX 08/09/83

<mix-number> AX 08/08/83-08/09/83

Similarly, the response to the message

ENTER STARTING AND ENDING TIMES (HH:MM-HH:MM)

would be in the following format:

<mix-number> AX <hour>:<minutes>-<hour>:<minutes>

Example:

<mix-number> AX 15:02 - 16:30

ERROR MESSAGES

The SYSTEM/ODTLOGOUT program generates the following error messages:

INVALID LOG ENTRY FOUND

A log entry was encountered with a message that exceeded the normal limits.

ERROR - NO DATES SPECIFIED

NO OUTPUT PRINTED, THIS ODTLOG DOES NOT CONTAIN THE THE SPECIFIED DATE(S)
AND/OR TIME(S)

ERROR - NO TIME FRAME SPECIFIED

No dates or times or time frames were specified in the response to the request by the program for parameters.

I/O ERROR

An I/O error was encountered during a read of the LOGFILE.

INVALID MESSAGE SIZE

The ODTLOG entry was greater than 180 bytes.

INTERNAL FILES

The following files are used by the SYSTEM/ODTLOGOUT program:

Internal File Name	Function
LOGFILE	Stores transferred ODTLOG. Serves as input to program.
	LINE Printer file.

ODTLOG FILE LAYOUT

The ODTLOG file records consist of a 16-bit message header followed by the ODT message. Table 46-1 shows the ODTLOG layout of the 16-bit message header.

Table 46-1. ODTLOG File Layout

Field Name	Length (in bits)	Value	Description
MESSAGE-LENGTH	8	<length>	Contains the length of the ODT message in characters.
CONTINUATION	1	0	Message is not continued from previous sector.
		1	Message is continued from previous sector.
COMPLETE	1	0	Message is not complete.
		1	Message is complete.
SOURCE	4	0	Card reader input.
		1	ODT input.
		2	Queue input.
		3	Pseudo input.
		4	ZIP input.
		5	Task input.
		6	MCP-initiated input.
		7	Output message
		8	RJE message.
		9	BNA message.
		10	MCP debug message.
11	Enhanced display message.		
FILLER	2	--	--
TEXT	<length>	<character>	Text of ODT message.

SECTION 47

SYSTEM/PACKCOPY

SYSTEM/PACKCOPY is a normal-state utility program that is used to drive a B 9498 PE tape unit in the 100 ips mode for fast backup of user disk packs.

OVERVIEW

Two programs are actually involved in the copy process: SYSTEM/PACKCOPY for the disk I/O operations and communication with the operator, and SYSTEM/PACKCOPY-T for the tape I/O. The programs use extensive buffering and a direct I/O capability. Direct I/O eliminates the open and close operations normally associated with reading and writing files.

Synopsis of Program Operation

An operator (through AC or AX commands to SYSTEM/PACKCOPY) uses syntax similar to that of the SYSTEM/COPY program to specify the copy parameters. The result is a dump of all the files on the user disk to tape. The disk is copied in a file-by-file manner; therefore, no penalty is incurred in dumping packs with unused capacity. If the COPY command with the COMPARE option is selected, the SYSTEM/PACKCOPY program copies and compares the input disk files to the output tape files (including the tape directory file) in a reel-by-reel fashion. The program writes a reel of tape, rewinds the tape, compares that reel of tape to disk, and then proceeds in the same manner with additional tape reels until all files on the user disk are copied and compared. Because the output tape is written in the SYSTEM/COPY format, any or all of its files may be retrieved by means of the SYSTEM/COPY program.

Restrictions

The SYSTEM/PACKCOPY program is not a substitute for the SYSTEM/COPY program, and does not include all of the options or extensive capabilities of that program. The SYSTEM/PACKCOPY program is not designed for copying files from system disk, dumping files individually, copying multi-pack files, or loading tape to disk.

For effective use, the SYSTEM/PACKCOPY program should be the only job in a mix except for SYSTEM/ODT.

Other Considerations

The SYSTEM/PACKCOPY and SYSTEM/PACKCOPY-T programs use approximately 260K bytes of memory including approximately 218K bytes of dynamic memory for data buffers.

A pack that is being copied is not available through the MCP for any other purpose. If the SYSTEM/PACKCOPY program terminates abnormally, the pack must be readied before it can be accessed by the MCP. Abnormal termination of the SYSTEM/PACKCOPY program results in termination of the SYSTEM/PACKCOPY-T program.

OPERATING INSTRUCTIONS

The SYSTEM/PACKCOPY program communicates with the operator through the MCP ACCEPT mechanism, and accepts as input a proper subset of the syntax of the SYSTEM/COPY program.

Input must consist of one or more complete commands, each separated by a semicolon (;). Each command entered is completely parsed and executed before the next command in the input string is processed. For this reason, the <mix-number> AC <response> form of the response to an ACCEPT request must be used rather than the AX form. (Refer to the AC and AX system commands, *B 1000 Systems System Software Operation Guide, Volume 1*, section 5.)

Execution Examples

```
EX SYSTEM/PACKCOPY;AC COPY AND COMPARE = FROM MYPACK TO  
MYTAPE (KIND=TAPE);BYE
```

This entry causes the files on a user pack labeled MYPACK to be copied and compared to a tape named MYTAPE.

```
EX SYSTEM/PACKCOPY
```

```
SYSTEM/PACKCOPY =7771 BOJ. PP=4, MP=4 TIME = 16:07:26.0  
% SYSTEM/PACKCOPY =7771 ENTER THE NEXT COMMAND, PLEASE.  
SYSTEM/PACKCOPY =7771 ACCEPT.
```

```
7771AC COPY = FROM MYPACK (KIND = DISK) TO MYTAPE (KIND=TAPE)
```

```
SYSTEM/PACKCOPY-T =7772 BOJ. TIME = 16:07:57.8  
% SYSTEM/PACKCOPY =7771 BEGIN: COPY 134 FILES FROM MYPACK (SERIALNO=  
588301) TO MYTAPE (KIND=TAPE)  
SYSTEM/PACKCOPY-T =7772 EOJ. TIME = 16:13:46.5  
% SYSTEM/PACKCOPY =7771 END: COPY 134 FILES FROM MYPACK TO MYTAPE  
(KIND=TAPE)  
SYSTEM/PACKCOPY =7771 EOJ. TIME = 16:14:46.8
```

The previous sequence traces the copying of the files on a user pack labeled MYPACK to a tape labeled MYTAPE.

PR?

Syntax Errors

A syntax error encountered while parsing a command causes an error message to be displayed and any pending command input to be discarded. The operator is asked to enter the next command. A syntax error is reported with the following display:

```
-- -- -- > SYNTAX ERROR : <error message>
```

Example:

```
EX SYSTEM/PACKCOPY

SYSTEM/PACKCOPY =114 BOJ. PP=4, MP=4 TIME = 08:18:48.8
% SYSTEM/PACKCOPY =114 ENTER THE NEXT COMMAND, PLEASE
SYSTEM/PACKCOPY =114 ACCEPT.

114AC COPY AND COMPARE = FROM MYDISK TO MYTAPE (KIND=DISK)

% SYSTEM/PACKCOPY =114 ----> SYNTAX ERROR : "TO" <TAPE NAME>
"(KIND=TAPE)" REQUIRED
% SYSTEM/PACKCOPY =114 ENTER THE NEXT COMMAND, PLEASE
SYSTEM/PACKCOPY =114 ACCEPT.

114AC COPY AND COMPARE = FROM MYDISK TO MYTAPE (KIND=TAPE)

SYSTEM/PACKCOPY-T =115 BOJ. PP=5, MP=5 TIME = 08:19:07.6
% SYSTEM/PACKCOPY =114 BEGIN: COPY AND COMPARE 134 FILES FROM
MYDISK (SERIALNO=140012) TO MYTAPE (KIND=TAPE)
```

Execution Errors

If a problem that can be corrected by the operator is encountered (e.g., PACK CANNOT BE FOUND) during the execution of a command, the operator is requested to enter OK, NEXT or NEW.

OK is used after the problem is corrected. The program makes another attempt to execute the command.

NEXT causes the program to skip to the next command if more than one command was originally entered with the COPY command.

NEW, entered if the problem has not been corrected, causes all pending command input to be discarded. The program returns a request for new command entry.

Example:

```
114AC COPY = FROM MYDISK TO MYTAPE (KIND=TAPE)

% SYSTEM/PACKCOPY =114 PACK "MYDISK" CANNOT BE FOUND
% SYSTEM/PACKCOPY =114 ENTER "OK" IF PROBLEM FIXED, "NEXT" TO
PROCESS NEXT COMMAND IN INPUT STRING OR "NEW" TO ENTER
NEW COMMAND(S)

RY DPC

SYSTEM/PACKCOPY =114 ACCEPT.

114AC OK
```

I/O Errors

The MCP automatically tries to recover from tape errors. It tries up to 10 times to write a given block, moving the tape ahead after each try. If the data block cannot be written on tape after 10 tries, a write-parity-error-on-tape message is given. The name of the file that was being written at the time of the error is also displayed. The SYSTEM/PACKCOPY program then requests the operator to enter NEXT or EOJ.

NEXT causes the program to skip to the next command if more than one command was entered. If not, the program goes to EOJ. EOJ forces the program to end of job. There is no recovery of the current copy process from this error.

Example:

```
SYSTEM/PACKCOPY-T =7772 ENCOUNTERED A PARITY ERROR ON MTA ON WRITE
% SYSTEM/PACKCOPY-T =7772 ** FATAL TAPE ERROR (DATA ERROR) ,
  RESULT = @D00080@
% SYSTEM/PACKCOPY-T =7772 CURRENT FILE IS FAMILYNAME/FILENAME
% SYSTEM/PACKCOPY =7771 ENTER "NEXT" TO PROCESS NEXT COMMAND IN
  INPUT STRING OR "EOJ"
SYSTEM/PACKCOPY =7771 ACCEPT.

7771AX EOJ
% SYSTEM/PACKCOPY =7771 CURRENT COPY TERMINATED
SYSTEM/PACKCOPY-T =7772 EOJ. TIME = 14:43:25.1
% SYSTEM/PACKCOPY =7771 END: COPY 23 OF 134 FILES FROM SB TO
  SBT(KIND=TAPE)
SYSTEM/PACKCOPY =7771 EOJ. TIME = 14:43:27.0
```

The MCP automatically tries to recover from disk errors also. If the disk error is irrecoverable, however, the SYSTEM/PACKCOPY program requests entry of CONTINUE, QUIT, NEXT, or EOJ.

If the error occurs while reading user data, entering CONTINUE instructs the program to ignore it. The current data block is marked as unreliable and copying continues with the writing of the remaining blocks of the file to tape.

If the error occurs while reading a disk directory sector entry, CONTINUE instructs the program to skip to the next sector entry. Copying continues with the next available file in the next directory sector.

A QUIT response is requested only for errors in user data. QUIT instructs the program to abandon the file. A file-abort indication is written to tape and copying continues with the next file in the directory.

A NEXT response instructs the program to skip to the next command if more than one command was entered with the current COPY command.

An EOJ response instructs the program to go to end of job.

Example:

```
% SYSTEM/PACKCOPY =125 IRRECOVERABLE DISK READ ERROR  
ENCOUNTERED ON PACK "MYPACK"  
% SYSTEM/PACKCOPY =125 ERROR IN FILE FAMILYNAME/FILENAME ON  
MYPACK, SECTORS @01FADE@ THROUGH @01FAFB@  
% SYSTEM/PACKCOPY =125 ENTER "CONTINUE" TO IGNORE THE ERROR,  
"QUIT" TO ABANDON THE CURRENT FILE, "NEXT" TO PROCESS NEXT  
COMMAND IN INPUT STRING OR "EOJ"  
SYSTEM/PACKCOPY =125 ACCEPT.
```

125AX CONTINUE

```
SYSTEM/PACKCOPY-T =126 EOJ. TIME 23:55:34.7  
% SYSTEM/PACKCOPY =125 END: COPY 10 FILES FROM MYPACK TO  
MYTAPE (KIND=TAPE)  
SYSTEM/PACKCOPY =125 EOJ. TIME 23:55:40.2
```

Comparison Errors

When the COMPARE option is specified, SYSTEM/PACKCOPY compares the input disk files to the output tape files in a reel-by-reel manner. In the event of a comparison error, a listing is produced indicating the name and hardware type of both the source (FROM) and copy (TO) volumes, the name of the file currently being compared, the type of tape records being compared, and the contents of those records. The name of the file and the type of tape record is also displayed.

Comparison errors occurring in user-file data records or tape directory entry records are not fatal errors and cause the program to request the operator to enter the SKIP, NEXT, or EOJ command. Comparison errors occurring in more critical kinds of records, such as tape header records or tape end-of-file records, are fatal and the operator is only given a choice of the NEXT or EOJ command.

The SKIP command, when entered in response to a user-file data comparison error, stops the compare process until the end of the current tape file is reached, or in the case of a file that crosses reel boundaries, the end of the reel. The compare process resumes with the start of the next file, or in the case of a file that crosses reel boundaries, the start of the next reel. If the comparison error occurred in a tape directory entry record, the SKIP command instructs the program to continue the compare process with the next entry.

The NEXT command causes the program to skip to the next command if more than one was originally entered.

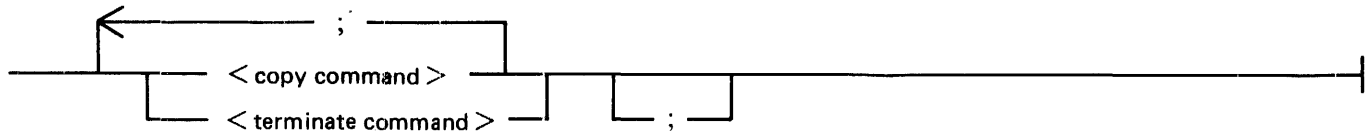
The EOJ command causes the program to go to end of job (EOJ).

Example:

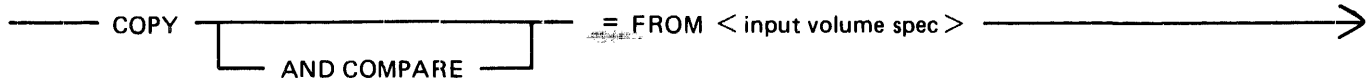
```
% SYSTEM/PACKCOPY-T =298 ** COMPARISON ERROR IN FILE  
FAMILYNAME/FILENAME  
% SYSTEM/PACKCOPY =297 ENTER "SKIP" TO SKIP THE CURRENT FILE,  
"NEXT" TO PROCESS NEXT COMMAND IN INPUT STRING OR "EOJ"  
SYSTEM/PACKCOPY =297 ACCEPT.
```

297AX SKIP

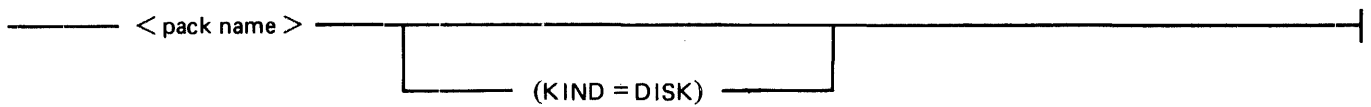
Syntax:



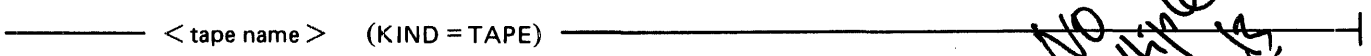
< copy command >



< input volume spec >



< output volume spec >



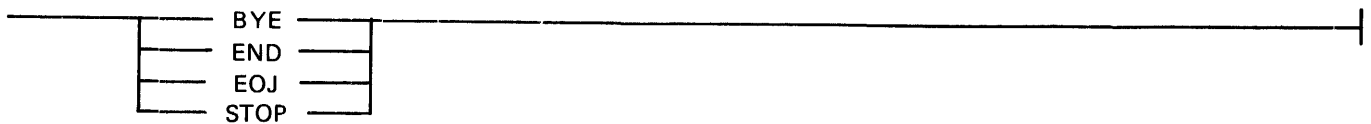
< pack name > is any valid MCP pack name of 10 or fewer characters.

< tape name > is composed of 10 or fewer characters, consisting of:

- A-Z (upper-case letters),
- 0-9 (numbers),
- (hyphen),
- _ (underscore).

No Multiple outputs

< terminate command >



Note that the SYSTEM/PACKCOPY program automatically goes to end-of-job (EOJ) after all commands have been processed. This means that under normal circumstances a command to terminate is not required. However, if program switch 1 = 1, the program stays in the mix and waits for additional copy commands or for a command to terminate.

SPECIAL OPERATIONAL CONSIDERATIONS

The SYSTEM/PACKCOPY program utilizes the Burroughs B 9498 Streamer PE Tape Unit in the 100-ips mode, but it also works well with other PE tape drives. The B 9498 unit reads and writes 1600-bpi, phase-encoded tape at either 25 ips or 100 ips (speed selectable at the interface). The SYSTEM/PACKCOPY program operates the B 9498 drive at 100 ips only.

Because the drive is designed for streaming, stop/start action adds throughput overhead -- positioning times are much longer than is the case with a standard stop/start tape drive. This overhead is much greater at 100 ips than at 25 ips, as shown in the following table:

	B 9498 25 ips	B 9498 100 ips
Reinstruct time	16 ms	4 ms
Burden for stop/start	170 ms	910 ms

If a read or write command is not issued within the window specified by the reinstruct time, the B 9498 stops and then starts whenever the read or write instruction is received.

The B 9495, a stop-in-the-gap tape drive, has several speeds ranging from 25 ips to 125 ips. The following table shows, for each speed, the stop and start times and the resulting throughput burdens. The burdens are compared, where appropriate, with those of the B 9498 streamer unit.

	B 9495 Stop Time	B 9495 Start Time	B 9495 Burden	B 9498 Burden
25 ips	27.0 ms	16.0 ms	43 ms	170 ms
50 ips	6.5 ms	6.5 ms	13 ms	--
75 ips	5.0 ms	5.0 ms	10 ms	--
100 ips	---	---	--	910 ms
125 ips	3.5 ms	3.5 ms	7 ms	---

B 9498 performance depends on many variables, some of which are given below. Except under ideal conditions, the performance of the B 9498 is better at 25 ips than at 100 ips.

Following are the variables that influence the performance of the B 9498 drive:

Type of B 1000:
6 MHz or 4 MHz
Cache
Control Memory
No control memory and no cache

Type of Disk:
Cartridge
225
206
207

Operating Environment:

- Number and priorities of jobs in mix
- Record, block and file size
- Number of Buffers
- Number of tapes
- Program driving the B 9498
- General system overhead

Ideal conditions:

- B 1955/1985
- 512K or more bytes of main storage
- One 206 user disk pack
- One B 9498 tape unit
- One B 1491-91 (MTC-6) controller (required for B 9498)
- SYSTEM/PACKCOPY the only job in the mix

The SYSTEM/PACKCOPY program minimizes the stop/start burden with a large (approximately 218K bytes) data buffer, as well as by having the program handle much of the I/O process itself, and restricting the mix to the SYSTEM/PACKCOPY program only. Changes in running conditions, additional jobs in the mix or a smaller buffer size, for example, drastically affect throughput.

SECTION 48

SYSTEM/PANDA

The SYSTEM/PANDA program (PANDA stands for "Package of Algorithms for Normal-state Directory Analysis") accesses the disk directory and disk management tables on a specified on-line disk to produce various tabulations and printed reports about disk usage. Specific tabulations or reports may be selected by entering one or more keywords to the SYSTEM/PANDA program, as is described in the Operating Instructions subsection that follows the Options Keyword Glossary.

REQUIRED SYSTEM FILE

The SYSTEM/PANDA program is a required system program and it must reside on the system disk. The mnemonic PAN has been added as a Name Table entry to identify the SYSTEM/PANDA program. All MCP commands that implicitly invoke the SYSTEM/PANDA program use the PAN Name Table entry.

When a user disk becomes ready, or is ready during a Clear/Start operation, and the disk integrity is in doubt, the current SYSTEM/PANDA program is executed to check the disk integrity. After the disk integrity is checked, the SYSTEM/PANDA program absolutizes the pack if necessary, removes temporary files, and generates correct end of file pointers for protected files.

If the SYSTEM/PANDA program finds a fatal error, the SYSTEM/PANDA program waits for an accept from the operator. The operator can override this warning, but it is strongly recommended that the pack not be updated and all the files be copied to a safe medium.

The MCP still cleans up the system disk directory after a Clear/Start operation, but the SYSTEM/PANDA program is initiated to check the disk directory and handle protected files.

OPTIONS KEYWORD GLOSSARY

The results produced by use of the following keywords are for the disk specified after invoking program execution, as described in the Operating Instructions subsection.

ALPHA

Tabulates disk file usage by file name to produce a list similar to the listing produced by the KA system command. The file identifiers are sorted into ascending alphabetical order before printing.

CHECK

Collects and sorts by ascending address all areas ("chunks") referenced by all tables and by the disk directory. The set of address/length pairs that results should represent a map of the range of disk addresses from address zero to the maximum disk address. A list is produced of all the address/length chunks that either are unreferenced by any table (missing disk areas) or represent areas referenced by more than one table (overlapping disk areas).

DEBUG

This option causes the SYSTEM/PANDA program to identify and print (hexadecimal format) all chunks of disk it accesses while producing the specified reports. Burroughs technical personnel can use this information to analyze corrupted disks and, possibly, to recover them.

DECK

Generates a card punch file (by default) that contains the file identifiers of all files on the selected disk, one per card.

DSKAVL

Produces a list of all available areas on the disk. The areas are listed in order of descending size (in segments).

KA

Combines the DSKAVL and ALPHA options to produce a list of the available chunks of disk and a tabulation of disk file usage.

MAP

Collects and sorts (as in the CHECK option) the address/length pairs, producing a list of all the pairs. Thus, the current usage of any sector on the disk can be identified.

PD

Creates a compact list of the file identifiers of all files on the disk.

SAVE

Places the data tabulated by use of the ALPHA option (the disk file header information) into a disk file in a format accessible by user language programs.

UNSORTED

Suppresses the sorting for the DSKAVL, PD, ALPHA, DECK, SAVE, and KA options so that the files that result are in the sequence in which the SYSTEM/PANDA program encountered the appropriate information.

OPERATING INSTRUCTIONS

The SYSTEM/PANDA program may be initiated by the MCP in response to operator entry of certain system commands (KA, PD), but the discussion here is limited to explicit execution by a normal EXECUTE statement.

At BOJ, if there are no queued-up input messages, the program displays a prompt asking for the name or unit mnemonic of the desired disk and the options (keywords). The operator responds by means of a single free-form AX command with the input message.

The following considerations apply to disk selection:

- If the system has more than one system disk drive, specifying any one of them causes all to be processed.
- The keyword DISK or SYSTEM may be used instead of a unit mnemonic to represent the system disk(s).
- If the first token within the accepted input is neither a recognized unit mnemonic nor the word DISK or SYSTEM, it is taken to be the name of a user disk and the program searches for a device with that name.

A typical execute sequence is illustrated below:

EX SYSTEM/PANDA

SYSTEM/PANDA = 7071 BOJ...

% SYSTEM/PANDA = 7071 ENTER UNIT MNEMONIC OR PACK-ID AND OPTIONS

7071AX SYSTEM CHECK PD

PROGRAM SWITCHES

Program switches 0 through 6 may be used to select program options in place of entries made with AC or AX commands.

Program Switch 0

With program switch 0 = 1, the system disk is automatically selected for the requested operations.

With program switch 0 = 2, the disk to be processed is specified as the pack-id of the internal file DUMMY.

In either case, the "... <unit-mnemonic> OR PACK-ID ..." portion of the ACCEPT message is suppressed.

Program Switches 1-5

These switches indicate the program options. Setting any of them to a non-zero value causes the option portion of the ACCEPT message to be suppressed. Thus, if both program switch 0 and one or more of program switches 1 through 5 are set to a non-zero value, prompts requesting AX inputs are not provided and unsolicited inputs are not processed by the SYSTEM/PANDA program.

The effects of setting program switches 1-5 are detailed in table 48-1.

Table 48-1. SYSTEM/PANDA Program Switches 1-5

SWITCH	VALUE	OPTIONS SET
1	1	CHECK
	2-15	MAP, CHECK
2	1	ALPHA
	2	DSKAVL
	3-15	ALPHA, DSKAVL
3	1-15	SAVE
4	1-15	DECK
5	ODD	PD
	Bit 2 set	KA <family>/=, using the multi-file-ID of file DUMMY. (Bit 0 is the leftmost bit of the 4-bit switch.)

Program Switch 6

To ensure integrity of the information collected, the SYSTEM/PANDA program requires a user disk to have no users while being processed. If the program is asked to analyze a disk that is in use, a warning message is displayed and the program goes into an idle state, periodically checking the user count. When the count goes to zero, processing begins.

However, if program switch 6 is set to a non-zero value, there is no wait for a zero user count. A warning message is displayed if there are any users.

A system disk can never have zero users; therefore, this switch has no effect on the use of the program to analyze the system disk.

Note that a SYSTEM/PANDA operation on any disk with users, be it a system disk or a user disk, can result in analysis errors because of the constant manipulation of the numerous disk tables by the MCP. False indications of overlapping areas are possible because of this.

PROGRAM EXECUTION

The SYSTEM/PANDA program operates in two phases. In the first phase, the various directories and tables on the disk are accessed and appropriate information is collected for later processing. Unless program switch 6, discussed above, is used, a user disk is reserved for the program and made otherwise unavailable to the system during this phase. With a system disk, however, such security is not possible, and any disturbances during the first phase can lead to erroneous or incomplete analyses because these disturbances cannot be programatically detected by the program. Certain errors are detected and listed during this phase; they are described next, in MAP and CHECK Options.

In the second phase, the information collected during the first phase is sorted, if appropriate, and the requested listings and tabulations are produced.

ERROR AND ADVISORY MESSAGES

Two types of error messages may be returned: (1) messages related to disk integrity, resulting from use of the MAP or CHECK options, and (2) error or advisory messages unrelated to disk integrity.

MAP and CHECK Related Messages

The listings produced by the MAP option and the CHECK option each contain one line for each chunk of disk space. This line shows the starting and ending addresses, the size in sectors, a description of the chunk, and, for files, the file identifier and area number. Missing areas and overlapping areas are listed separately, and other errors detected in the first phase are presented in a tabulation that precedes the main MAP or CHECK listing.

Possible errors are listed and defined in the paragraphs that follow. For any error except MISSING DISK AREA, processing should be stopped and a STANDALONE/DISK-DUMP performed. Processing may continue in the case of a MISSING DISK AREA error.

BAD AREA ADDRESS

An area address within a file header pointed to an address not on disk (or set of system disks).

BAD BACK FIELD or BAD FORWARD FIELD

Most disk structures are represented by doubly-linked lists, with each segment linked forward and backward to the adjacent segments in the list. This error message is produced when some conflict in the list is detected and the forward or backward link of one segment does not correspond to the self address of the segment pointed to by the link.

BAD DFH__AREA__ADDR__OFFSET

A file header was found in which this field, which describes the displacement of the list of area addresses within the header, was not within the allowed range of 540-580, inclusive.

BAD DFH__AREA__CTR

A file header was found whose count of areas in use exceeded the number of areas requested.

BAD DFH AREAS RQST

A file header describing a file of either zero or more than 105 possible areas was detected.

BAD DISK ADDRESS

A disk address was found to have a port, channel, or unit other than that of the disk being processed (or, in case of multiple system disks, not in the set of valid PCU's). The address is not read.

BAD HEADER SIZE

A multi-pack file was found whose header size (in the MPF information table) was either less than the minimum possible size of 540 bits or greater than the maximum size of 4320 bits.

BAD SELF FIELD

To help ensure disk integrity, many MCP disk structures (directories and tables) include self pointers. A self pointer represents the segment in which the field itself is located. The SYSTEM/PANDA program displays the BAD SELF FIELD diagnostic message when the self field of a sector does not correspond to the sector from which the information was read.

ILLEGAL ALL-ZERO AVAILABLE TABLE ENTRIES

The available tables, which describe unassigned chunks of disk available for MCP use or for the MCP to give to user programs, normally consist of address/length pairs of non-zero size. This diagnostic is displayed if the SYSTEM/PANDA program detects one or more address/length pairs with a length of zero segments.

MAX ADDRESS EXCEEDED

A chunk of disk was found to include an address or addresses greater than the size of the disk as determined from the Master Available Table.

MISSING DISK AREA

The specified chunk of disk is unreferenced by any MCP directory or table and, consequently, is unavailable for use.

MULTI-PACK AREA ADDRESS ON SYSTEM DISK

The header for a multi-pack file indicated that an area for the file resided on a system disk; this is illegal.

OVERLAPPING AREAS

The specified chunk of disk is described by more than one disk directory entry, disk file header, or table. Where possible, the corrupted areas are identified.

<number> ERRORS DETECTED DURING DATA COLLECTION PHASE

If errors occur during the first phase of execution of the SYSTEM/PANDA program, analyses performed during the second phase will be incomplete and may indicate severe corruption.

Other Advisory and Error Messages

A number of advisory or error messages not related to disk integrity are provided by the SYSTEM/PANDA program.

ACTIVITY MUST CEASE ON <unit-mnemonic> PRIOR TO PHASE

1...SYSTEM/PANDA IDLE AND WAITING

The user disk on which analysis was requested has a non-zero user count. The program waits for the user count to reach zero before beginning analysis.

<unit-mnemonic> TAKEN BY SYSTEM/PANDA...PROCESSING BEGINS

After successfully waiting for the user count of a disk to drop to zero, the SYSTEM/PANDA program has reserved the device and is commencing processing.

<unit-mnemonic> IN USE DURING MAP

Analysis is being performed on a user disk despite a non-zero user count because of the setting of program switch 6.

BAD FIELD HINTS.SYSTEM__UNIT...EOJ

The system disk PCU (port, channel, and unit) contained within the MCP's table of system disks was not found in the IOAT (Input/Output Assignment Table). SYSTEM/PANDA goes to end of job.

BAD REFERENCE ADDRESS

The I/O subsystem detected a reference address error in one of the disk reads performed by SYSTEM/PANDA. Unlike other error messages in this section, this does not represent a corruption of disk. Rather, a hardware malfunction would be initially suspected. SYSTEM/PANDA goes to end of job.

ENTER UNIT MNEMONIC OR PACK-ID AND OPTIONS

ENTER UNIT MNEMONIC OR PACK-ID

ENTER OPTIONS

If the program switches are appropriately set, one of these messages prompts the operator for AX input.

FATAL ERROR: CORRUPTED CSV.SYSTEM__UNITS FIELD

The port, channel, and unit of each system disk is maintained in a list within the coldstart variables. One such non-zero entry was found to have a port not equal to seven (representing soft I/O). The program goes to end of job.

SOFTWARE INCOMPATIBILITY

The software release under which SYSTEM/PANDA was run did not match the release under which it was intended to be run. Both values are identified and the program goes to end of job.

SORRY, NO TUTELAGE AVAILABLE...EOJ

This is the program's response to the entry of a HELP or TEACH request.

UNIT NOT FOUND...EOJ

The disk upon which analysis was requested could not be found or was not on line. The program goes to end of job.

UNIT <unit-mnemonic> LABELED <pack-id> IS AN INTERCHANGE PACK...CANNOT PROCESS

A Burroughs interchange pack cannot be analyzed by SYSTEM/PANDA. The program goes to end of job.

UNRECOGNIZED TOKEN: <token>; PLEASE RESUBMIT INPUT

An illegal or misspelled option was input to the SYSTEM/PANDA program. A new AX response with correct information is needed.

NOTES ON SAVE OPTION FILE

The file produced by the SAVE option has the following attributes:

internal file name:	"USER"
external file name:	"USER"
record size:	180 bytes
records per block:	5
areas:	25
blocks per area:	100

The information for each file on the disk occupies from one to four records in this file, depending upon the number of disk areas in use by the file. The record format (in COBOL notation) is as follows:

```
01 DATA-RECORD
  02 FILE-IDENT PIC X(20).
  02 RECORD-COUNT PIC 9 COMP.
  02 HEADER-DISK-ADDRESS PIC 9(12) C OMP.
  02 RECORD-SIZE PIC 9(04) COMP.
  02 RECORDS-PER-BLOCK PIC 9(04) COMP.
  02 BLOCKS-PER-AREA PIC 9(06) COMP.
  02 SEGMENTS-PER-AREA PIC 9(06) COMP.
  02 AREAS-DECLARED PIC 999 COMP.
  02 AREAS-IN-USE PIC 999 COMP.
  02 EOF-POINTER PIC 9(07) COMP.
  02 FILE-TYPE PIC 999 COMP.
  02 CREATION-DATE PIC 9(05) COMP.
  02 LAST-ACCESS-DATE PIC 9(05) COMP.
  02 SAVE-FACTOR PIC 9(04) COMP.
  02 CREATE-TIME PIC 9(07) COMP.
  02 UPDATE-DATE PIC 9(05) COMP.
01 AREA-ADDRESSES REDEFINES DATA-RECORD.
  02 AREA-ADDRESS OCCURS 30 PIC 9(12) COMP.
```

The RECORD-COUNT field contains the number of records that are required for the file description. For example, if a file has less than 16 areas in use (AREAS-IN-USE), it requires only one record; for 16-45 areas in use, two records are required, and so forth.

The HEADER-DISK-ADDRESS is stored in the following format:

01	DISK-ADDRESS-FORMAT.		
02	SERIAL-NUMBER-FLAG	PIC 9	COMP.
02	PORT-NUMBER	PIC 9	COMP.
02	CHANNEL-NUMBER	PIC 99	COMP.
02	UNIT-NUMBER	PIC 99	COMP.
02	DISK-ADDRESS	PIC 9(06)	COMP.

If the SERIAL-NUMBER-FLAG is 1, PORT-NUMBER, CHANNEL-NUMBER, and UNIT-NUMBER are not used; DISK-ADDRESS contains a disk cartridge or disk pack serial number (for multi-pack files). If the SERIAL-NUMBER-FLAG is 0, the other fields contain decimal values describing the disk address.

The entire AREA-ADDRESS field is zero for an unallocated area.

Using the above format, the AREA-ADDRESSES are stored 30 to each 180-byte record except for the first 15 AREA-ADDRESSES, which are stored in the second half of the first record, as shown below.

AREA NUMBER	RECORD NUMBER	SUBSCRIPTS
1 - 15	1	16 - 30
15 - 45	2	1 - 30
46 - 75	3	1 - 30
76 - 105	4	1 - 30

Thus, the address for area number 3 is located in AREA-ADDRESS(18) of record #1, the address for area number 30 is located in AREA-ADDRESS(15) of record #2, and so forth.

Date codes, CREATION-DATE, LAST-ACCESS-DATE, UPDATE-DATE, all PIC 9(05), are in Julian date format: YYDDD (last two year digits, 000-366 day digits). CREATE-TIME is HHMMSSST (hours-minutes-seconds-tenths).

FILE-TYPE codes are listed in table 48-2.

Table 48-2. FILE-TYPE Codes

SYSTEM FILES	CODE	CODE FILES	CODE
LOG_FILE	001	CODE_FILE	008
DIRECTORY_FILE	002	SDLCODE	110
PSR_DECK	003	COBOL68CODE	111
PRT_FILE	004	RPGCODE	112
PCH_FILE	005	NDLCODE	113
DUMP_TYPE_FILE	006	FORTRANCODE	114
INACTIVE_LOG_FILE	010	MILCODE	115
INTRIN_FILE	012	BASICCODE	116
USERCODE_FILE	016	UPLCODE	117
MCP_TEMPORARY FILE	023	COBOL74CODE	118
JOBLOG_FILE	024	FORTRAN77CODE	119
		IBASICCODE	120
		PASCALCODE	121
		SMPCODE	123
		MCSCCODE	124
		NON_NATIVECODE	125
		B500CODE	126
		IBM1400CODE	127
		SORTCODE	128
		SDL2UNBOUNDCODE	129
		JOBCODE	130
		MICROCODE FILES	CODE
		INTERPRETER_FILE	007
		SDLINTERPRETER	140
		COBOL68INTERPRETER	141
		RPGINTERPRETER	142
		NDLMICROCODE	143
		FORTRANINTERPRETER	144
		BASICINTERPRETER	145
		UPLINTERPRETER	146
		COBOL74INTERPRETER	147
		FORTRAN77INTERPRETER	148
		IBASICINTERPRETER	149
		PASCALINTERPRETER	150
		SDL2INTERPRETER	151
		MICROMCP	152
		GISMO	153
		SYSTEM_INITIALIZER	154
		B500INTERPRETER	155
		IBM1400INTERPRETER	156
		DATA FILES	CODE
UNSPECIFIED_DATA_FILE	000		
DATA_FILE	009		
VARIABLE_LENGTH_FILE	011		
DMS_DATA_FILE	013		
DMS_DICTIONARY	014		
DMS_AUDIT_FILE	015		
RELATIVE	017		
INDEX_SEQ_GLOBAL_FILE	018		
INDEX_SEQ_DATA_SET_FILE	019		
INDEX_SEQ_INDEX_FILE	020		
INDEXED_DATA_FILE	022		
SDLSYMBOL	060		
COBOL68SYMBOL	061		
RPGSYMBOL	062		
NDLSYMBOL	063		
FORTRANSYMBOL	064		
MILSYMBOL	065		
BASICSYMBOL	066		
UPLSYMBOL	067		
JOBOL74SYMBOL	068		
FORTRAN77SYMBOL	069		
IBASICSYMBOL	070		
DASDYLSYMBOL	071		
PASCALSYMBOL	072		
SDL2SYMBOL	073		
NON_NATIVE DATA	074		
IBASIC_INTERNAL	075		
FORTRAN77_UNFORMATTED	076		
COBDMS_LIB	077		
RPGDMS_LIB	078		
NETWORK_INFORMATION	079		
NDL_LIBRARY	080		
PASCAL_INTERCHANGE_DATA	081		
PASCAL_MODULE-DATA	082		
SORTSYMBOL	083		
SEQDATA	084		
JOBSYMBOL	085		

USER FILE HEADING

The following record, in COBOL notation, is the user file record for the header record.

```
01  USER FILE RECORD
    02  PACK-NAME                PIC X (10).
    02  PACK-SERIAL-NO          PIC X (06).
    02  PACK-HDWR               PIC X (01).
    02  PACK-BPS-NO             PIC X (06).
    02  PACK-CONTINUATION       PIC X (01).
    02  PACK-NO-OF-AVL-AREAS    PIC X (05).
    02  PACK-LARGEST-AVL        PIC X (07).
    02  PACK-TOTAL-AVL-SEGS     PIC X (07).
```

SECTION 49

SYSTEM/SQUASH

The SYSTEM/SQUASH program is a normal-state utility program that consolidates available storage areas on a user disk. In performing this consolidation, the squash algorithm moves data into holes in the available disk space, and then frees the previously used data space. When disk space is extremely fragmented, successive squashes can further consolidate the available area. The SYSTEM/SQUASH program uses an external sort program called SORT/VSORT. The SORT/VSORT program displays the number of records sorted. This number consists of all disk pieces, which can or cannot be moved.

The SYSTEM/SQUASH program can run while other programs are in the mix; however, no other program can access the disk that is being squashed. The disk being squashed is unavailable to the system when the squash operation begins. The SYSTEM/SQUASH program waits until there is no activity on the disk before beginning the squash.

OPERATING INSTRUCTIONS

The SYSTEM/SQUASH program can be initiated with an explicit EXECUTE command, or it can be initiated implicitly with the SQ system command. (Refer to SQ in section 5 of Volume 1 of the *B 1000 Systems System Software Operation Guide* for further details). The explicit execution of SYSTEM/SQUASH is described in the following example:

```
EXECUTE SYSTEM/SQUASH
SYSTEM/SQUASH = 1111 BOJ.
% SYSTEM/SQUASH = 1111 ENTER UNIT TO BE SQUASHED:
                                DP? OR DC?
SYSTEM/SQUASH = 1111 ACCEPT.
1111 AX DPE
```

No further input is required. The program displays appropriate messages at the beginning of the job and when the job cannot be started due to programs still running on the designated disk drive. The following messages are displayed.

DO NOT DISTURB <unit-mnemonic> UNTIL FURTHER NOTICE
Displayed at the beginning of the job.

ACTIVITY MUST CEASE ON <unit-mnemonic>FOR DISK SQUASH TO CONTINUE
Displayed if specified pack is in use.

As soon as the specified unit becomes available (when all activity on the device stops) the SYSTEM/SQUASH program performs an integrity check on the disk and then squashes the disk.

When the squash operation is complete, the SYSTEM/SQUASH program displays the following messages:

```
% SYSTEM/SQUASH =1111 MAXIMUM CONTIGUOUS SPACE
                                AVAILABLE IS <integer> SECTORS
% SYSTEM/SQUASH =1111 TOTAL SPACE AVAILABLE IS
                                <integer> SECTORS
% SYSTEM/SQUASH =1111 SQUASH COMPLETE
% SYSTEM/SQUASH =1111 <unit-mnemonic> LABELED <pack-id>
                                IS NOW AVAILABLE FOR USE
SYSTEM/SQUASH =1111 EOJ.
```

PROGRAM SWITCHES

The following paragraphs describe the program switches available for use with the SYSTEM/SQUASH program.

Switch 1

Setting program switch 1 to 0 causes the SYSTEM/SQUASH program to generate a map list of the disk before squashing the disk. Setting program switch 1 to a non-zero value does not.

Switch 2

Setting program switch 2 to 0 causes the SYSTEM/SQUASH program to generate a map list of the disk after squashing the disk. Setting program switch 2 to a non-zero value does not.

INTEGRITY CHECK

The SYSTEM/SQUASH program checks the disk for overlapping areas before beginning the squash operation. If any overlapping areas are found the SYSTEM/SQUASH program describes the size and location of each overlapping area in an ODT message and then aborts the squash operation.

Examples of overlapping disk area messages:

OVERLAPPING SECTORS ON UNIT DPB LABELLED DOCGRP

1 SECTOR OVERLAPPING STARTING AT SECTOR @000ABB@
1 SECTOR OVERLAPPING STARTING AT SECTOR @000ABD
128 SECTORS OVERLAPPING STARTING AT SECTOR @000D15@

Overlapping areas cause the program to abort before performing the squash operation. The following message is displayed when this error occurs:

<integer> OVERLAPPING AREAS -- SQUASH ABORTED

The SYSTEM/SQUASH program also checks the available table for out-of-sequence or zero-length entries. These are flagged on the first KA listing (generated by the SYSTEM/SQUASH program), and the program aborts before performing the squash operation with the following message:

BAD AVAILABLE TABLE -- SQUASH ABORTED

RUN-TIME OPTIONS

While the squash operation is in progress, the operator can set several options by entering unsolicited AX or AC commands, as follows:

<mix-number> AX <option>

Descriptions of valid options and the actions they request follow.

SIZE

The program displays the current address, the maximum contiguous disk space available, and the total disk space available. The format of these messages is identical to the space available messages displayed just prior to end of job (EOJ). The values displayed reflect the contiguous and total space available at the time the SIZE interrogation is received.

STOP

The program terminates as soon as possible without corrupting data. The SYSTEM/SQUASH program must never be discontinued with a DS or DP system command; the STOP run-time option must be used instead.

NEED <n>

The program terminates as soon as the <n> contiguous sectors are made available. This option does not ensure that <n> contiguous sectors are made available; if they are made available, however, the program terminates without consolidating (squashing) any more of the disk.

TRACE <n>

The program generates trace output on a line printer listing. The value of <n> must be either 0, 1, 2, or 3. Refer to the following subsection entitled Trace Functions for further details.

TRACE FUNCTIONS

The TRACE option allows the operator to follow the progress of the squash. One of three different types of trace functions can be requested. The value <n> entered with the TRACE keyword selects the trace function.

0

Nothing is traced. This is the default value of trace. Entering TRACE 0 turns off previously requested trace options.

1

Each piece of data is traced when it is moved. Example of TRACE 1 output:

```
AVAILABLE TABLE MOVED FROM @00005F@ TO @000042@
FILE HEADER MOVED FROM @00007B@ TO 00004B@
SUB-DIRECTORY MOVED FROM @00008A@ TO @00004F@
FILE AREA MOVED FROM @00009A@ TO @000053@
```

2

Each disk I/O (read or write) operation is traced. Example of TRACE 2 output:

```
READ @0000EC@
WRITE @00006F@
READ @00006E@
WRITE @00006E@
```

3

Each disk I/O (read or write) operation is traced, and the contents of the read/write buffer are listed on the trace output. Example of TRACE 3 output:

```
READ      @000030@ F21000031000000000 F210000300F21 00007800008F
WRITE     @000030@ F21000031000000 000F210000300F 2100007 900008E..
```

PROGRAM OUTPUT

The SYSTEM/SQUASH program generates a summary printer listing and a squashed input disk.

Summary Listing

The SYSTEM/SQUASH program generates a line printer listing that describes the disk before and after the squash is performed. This summary listing is similar to two KA listings, with additional information describing the available disk areas printed at the beginning of the listing.

TRACE information, if requested, is included in this listing. Also, out-of-sequence or zero-length available table entries are flagged on this listing if the SYSTEM/SQUASH program discovers table errors during its integrity check.

If no line printer is available, a backup print file containing the information is created.

Squashed Disk

After the squash operation is complete, the user disk contains the same data that it did prior to the squash, although the data can be in a physically different location on the disk. The available tables are updated during the squash, with the available areas consolidated whenever possible.

ERROR RECOVERY

An abnormal termination of the SYSTEM/SQUASH program due to a CLEAR/START operation, a DS or DP system command, or a program failure, might corrupt the disk being squashed. How much information, if any, is lost depends on the state of the program at the time it was aborted.

The SYSTEM/PANDA program must be run to determine what errors, if any, there are. Using the SYSTEM/PANDA output, along with a KA listing of the disk before and after the squash operation, the operator can determine what was being moved at the time of failure. Recovery of lost data may be attempted by removing any bad areas and using IAD (Installation Allocated Disk) files to recover lost information. The order of the squash operation follows.

1. The data is moved.
2. Relatives are notified of the move (for example, the area-address field of a disk file header must be updated when one of its areas is moved).
3. The old space is returned to the disk available table.
4. The new space is removed from the disk available table.

ERROR MESSAGES

The following paragraphs describe the error messages.

Nonfatal errors

Nonfatal errors generate a message that describes the error. When the error exists in an input specification string, the SYSTEM/SQUASH program requests a response and waits for an AC or AX command input. The operator must correct the error and re-enter the specifications, or terminate the program.

<unit> IS AN INVALID UNIT NAME

Valid unit names are DP<letter or number> and DC<letter>. These mnemonics reference the disk drive on which the disk to be squashed is located. DP<letter or number> indicates a disk pack device; DC<letter> indicates a disk cartridge device.

<unit> COULD NOT BE ASSIGNED

The unit requested could not be assigned to the program. Make sure that the unit is on-line and that there is no activity on the disk. While a disk is being squashed, no program can access data on the disk.

MUST USE MCP7S "SQ" FOR SYSTEM PACK

The SYSTEM/SQUASH program can only squash user disks when it is started explicitly with an EXECUTE command. Refer to the SQ system command in section 5 of Volume 1 of the *B 1000 Systems System Software Operation Guide*.

<option> IS NOT A KNOWN OPTION

An invalid run-time option was requested with an unsolicited AC or AX command. See Run-Time Options subsection for valid options.

VALUE OF <value> FOR <option> IS NOT NUMERIC

TRACE and NEED require numeric values. Re-enter the request.

TRACE CANNOT BE SET TO <value>

TRACE can only be set to 0, 1, 2, or 3.

NOT READY ON <unit> -- PLEASE CORRECT AND AX WITH OK

The unit being squashed has become not ready. The unit must be readied for the squash to continue. The SYSTEM/SQUASH program waits for AC or AX command input before continuing. Data is not corrupted.

WRITE LOCKOUT ON <unit> -- PLEASE CORRECT AND AX WITH OK

A write lockout has occurred on the unit being squashed. The unit must be write-enabled before the squash operation can continue. The SYSTEM/SQUASH program waits for an AC or AX command input before continuing. Data is not corrupted.

Fatal Errors

When the SYSTEM/SQUASH program discovers an uncorrectable error it issues an error message describing the error, and then aborts with the message SQUASH ABORTED.

Following an abnormal termination of the SYSTEM/SQUASH program, the unit being squashed is not readied. The operator must check the integrity of the disk before using it. Refer to the Error Recovery subsection for details.

I/O UNCORRECTED AFTER 10 RETRIES

An irrecoverable disk I/O error occurred. This error indicates that the disk itself is corrupted.

INVALID FILE HEADER AT SECTOR <hex address>

A file header contained invalid information. Two possible causes of this error follow.

1. More than 105 areas were requested for the file.
2. The number of assigned areas for a file is greater than the number of areas requested.

PROGRAM ERROR

The squash aborted due to a programmatic error. The SYSTEM/SQUASH program generates a memory dump of its memory areas before going to EOJ. Save the memory dump, and call a Burroughs technical representative.

<integer> OVERLAPPING AREAS

Refer to the Integrity Check subsection.

BAD AVAILABLE TABLE

Refer to the Integrity Check subsection.

SECTION 50 TAPECOPY

The TAPECOPY program is a normal-state utility program that duplicates, compares, merges, and concatenates non-library, non-Load Dump multi-file or single-file data tapes. It allows selective deletion of any of the files processed and allows production of multiple copies of an output tape. The TAPECOPY program accepts input files from tape or disk.

ANSI standards do not allow a blank file-id on a multi-file tape; however, a blank file-id is allowed on a single-file tape.

The TAPECOPY program does not accept library tapes (produced by the SYSTEM/COPY program) or tapes created by the SYSTEM/LOAD.DUMP program. To copy or merge COPY library tapes, the SYSTEM/COPY program must be used. To copy, merge, or compare Load Dump tapes, the SYCOPY program must be used.

OPERATING INSTRUCTIONS

Execution and control of the TAPECOPY program is through the ODT. Upon execution, the TAPECOPY program displays the following message:

```
ENTER <ACTION: CPY, CMP, CCM, MRG, OR CAT>  
      <DEVICE: T(APE) or D(ISK)> <TAPE COUNT>
```

INPUT SPECIFICATIONS

The response to the previous message must have the following format:

```
<mix-number> AX <action> [ <device> ] [ <integer> ]
```

The following five possible actions can be specified in the response:

CPY

Produces <integer> tape copies of the input, with input files coming from a tape labeled SOURCE or disk files with a <family-id> of SYS. The input tape can be IL-ed if the name is other than SOURCE. If the <family-id> of the disk files is other than SYS, modify the TAPECOPY code as follows:

```
MODIFY TAPECOPY FI DISKFILE NAME <family-id>;
```

CMP

Compares (only) <integer> copies with the tape or disk files. The tapes to be compared are expected to have names of SOURCE and OUTSYS, but they can be IL-ed. Disk files are expected to have a <family-id> of SYS, but this can be changed by modifying TAPECOPY as shown previously. Files with different record sizes can be compared; in such a case, the shorter record is padded on the right with blanks.

CCM

Creates <integer> copies of the source tape or disk files and compares all copies with the original. If any tape is not identical with the original, that tape is purged and the system operator is notified by a console display message. The TAPECOPY program continues to compare the remaining tapes.

MRG

Merges the files from a source tape with the files on a second tape or a family of disk files and creates <integer> copies. The source tape is expected to be labeled SOURCE, and the update tape is expected to be labeled UPDATE. Both tapes can be IL-ed if the tape identifiers are other than the default. If the update files are on disk, they are assumed to have the same <family-id> as the label of the source tape. The TAPECOPY program can be modified as shown above to change the <family-id>, if necessary. The update file takes precedence in the case of duplicate files and is the file placed on the output tape.

CAT

Concatenates the input files onto one tape file with <integer> copies. The output tape is given the same identifier as the first input file. Subsequent input tapes are expected to be labeled NEXT-FILE but can be IL-ed if necessary. Only the first file on each tape is used. To terminate the concatenation process, enter the following message when a new tape is requested:

<mix-number> OF

Input disk files are expected to have a <family-id> of SYS, but this can be changed by modifying the TAPECOPY program as shown previously. The following message appears on the operator display terminal (ODT) for each disk file to be concatenated.

ENTER FILE ID OF NEXT FILE

An AC or AX command with blank input terminates the requests. If the TAPECOPY program is executed with switch 2 = 1, the <file=id>s are read from a card file labeled TCARDS, with one <file-id> per card.

From one to seven tapes can be specified by <integer>. If <integer> is omitted, one tape is assumed by default. If <device> is omitted, TAPE is assumed.

The disk files associated with all functions are expected to be on the system disk that has a <family-id> of SYS. Both the <pack-id> and the <family-id> can be changed by using the FILE statement. The <internal-file-id> of the disk file is DISKFILE.

PROGRAM SWITCHES

The TAPECOPY program responds to the values of program switches 0, 1, and 2 as shown in table 50-1.

Table 50-1. TAPECOPY Program Switches

Switch	Value	Meaning
0	1	Causes a summary listing of the actions taken and the files accessed by TAPECOPY to be printed on the line printer.
1	1	Allows files to be deleted. The file-identifiers are read from a card file labeled TCARDS, one file-id to a card.
1	2	Allows files to be deleted. The file-identifiers are entered from the ODT.
2	1	Valid only when concatenating files. Allows the file identifiers to be specified in the card file labeled TCARDS.
3	1	Causes TAPECOPY to read all command specifications from the card file TCARDS rather than from the ODT.

At the completion of processing, the TAPECOPY program displays the following messages:

<integer> TAPE(S) <action taken>

<integer> TAPE(S) FAILED TO COMPARE/PURGED

The second message is displayed only if any tapes failed to compare or were purged.

APPENDIX A DISK DEVICE CHARACTERISTICS

DISK CARTRIDGE CHARACTERISTICS

Physical Characteristics

	B 9480-12	B 9481-12	B 9482-32
Bytes-per-sector	180	180	180
Sectors-per-track	32	32	64
Tracks-per-cylinder	2	2	2
Cylinders-per-cartridge	203	406	406
Sectors-per-cartridge	12,992	25,984	51,968
Bytes-per-cartridge	2,338,560	4,677,120	9,354,240
Transfer Rate (MBITS/SEC)	1.55	1.55	3.10
Rotational Speed (RPM)	1,500	1,500	1,500

Initialization Error Limits

Errors-per-cartridge	406	812	812
----------------------	-----	-----	-----

Initialization Characteristics

All three types:

Initialize with @6363@ pattern
Read/Verify one pass

B 9484 (205 AND 206) DISK PACK CHARACTERISTICS

Physical Characteristics

	205 B 9484-25	206 B 9484-55
Bytes-per-sector	180	180
Sectors-per-track	90	90
Tracks-per-cylinder	5	5
Usable sectors-per-cylinder	445	445
Cylinders-per-pack	408	815
Usable cylinders-per-pack	407	814
Sectors-per-pack	183,600	366,700
Usable sectors-per-pack	181,115	362,230
Bytes-per-pack	33,048,000	66,015,000
Usable bytes-per-pack	32,600,700	65,201,400
Spare sectors-per-cylinder	5	5
Spare Sectors-per-pack	2,040	4,075
Usable spare sectors-per-pack	2,035	4,070
Transfer Rate (MBITS/SEC)		9.86
Nominal Rotational Speed (RPM)	3,672	3,672

Initialization Error Limits

Errors-per-cylinder	5	5
Errors-per-pack	2,035	4,070

Initialization Characteristics

Both types:

Initialize with @6363@ pattern
 Read/Verify three passes (0, +, -)
 Write @9C9C@ pattern
 Read/Verify three passes (0, +, -)

NOTE

The preceding 0, +, and - refer to verification with no offset (0), offset in (+), and offset out (-).

B 9499 (223, 225 AND 207) DISK PACK CHARACTERISTICS

Physical Characteristics

	223 B 9499-8	225 B 9499-7	207 B 9499-6
Bytes-per-sector	180	180	180
Sectors-per-track	60	60	90
Tracks-per-cylinder	20	20	8
Usable sectors-per-cylinder	1195	1195	715
Cylinders-per-pack	203	406	1,564
Usable cylinders-per-pack	203	406	1,563
Sectors-per-pack	243,600	487,200	1,126,080
Usable sectors-per-pack	242,585	485,170	1,117,545
Bytes-per-pack	43,848,000	87,696,000	202,694,400
Usable bytes-per-pack	43,665,300	87,330,600	201,158,100
Spare sectors-per-cylinder	5	5	5
Spare sectors-per-pack	1,015	2,030	7,820
Usable spare secs-per-pack	1,015	2,030	7,815
Transfer Rate (MBITS/SEC)		5.0	10.71
Nominal RPM	2,400	2,400	3,672

Initialization Error Limits

Errors-per-cylinder	5	5	715
Errors-per-pack	1,015	2,030	7,815

Initialization Characteristics

B 9499-8/7

Initialize with @6DB6@ pattern
 Read/Verify three passes (0, +, -)
 Write @6363@ pattern
 Read/Verify three passes (0, +, -)

B 9499-6

Initialize with @0000@ pattern
 Read/Verify three passes (+, -, 0), pattern @6DB@
 Read/Verify three passes (+, -, 0), pattern @DB6@
 Read/Verify three passes (+, -, 0), pattern @B6D@

NOTE

The preceding 0, +, and - refer to verification with no offset (0), offset in (+), and offset out (-) for the B 9499-8/7 (223, 225); and to normal (0), high (+) and low (-) threshold settings for the B 9499-6 (207).

B 9484-12, B 9494-5, AND B 9494-10 DISK PACK CHARACTERISTICS

Physical Characteristics

	677	659
	B 9484-12	B 9494-10 B 9494-5
Bytes-per-sector	180	180
Sectors-per-track	90	90
Tracks-per-cylinder	19	15
Usable sectors-per-cylinder	1705	1345
Cylinders-per-pack	823	2242
Usable cylinders-per-pack	822	2240
Sectors-per-pack	1,403,215	3,014,145
Usable sectors-per-pack	1,401,510	3,012,800
Bytes-per-pack	252,578,700	542,546,100
Usable bytes-per-pack	252,271,800	542,304,000
Spare sectors-per-cylinder	5	5
Spare Sectors-per-pack	4,115	11,210
Usable spare sectors-per-pack	4,110	11,200
Transfer Rate (MBITS/SEC)		
Nominal Rotational Speed (RPM)	3,672	3,672

Initialization Error Limits

Maximum removed sectors	100	1,600
Maximum transients per cylinder	20	20
Maximum transients per disk	1,000	1,000

Initialization Characteristics

Both types:

Initialize with @6DB0@ pattern
Write @6DB@ pattern
Read/Verify and compare three passes (+, -, 0)
Write @DB6@ pattern
Read/Verify three passes (+, -, 0)
Write @B6D@ pattern
Read/Verify three passes(+, -, 0)

NOTE

The preceding 0, +, and - refer to verification with no offset (0), offset in (+), and offset out (-).

APPENDIX B SYNTAX CONVENTIONS

Railroad diagrams show how syntactically valid statements can be constructed.

Traversing a railroad diagram from left to right, or in the direction of the arrow heads, and adhering to the limits illustrated by bridges produces a syntactically valid statement. Continuation from one line of a diagram to another is represented by a right arrow (→) appearing at the end of the current line and the beginning of the next line. The complete syntax diagram is terminated by a vertical bar (|).

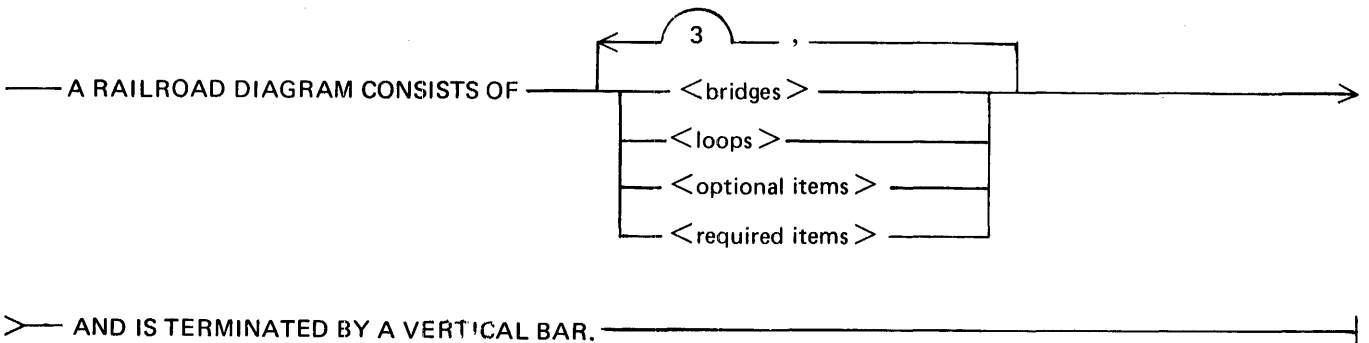
Words appearing in all upper-case characters are keywords and must appear as presented. Words in lower-case characters and embraced in left and right broken bracket characters are symbols for operator-supplied words or values.

Where keywords can be abbreviated, the abbreviation is denoted by the letters underlined in the keywords. Each keyword, keyword abbreviation, or operator-supplied item must be delimited from another such word by at least a single space or by a special character. Parentheses, commas, periods, and colon characters can be used as delimiters. Redundant spaces are ignored.

Items contained in broken brackets (< >) are syntactic variables that are further defined or are an indication the operator must supply requested information.

Upper-case keywords must appear literally. Minimum abbreviations are underlined.

Example:



G50051

The following syntactically valid statements can be constructed from the previous diagram:

A RAILROAD DIAGRAM CONSISTS OF <bridges> AND IS TERMINATED BY A VERTICAL BAR.

A RAILROAD DIAGRAM CONSISTS OF <optional items> AND IS TERMINATED BY A VERTICAL BAR.

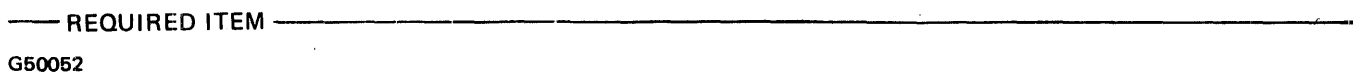
A RAILROAD DIAGRAM CONSISTS OF <bridges>, <loops> AND IS TERMINATED BY A VERTICAL BAR.

A RAILROAD DIAGRAM CONSISTS OF <optional items>, <required items>, <bridges>, <loops> AND IS TERMINATED BY A VERTICAL BAR.

REQUIRED ITEMS

No alternate path through the railroad diagram exists for required items.

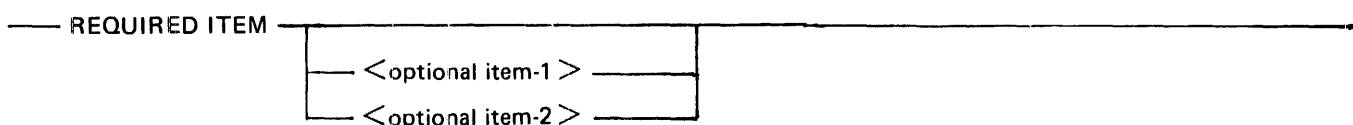
Example:



OPTIONAL ITEMS

Items shown as a vertical list indicate that the operator must make a choice of the items specified. An empty path through the list allows the optional item to be absent.

Example:



G50053

The following valid statements can be constructed from the previous diagram:

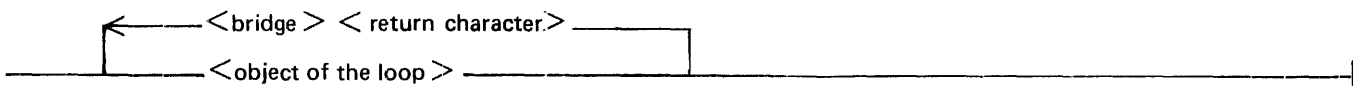
REQUIRED ITEM

REQUIRED ITEM <optional item-1>

REQUIRED ITEM <optional item-2>

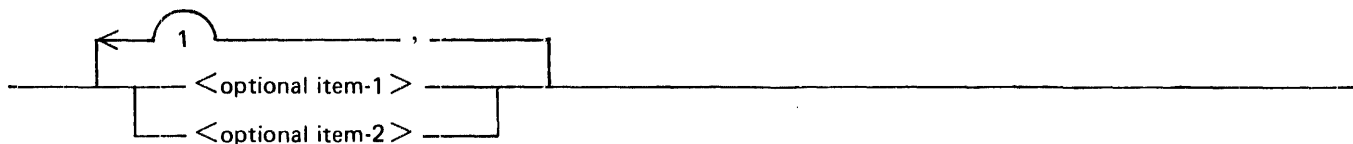
LOOPS

A loop is a recurrent path through a railroad diagram and has the following format:



G50054

Example:



G50055

The following statements can be constructed from the previous railroad diagram:


- < optional item-1 >
- < optional item-2 >
- < optional item-1 > , < optional item-1 >
- < optional item-1 > , < optional item-2 >
- < optional item-2 > , < optional item-1 >
- < optional item-2 > , < optional item-2 >

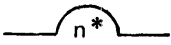
A loop must be traversed in the direction of the arrow heads, and the limits specified by bridges cannot be exceeded.

BRIDGES

A bridge illustrates the minimum or maximum number of times a path can be traversed in a railroad diagram.

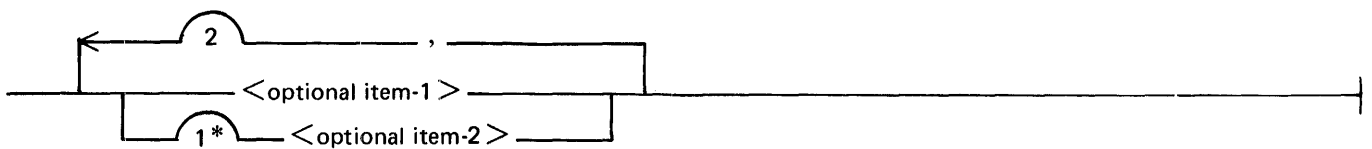
The following are the two forms of bridges.

 n is an integer that specifies the maximum number of times the path can be traversed.

 n is an integer that specifies the minimum number of times the path must be traversed.

G50056

Example:



G50057

The loop can be traversed a maximum of two times; however, the path for < optional item-2 > must be traversed at least one time.

The following statements can be constructed from the previous railroad diagram:

- < optional item-2 >
- < optional item-1 > , < optional item-2 >
- < optional item-2 > , < optional item-2 > , < optional item-1 >
- < optional item-2 > , < optional item-2 > , < optional item-2 >

APPENDIX C

BACKUP FILE FORMAT

Backup files produced by the MCPPI are in a fixed record-length, fixed blocking-factor format. The record size assigned to the backup file is two bytes larger than the record size of the original printer or punch file and one byte larger than the previous backup format. The extra bytes are used for carriage control and stacker select information. The record size for tape backup files is made modulo six bytes by the MCPPI due to the hardware requirements of 7-track and phase-encoded tapes. Backup files on tape cassette are not allowed.

The minimum allowable backup record size that can be written to disk is three bytes. This includes two bytes of control information plus one data byte. The block size of the backup file is computed by the MCPPI so that maximum usage of disk space is accomplished. The SYSTEM/BACKUP program always opens the input backup file with default in order to read it with the proper record size and blocking factor.

The first record of a backup file contains control information for the SYSTEM/BACKUP program. If the backup file record size is less than the size of any of the control records, the MCPPI produces multiple records, dividing the information between the records as necessary.

CONTROL INFORMATION

The control information begins in the first record of the backup file and consists of the fields (all numbers are in binary) listed in table C-1.

Table C-1. Control Information Record

Start Pos:Length	Description
[0:24]	Logical record size (in bits) of the backup file.
[24:6]	Increment (in bits) that was added to the record size for tape backup fields. This increment can be from 0 to 40 (always zero for disk files).
[30:18]	Length (in bits) of the control information.
[48:24]	Length (in bits) of the file parameter block (FPB) (refer to the subsection entitled FPB).
[72:24]	Length (in bits) of the label (refer to the subsection entitled Label).
[96:4]	Label type of the backup file. A value of 0001 indicates unlabeled. All other values indicate a labeled file.
[100:4]	Format type of the backup file. A value of 0001 indicates a Mark 9.0 release backup file.

+ 13.6 stuff

File Parameter Block (FPB)

A copy of the original printer/punch file parameter block (FPB) begins in the record following the control information.

Label

A copy of the beginning label for the printer/punch file (even for an unlabeled file) begins in the record following the FPB. The first two bytes of this record contain the appropriate carriage control information (refer to the subsection entitled Carriage Control Field).

Data Record

The remainder of the file following the label record consists of printer/punch data records. A data record which contains all zeroes is ignored. It is a filler record created by execution of a CLOSE ROLL-OUT operation on the backup file.

Carriage Control Field

The first two bytes (16 bits) of each data record comprise the carriage control field, the contents of which specify the spacing, skipping, or stacker selection required. For punch backup files without stacker selection the control field is not used. The low-order four bits of the control field, referred to as TYPE, determine the specific action to be taken. For example, a value of 0011 is TYPE 3. The five field types are shown in table C-2.

Table C-2. TYPE Field Definition

TYPE	Function
0	Print the data buffer with channel skipping or single or double spacing as specified in bits 4 through 11 of the 2-byte carriage control field.
1	Punch the data buffer, and route the card to the stacker specified in bits 4 through 11 of the 2-byte carriage control field.
2	Space the number of lines specified by bits 4 through 11 of the 2-byte carriage control field. The data buffer is ignored. A maximum of 255 spaces is allowed.
3	Skip to the channel specified in bits 4 through 11 of the 2-byte carriage control field. The data buffer is ignored.
4	Print the data buffer, and space as specified in bits 4 through 11 of the 2-byte carriage control field. A maximum of 255 spaces is allowed.

Following is the format of the carriage control field:

Bits 0-2	Reserved.
Bit 3	WRITE BEFORE or WRITE AFTER bit. If reset (0), MCP II prints before spacing; if set (1), MCP II prints after spacing.
Bits 4-11	Printing: skip and space information; punching: stacker selection information.
Bits 12-15	Type of operation.

Tables C-3 and C-4 display the meanings of the space and skip information (bits 4 through 11) for the various field types. Table C-5 shows the values of the 16-bit carriage control field for some commonly used operations.

Table C-3. Values for Printer Spacing/Skipping

TYPE	Value	Function
0 or 3	@00@	No paper advance
0 or 3	@01@	Skip to channel 1
0 or 3	@02@	Skip to channel 2
-	-	-
-	-	-
-	-	-
0 or 3	@0C@	Skip to channel 12
0 or 3	@0D@	Skip to next channel
0 or 3	@0E@	Space one line
0 or 3	@0F@	Space two lines
2 or 4	@00@-	Space number of lines
	@FF@	
	(0-255)	indicated (0-255)

Table C-4. Values for Punch Stacker-Selection

TYPE	Value	Description
1	@00@	Reject Stacker
1	@01@	Stacker 1
1	@02@	Stacker 2
-	-	-
-	-	-
-	-	-
1	@06@	Stacker 6

Table C-5. Some Carriage Control Field Values

Value	Meaning
@0000@	Print data buffer; no paper advance.
@0010@	Print data buffer; then skip to channel one.
@1010@	Skip to channel one; then print data buffer.
@00E0@	Print data buffer; then single space.
@10E0@	Single space; then print data buffer.
@0013@	Skip to channel one (no print).
@0052@	Space five lines (no print).
@0012@	Space one line (no print).
@0000@	Punch and stacker-select (reject pocket).
@0041@	Punch and stacker-select (pocket number 4).

INDEX

Card File Copy 1-6
Cassette File Copy 1-6
Categories of Utility Programs 1-1
Copy/File Copy Programs 1-3
Disk File Copy 1-4
Disk Initializer Programs 1-7
Diskette File Copy 1-6
File Initializer Program 1-7
General Description of Utility
Programs 1-3
General Maintenance Programs 1-7
Log Analysis Programs 1-9
Printer Listings 1-5
System Start-Up Programs 1-3
Tape File Copy 1-5

Error Messages 2-2
Files Created 2-1
Informative Messages 2-2
Input Specifications 2-1
Operating Instructions 2-1
Program Output 2-1
Sample Execution Commands 2-1

Program Product No Longer Supported 3-1

B 1990 Systems 4-5
Clear Start Functions 4-1
Clear-starting B 1990 Systems 4-3
Clear-starting Other B 1000 Systems 4-3
CLEAR/START Operating Instructions 4-2
Description of the Name Table 4-4
Disk Cache Disable Function 4-11
Disk Drive Unit Order Selection 4-10
I/O Channel Deletion 4-8
MCP Functions 4-2
Other B 1000 Systems 4-7
Pseudo-Memory Size Specification 4-9
Supplementary Information Required With
A Memory Dump 4-11
System Disk Channel Selection Override 4-8
System Initialization Functions 4-2
Temporary Environment Changes 4-5

Operating Instructions 5-1
Program Switches 5-1

COLDSTART/DISK Program Overview 6-1
Error and Advisory Messages to the ODT 6-6

INDEX (CONT)

Halts 6-6
Input and Output Disks 6-1
Operating Instructions 6-3
Program Execution 6-4
Program Loading -- All Other B 1000
Systems 6-4
Program Loading -- B 1990 Systems 6-3
Program Options 6-2
System Software Files 6-2

Halts 7-4
Input and Output Media 7-1
Operating Instructions 7-2
Program Execution 7-3
Program Loading -- All Other B 1000
Systems 7-3
Program Loading -- B 1990 Systems 7-2
Program Options 7-2
Program Overview 7-1
System Software Files 7-2

Error Messages 8-2
Operating Instructions 8-1

File Screen Semantics 9-2
Key Screen Semantics 9-3
ODT Interface 9-4
ODT Interface Messages 9-7
ODT Interface Sample Executions 9-6
Operating Instructions 9-1
Output Messages 9-6
Terminal Interface 9-1
Terminal Interface Messages 9-6

Error Messages 10-5
Internal Files 10-6
Operating Instructions 10-2
Option Summary 10-4
Program Input 10-2
Translate Header Format 10-1
Translation Table 10-1

Audit File Blocking 11-2
Audit File Buffering 11-2
DC/AUDIT Mnemonic Characters 11-17
DC/AUDIT Operating Procedures 11-2
Executing the DC/AUDIT Program 11-3
I/O Completions 11-15
Input Commands 11-5

INDEX (CONT)

ODT Messages 11-4
Output Format 1 11-12
Output Format 2 11-13
Output Format 3 11-14
Output Format 4 11-15
Overview 11-1
Parameter Entry Alternatives 11-2
Parameter Entry Error Message 11-3
Parameter Entry Format 11-3
Parameters 11-5
Printer Output Formats 11-11
Processing 11-4
Report Formats 11-14
Restrictions 11-1
Structure of the Audit File 11-1
Termination 11-5
The Audit Capability 11-1
Writing The Audit File 11-1

Allocated File Structure 12-2
Command Specifications 12-1
Error Messages 12-3
Execution 12-1
Operating Instructions 12-1
Sample Execution 12-2

Error Messages 13-3
Operating Instructions 13-2
Program loading on B 1990 Systems 13-1
Program loading on other B 1000
Systems 13-2
STATUS Command 13-3

AUTOLOAD Facility 14-2
BYPASS Option 14-4
COPY Command 14-2
Error Messages 14-7
Input File Specifications 14-3
Input Specifications 14-1
KA Command 14-4
Operating Instructions 14-1
Output File Specifications 14-4
Program Switches 14-1
PSEUDO Option 14-4
PURGE Command 14-6
RELABEL Command 14-6
Routine Type Specifier 14-3
Sample Input Specifications 14-6

INDEX (CONT)

Error Messages 15-2
Operating Instructions 15-1

Card Files 16-25
CAT (Concatenation) Command 16-5
Commands 16-5
COPY (or PERFORM) Command 16-7
Disk Files 16-25
Error Messages 16-27
Files 16-25
Input to the DMPALL Program 16-3
Internal Files 16-26
LIST Command 16-10
Operating Instructions 16-1
Output Files 16-25
Paper Tape Files 16-25
PD (Print Directory) Command 16-12
Printer Listings 16-25
Program Switches 16-2
Sample Executions 16-4
Sample Input Specification Strings 16-3
Specifications 16-13
Tape Files 16-25
Tape Parity Modifications 16-4

Error Messages 17-12
Input Specifications 17-3
Internal Files 17-14
Operating Instructions 17-1

Error Messages 18-1
Operating Instructions 18-1
Program Output 18-1

COBOL Record Formats 19-22
Error and Advisory Messages 19-2
General Information 19-1
Operating Instructions 19-1
Output File Format 19-1
Record Format Tables 19-3
Record Types 19-2
RPGII Record Formats 19-20
Sample Program Record Declarations 19-13

Action 20-6
Card Reader Operation 20-4
CYLINDER NUMBER 20-8
DATE 20-8
Dollar Sign Options 20-11

INDEX (CONT)

Execution Of The RD Program (With SMCS) 22-8
File Identifier 22-3
FN (File Name) 22-17
FRZ (Freeze) 22-20
Hardware Requirements 22-4
HELP (or TEACH) 22-18
ID (Identification) 22-18
Initializing the MCP Name Table 22-6
Installation Procedures 22-5
Introduction 22-1
JOBS (Program Information) 22-19
KA 22-20
KB (ODT Queue) 22-22
KC 22-24
KP 22-26
Memory Requirements 22-4
MIX 22-28
ODT Commands 22-50
OL 22-31
OMX (Offline Mix) 22-33
Operating Instructions 22-8
PAGE 22-51
PD (Print Directory) 22-34
PQ (Print Queue) 22-35
PRG (Programs) 22-50
Program Name 22-3
PRT (Print) 22-36
QF (Query File) 22-38
QP (Query Program) 22-40
Setting the RD Program Priority 22-7
Setting the RD Program Switches 22-7
SKIP and PAGE Keywords 22-3
Software Requirements 22-5
Support Capabilities 22-4
TD (Time and Date) 22-41
TEACH (See HELP Command) 22-41
Terminal Commands 22-10
Terminal Requirements 22-5
Termination Of The RD Program 22-9
TO (Test Option) 22-41
TRANSLATE 22-42
UF (User of File) 22-43
UP (User of Pack) 22-44
US (User) 22-45
USE (Usage Information) 22-46
W (Wait) 22-47
WHO (Current User Info) 22-47
WS (Waiting Schedule) 22-48

INDEX (CONT)

BACK 23-6
BYE 23-6
CANDE 23-6
Command Syntax 23-3
EOJ 23-7
ERROR 23-7
FIND 23-8
GET 23-8
HEX 23-9
Introduction 23-1
MARGIN 23-10
NEXT 23-10
Operating Instructions 23-1
PAGE 23-11
PRINT 23-12
Program Commands 23-4
Program Switches 23-2
REMOVE 23-14
SAME 23-14
SEARCH 23-14
SEQ 23-15
TEACH 23-16
TERMINAL 23-16
USER 23-16
WAIT 23-17
WARNING 23-17
WHAT 23-18

Backup File Names 24-5
Backup Output Translation 24-15
Banners 24-12
BANNERS 24-26
Channel Skips 24-7
CHARSPERINCH 24-27
DEBUG 24-20
Declaration Section 24-9
Default Options (Attributes
and Switches) 24-11
Default Options Summary 24-4
DEVICES 24-20
DOUBLESPACE 24-28
DUMP 24-20
EOJ 24-20
Error Messages 24-5
Example Network Controller 24-10
EXTRACT 24-29
File Section 24-9
File security 24-5
File Security Considerations 24-5

INDEX (CONT)

File Specification 24-23
FONT 24-31
HELP 24-20
INDENT 24-32
Line Section 24-9
LINESPERINCH 24-32
Network Controller Installation 24-8
ODT Execution 24-17
ODT Output Translation 24-15
OK 24-20
Operating Instructions 24-17
Output Device 24-13
Output Message Blocking 24-11
Overview and Features 24-2
Page Size 24-13
Page Width 24-12
PAGESIZE 24-33
PAGEWIDTH 24-33
PRINT 24-20
PRINT Command Options 24-23
Print Option Summary 24-3
PRINTCOPIES 24-34
Printer Buffer Sizes 24-7
Printer Busy Retries 24-5
Printer Switching 24-14
Program Command Summary 24-2
Program Commands 24-18
Program Termination 24-15
Record Size 24-23
Remote File Title 24-14
Remote Printer Control Characters 24-6
SAVEFILE 24-34
Saving/Purging of the Backup File 24-13
SKIP 24-21
Station Section 24-9
STATUS 24-21
STOP 24-21
Summary of Default Program
 Attributes 24-16
SUPPRESS 24-35
Terminal Section 24-8
TEST 24-22
TOPMARGIN 24-35
UPPERCASE 24-36
USER 24-37
USER option 24-5
WFL Execution 24-17
WHAT 24-22

INDEX (CONT)

Example 25-3
Operating Instructions 25-1

Creating MIL Program Cassettes 26-3
Creating SDL Program Cassettes 26-3
Error Correction 26-4
Error Messages 26-5
Modes 26-3
Operating Instructions 26-1
Program Switches 26-2

Error Messages 27-5
Program Loading on B 1990 Systems 27-2
Program Loading on Other B 1000
Systems 27-2
Program Operating Instructions 27-3
STANDALONE/DISK-DUMP Halts 27-8
STATUS Command 27-4

AB System Command 28-3
Error Messages 28-6
File Specifier Options 28-2
Files and Internal Structures 28-7
Operating Instructions 28-1
Options 28-3
Output Options 28-3
PB System Command 28-2
Program Switches 28-4
References 28-1
Run-Time Control Options 28-5

Input Record Format 29-2
Operating Instructions 29-1
Program Switches 29-2
Standard Translate Tables 29-3
Translate Table Format 29-1

Command Specifications 30-2
Errors 30-5
Fatal Errors 30-5
Informative Messages 30-8
Operating Instructions 30-1
Output Messages 30-5
Program Switches 30-2
Run-Time Interrogation 30-4
Warning Messages 30-7

ADD Command 31-4
Command String Clauses 31-9

INDEX (CONT)

COMPARE Command 31-6
COMPARE Errors (Line Printer Listings) 31-36
Comparison Errors 31-26
COPY Command 31-5
DIR Command 31-18
Disk to Disk 31-22
Duplicate File Identifier Checking 31-28
Error and Advisory Messages 31-28
Error Handling 31-25
Errors When Using HOSTNAME 31-35
Explanatory Messages 31-34
File Identifiers 31-2
FROM Clause, TO Clause 31-14
I/O Errors 31-25
In-Use Files 31-26
Library Maintenance Messages 31-33
Message Output Control (Switch Ø,
 MCPII COPY Option) 31-28
Messages Denoting Actions 31-34
Missing Files 31-27
Operating Instructions 31-2Ø
Operator Input (AC or AX Messages) 31-23
Overview 31-1
Program Switches 31-2Ø
QUIT Command 31-23
STATUS Command 31-23
Syntax and Semantics Advisory Messages 31-32
Syntax and Semantics Error Messages (Fatal
 Errors) 31-29
SYSTEM and User Disks 31-3
SYSTEM Disk to Tape 31-22
SYSTEM/COPY Input: Sample Strings 31-21
SYSTEM/COPY Output 31-24
SYSTEM/COPY Program Commands 31-4
SYSTEM/COPY Speed Considerations 31-28
Tape to Tape 31-22
TEACH Command, HELP Command 31-23
Terminology 31-2
User Disk to Tape 31-21

Dump Specifications 32-1
Error Messages 32-3
Informative Messages 32-2
Operating Instructions 32-1
Output Disk 32-2
Program Output 32-2

Action 33-4
Card Reader Operation 33-3

INDEX (CONT)

Dollar Sign Option 33-8
Error Messages 33-13
Informative Messages 33-12
Initialization Integrity Protection 33-7
Initialization Messages 33-6
Initialization Options 33-7
Initialization Specifications 33-4
LIMIT Option 33-8
Marginal Sectors Option 33-8
ODT Operation 33-2
Program Operation Instructions 33-2
Program Output 33-10
Program Switch Settings 33-2
REMOVE Option 33-8
STATUS Command 33-3
Summary Listing 33-12
User Interface Information 33-4

Cassette Error Report 34-3
Configuration Disk Unit Error Rate
Report 34-7
Disk Cache Error Report 34-3
Disk Error Address Analysis 34-5
Error Message 34-2
Memory Error Report 34-3
Operating Instructions 34-1
Operator Message Report 34-2
Output Reports 34-2
Power On/Off Error Rate Report 34-6
Program Switches 34-1
Result Code Definitions 34-4
Result Descriptor Analysis 34-5
System Information Report 34-2
Unit Error Reports 34-4
Unit Error Summary 34-7

Error Messages 35-3
Examples 35-3
Initialization 35-1
Program Execution 35-2
Relative Files 35-1

Error Messages 36-2
Operating Instructions 36-1

Commands 37-2
ERROR Messages 37-6
Error Messages (ERRORs and WARNINGs) 37-6
Input From a Data File 37-1

INDEX (CONT)

Input from the ODT 37-1
Internal Files 37-8
Operating Instructions 37-1
Sample Execution 37-5
WARNING Messages 37-6

Operating Instructions 38-1
Program Output 38-1
Program Switch Setting 38-1
Sample File Output 38-2

General Rules 39-2
Operating Instructions 39-1

Error Messages 40-1

Program Product No Longer Supported 41-1

CLEAR/START Records 42-1
Comment Records 42-3
DMS Information 42-2
File Information Records 42-3
Job Information Records 42-2
Operating Instructions 42-1
Program Output 42-1

ADD Command 43-4
CHANGE Command 43-6
Commands 43-4
COPY Command 43-7
CREATE Command 43-8
DEBUG Command 43-9
DELETE Command 43-9
DISPLAY Command 43-10
END/EOJ Command 43-11
Error Messages 43-14
File Security Attributes 43-2
HELP Command 43-11
LIST Command 43-11
Operating Instructions 43-1
PUNCH Command 43-13
The PURGE Command 43-13

Input Specifications 44-1
Operating Instructions 44-1
Program Output 44-2

Files (Reference Information) 45-1
KB System Command 45-1

INDEX (CONT)

Memory Requirements 45-2
NDL Handler (B 1990 Systems) 45-1
Operating Instructions 45-1

Error Messages 46-2
Internal Files 46-3
ODTLOG File Layout 46-3
Operating Instructions 46-1
Switch Settings 46-2

Comparison Errors 47-5
Execution Errors 47-3
Execution Examples 47-2
I/O Errors 47-4
Operating Instructions 47-2
Other Considerations 47-1
Overview 47-1
Restrictions 47-1
Special Operational Considerations 47-7
Synopsis of Program Operation 47-1
Syntax Errors 47-3

Error and Advisory Messages 48-4
MAP and CHECK Related Messages 48-4
Notes on SAVE Option File 48-7
Operating Instructions 48-2
Options Keyword Glossary 48-1
Other Advisory and Error Messages 48-6
Program Execution 48-4
Program Switches 48-3
Required System File 48-1
User File Heading 48-14

Error Messages 49-5
Error Recovery 49-4
Fatal Errors 49-6
Integrity Check 49-2
Nonfatal Errors 49-5
Operating Instructions 49-1
Program Output 49-4
Program Switches 49-2
Run-Time Options 49-3
Squashed Disk 49-4
Summary Listing 49-4
Switch 1 49-2
Switch 2 49-2
Trace Functions 49-3

Input Specifications 50-1
Operating Instructions 50-1
Program Switches 50-3

Documentation Evaluation Form

Title: B 1000 Systems System Software Operations Guide

Form No: 1169091
Date: February 1985

Burroughs Corporation is interested in receiving your comments
and suggestions, regarding this manual. Comments will be util-
ized in ensuing revisions to improve this manual.

Please check type of Suggestion:

- Addition Deletion Revision Error

Comments:

From: Name _____
Title _____
Company _____
Address _____

Phone Number _____ Date _____

Remove form and mail to:

Burroughs Corporation
Corporate Documentation - West
1300 John Reed Court
City of Industry, CA 91745
U.S.A.